



Ακαδημαϊκό έτος 2022-23

Ψηφιακά Συστήματα

ΠΛΗ21

2^η Γραπτή Εργασία

Copyright © 2022 Φροντιστήριο ΑΡΝΟΣ, επιμέλεια: ΕΥΘΥΜΗΣ ΜΠΑΛΤΖΟΓΛΟΥ (email: uthimis@hotmail.com)

Το παρόν έργο πνευματικής ιδιοκτησίας προστατεύεται κατά τις διατάξεις του Ελληνικού Νόμου (Ν. 2121/1993 όπως έχει τροποποιηθεί και ισχύει σήμερα) και τις διεθνείς συμβάσεις περί πνευματικής ιδιοκτησίας. Απαγορεύεται απολύτως η -κατά οποιονδήποτε τρόπο ή μέσο και άνευ γραπτής αδείας του δημιουργού/εκδότη- αντιγραφή, φωτοανατύπωση και εν γένει αναπαραγωγή, εκμίσθωση ή δανεισμός, μετάφραση, διασκευή, αναμετάδοση στο κοινό σε οιαδήποτε μορφή (ηλεκτρονική, μηχανική κ.λ.π.) και η εν γένει εκμετάλλευσή του συνόλου ή μέρους του έργου αυτού.

Έξυπνα και Εύκολα!

Άσκηση 1^η (15 μονάδες)

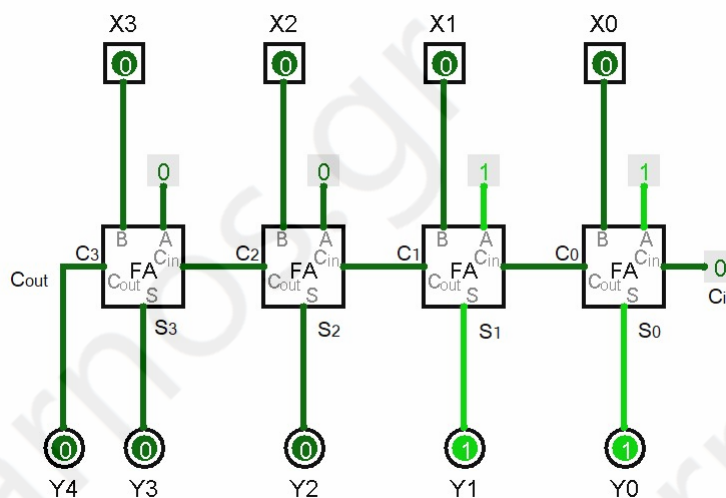
ερώτημα (α)

Θα παρατηρήσουμε ότι καθώς η είσοδος X «κινείται» στο διάστημα $(0,15)$ η έξοδος Y θα βρίσκεται στο διάστημα $(3,18)$, χρειάζονται λοιπόν 5 bits για να την παραστήσουμε:

X	X ₃	X ₂	X ₁	X ₀	Y	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	X+3=3	0	0	0	1	1
1	0	0	0	1	X+3=4	0	0	1	0	0
2	0	0	1	0	X+3=5	0	0	1	0	1
3	0	0	1	1	X+3=6	0	0	1	1	0
4	0	1	0	0	X+3=7	0	0	1	1	1
5	0	1	0	1	X+3=8	0	1	0	0	0
6	0	1	1	0	X+3=9	0	1	0	0	1
7	0	1	1	1	X+3=10	0	1	0	1	0
8	1	0	0	0	X+3=11	0	1	0	1	1
9	1	0	0	1	X+3=12	0	1	1	0	0
10	1	0	1	0	X+3=13	0	1	1	0	1
11	1	0	1	1	X+3=14	0	1	1	1	0
12	1	1	0	0	X+3=15	0	1	1	1	1
13	1	1	0	1	X+3=16	1	0	0	0	0
14	1	1	1	0	X+3=17	1	0	0	0	1
15	1	1	1	1	X+3=18	1	0	0	1	0

ερώτημα (β)

Κάνουμε ακριβώς ότι ζητείται στην εκφώνηση: στον δυαδικό αριθμό $X_3X_2X_1X_0$ των 4 bits, με 4 FAs προθέτουμε το $(3)_{10}$, δηλαδή το $(0011)_2$ με 0 στο κρατούμενο εισόδου και παίρνουμε ως αποτέλεσμα $Y_4Y_3Y_2Y_1Y_0$ σε 5 bits, όπου το MSbit Y_4 θα είναι το κρατούμενο εξόδου C_3 .



Σημειώστε ότι για το συμπλήρωμα σε όλη την εργασία χρησιμοποιώ την από πάνω $\neg$ και όχι τον $'$

Έξυπνα και Εύκολα!

ερώτημα (γ)

Από τη θεωρία γνωρίζουμε ότι οι συναρτήσεις των εξόδων στον HA είναι:

$$S = A \oplus B$$

$$C_{out} = AB$$

ενώ για τον FA αντίστοιχα είναι:

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + BC_{in} + AC_{in}$$

Αρκεί λοιπόν να κάνουμε αλγεβρικές πράξεις με τις σταθερές εισόδους του ερωτήματος (β):

$$Y_0 = S_0 = \underbrace{X_0 \oplus 1}_{\bar{X}_0} \oplus C_{in} = \bar{X}_0 \oplus 0 = \bar{X}_0 \text{ και}$$

$$C_0 = X_0 \cdot 1 + 1 \cdot C_{in} + X_0 \cdot C_{in} = X_0 + 1 \cdot 0 + X_0 \cdot 0 = X_0$$

Οπότε ο 1^{ος} FA μπορεί να αντικατασταθεί με μόνο 1 πύλη NOT. Συνεχίζοντας τις πράξεις:

$$Y_1 = S_1 = \underbrace{X_1 \oplus 1}_{\bar{X}_1} \oplus C_0 = \bar{X}_1 \oplus X_0 = X_1 \odot X_0 \text{ δηλαδή μία πύλη XNOR και}$$

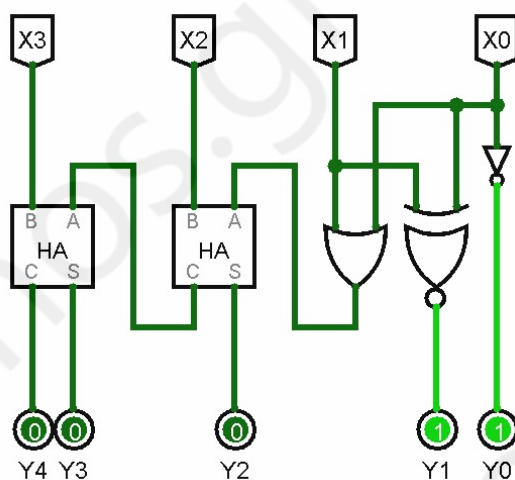
$$C_1 = X_1 \cdot 1 + 1 \cdot C_0 + X_1 \cdot C_0 = X_1 + X_0 + X_1 \cdot X_0 = X_1 + X_0 \text{ δηλαδή μία πύλη OR}$$

Οπότε ο 2^{ος} FA μπορεί να αντικατασταθεί με 2 πύλες 2 εισόδων, μία XNOR και μία OR. Από εδώ και πέρα, οι υπόλοιποι FAs μπορούν εύκολα να αντικατασταθούν με HAs αν αγνοήσουμε τις σταθερές εισόδους με 0 που έχουν (οι οποίες έτσι κι αλλιώς δεν θα μετέχουν στις πράξεις):

$$Y_2 = S_2 = \underbrace{X_2 \oplus 0}_{X_2} \oplus C_1 = X_2 \oplus C_1 \text{ και } C_2 = X_2 \cdot 0 + 0 \cdot C_1 + X_2 \cdot C_1 = X_2 \cdot C_1$$

$$Y_3 = S_3 = \underbrace{X_3 \oplus 0}_{X_3} \oplus C_2 = X_3 \oplus C_2 \text{ και } Y_4 = C_3 = X_3 \cdot 0 + 0 \cdot C_2 + X_3 \cdot C_2 = X_3 \cdot C_2$$

Καταλήγουμε λοιπόν σε μία βέλτιστη υλοποίηση με 2 HAs, 2 πύλες 2 εισόδων και 1 NOT:



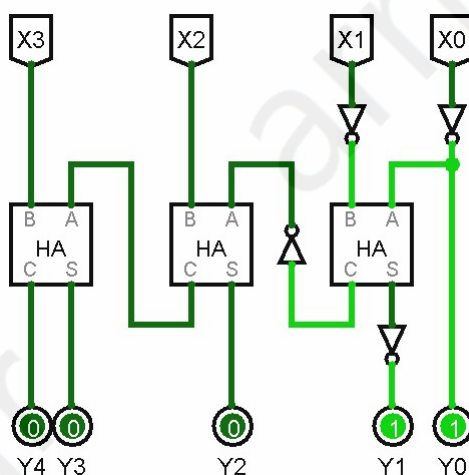
Έξυπνα και Εύκολα!

Αν συνεχίζαμε λίγο τις παραπάνω πράξεις για τα Y_1 και C_1 :

$$Y_1 = X_1 \odot X_0 = \overline{X_1 \oplus X_0} = \overline{\overline{X_1} \oplus \overline{X_0}} \quad \text{και}$$

$$C_1 = X_1 + X_0 = \overline{\overline{X_1} \overline{X_0}}$$

Οπότε ο 2^{os} FA θα μπορούσε να αντικατασταθεί με έναν HA και 3 πύλες NOT (οι 2 στις εισόδους X_1 και X_0 και η 3^η στην έξοδο C_1 του κρατουμένου). Οι υπόλοιποι 2 FAs όπως και πριν θα αντικατασταθούν με 2 HAs. Θα καταλήγαμε λοιπόν σε μία ισοδύναμη εναλλακτική υλοποίηση με 3 HAs και μόνο πύλες NOT (4 συνολικά):



Στην εκφώνηση για το ερώτημα (γ) ζητείται ελάχιστος αριθμός HAs και ταυτόχρονα ελάχιστος αριθμός πυλών, αυτό είναι αντικρουόμενο. Συγκρίνοντας τις 2 υλοποιήσεις, πρέπει κάποιος να σταθμίσει αν αξίζει τον κόπο να «θυσιάσει» έναν επιπλέον HA ώστε να χρησιμοποιήσει μόνο πύλες NOT (περισσότερες αριθμητικά αλλά απλούστερες και φθηνότερες από τις OR ή XNOR).

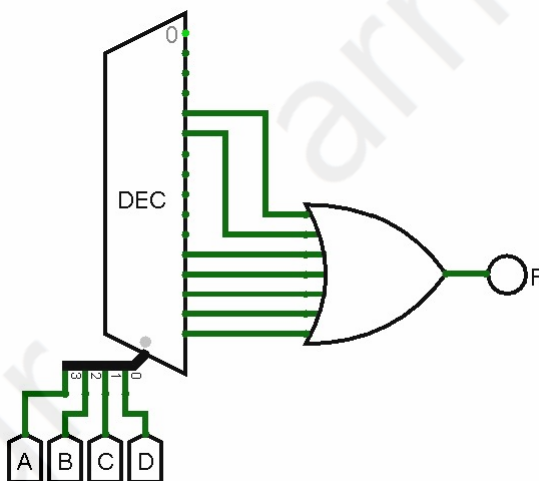
Άσκηση 2^η (15 μονάδες)

ερώτημα (α)

Για την υλοποίηση με τον Decoder, αρκεί να αναπτύξουμε την συνάρτηση σε κανονικό SoP:

$$\begin{aligned} F(A,B,C,D) &= AB + \bar{B}\bar{C} + ACD = AB(\bar{C} + C)(\bar{D} + D) + (\bar{A} + A)\bar{B}\bar{C}(\bar{D} + D) + A(\bar{B} + B)CD = \\ &= AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD = \\ &= \underbrace{\bar{A}\bar{B}\bar{C}\bar{D}}_{m_4} + \underbrace{\bar{A}\bar{B}\bar{C}D}_{m_5} + \underbrace{\bar{A}B\bar{C}\bar{D}}_{m_{11}} + \underbrace{\bar{A}B\bar{C}D}_{m_{12}} + \underbrace{\bar{A}BC\bar{D}}_{m_{13}} + \underbrace{\bar{A}BCD}_{m_{14}} + \underbrace{A\bar{B}C\bar{D}}_{m_{15}} = \sum(4, 5, 11, 12, 13, 14, 15) \end{aligned}$$

Θα χρειαστούμε λοιπόν έναν Decoder 4σε16 και μία OR 7 εισόδων (όσοι και οι ελαχιστόροι):



ερώτημα (β)

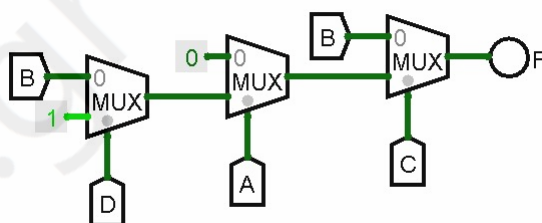
Ενώ η υλοποίηση με τον Decoder παραπάνω είναι «ξεκάθαρη» (με όποια σειρά κι αν πάρουμε τις μεταβλητές θα καταλήξουμε πάντα στο ίδιο πλήθος ελαχιστόρων άρα σε ιδίου μεγέθους OR), στην υλοποίηση με τους πολυπλέκτες θα πρέπει να σκεφτούμε «out-of-the-box» καθώς η σειρά επιλογής μεταβλητών εδώ θα δώσει διαφορετικές και ενίοτε απλούστερες υλοποιήσεις (δηλαδή πρέπει να δοκιμάσουμε ΟΛΟΥΣ τους δυνατούς συνδυασμούς ώστε να αποφανθούμε για το πιο απλό κύκλωμα!). Ενδεικτικά θα αναφέρω ότι αν ακολουθήσει κάποιος την τυπική σειρά A, B, C, D θα χρειαστεί 4 πολυπλέκτες 2σε1 και 1 πύλη NOT, ενώ αν επέλεγε την σειρά C, A, D, B:

Εφαρμόζοντας διαδοχικά το θεώρημα Shannon με την προαναφερθείσα σειρά μεταβλητών:

$$\begin{aligned} F(A,B,C,D) &= \bar{C} \cdot F(A,B,0,D) + C \cdot F(A,B,1,D) = \bar{C} (AB + \bar{B}\bar{C} + ACD)_{C=0} + C (AB + \bar{B}\bar{C} + ACD)_{C=1} = \\ &= \bar{C} (AB + \bar{B}\bar{0} + A\bar{0}D) + C (AB + \bar{B}\bar{1} + A\bar{1}D) = \bar{C} (AB + \bar{B}) + C (AB + AD) = \boxed{\bar{C}(B) + C(AB + AD)} = \\ &= \bar{C}(B) + C(\bar{A}(AB + AD)_{A=0} + A(AB + AD)_{A=1}) = \bar{C}(B) + C(\bar{A}(0 \cdot B + 0 \cdot D) + A(1 \cdot B + 1 \cdot D)) = \\ &= \bar{C}(B) + C(\boxed{\bar{A}(0) + A(B + D)}) = \bar{C}(B) + C(\bar{A}(0) + A(\bar{D}(B + D)_{D=0} + D(B + D)_{D=1})) = \\ &= \bar{C}(B) + C(\bar{A}(0) + A(\bar{D}(B + 0) + D(B + 1))) = \bar{C}(B) + C(\bar{A}(0) + A(\boxed{\bar{D}(B) + D(1)})) \end{aligned}$$

Έξυπνα και Εύκολα!

Καταλήγουμε δηλαδή σε μία βέλτιστη υλοποίηση με 3 μόνο πολυπλέκτες 2σε1, ως εξής:

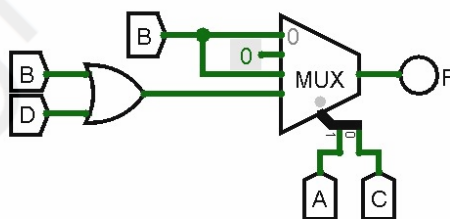


ερώτημα (γ)

Εφαρμόζοντας και πάλι το θεώρημα Shannon με την ίδια σειρά μεταβλητών όπως παραπάνω, επιλέγοντας όμως αυτήν την φορά τις 2 πρώτες (A και C) προκειμένου για πολυπλέκτη 4σε1:

$$\begin{aligned} F(A,B,C,D) &= \bar{A}\bar{C} \cdot F(0,B,0,D) + \bar{A}C \cdot F(0,B,1,D) + A\bar{C} \cdot F(1,B,0,D) + AC \cdot F(1,B,1,D) = \\ &= \bar{A}\bar{C}(0 \cdot B + B \cdot 0 + 0 \cdot 0 \cdot D) + \bar{A}C(0 \cdot B + B \cdot 1 + 0 \cdot 1 \cdot D) + A\bar{C}(1 \cdot B + B \cdot 0 + 1 \cdot 0 \cdot D) + AC(1 \cdot B + B \cdot 1 + 1 \cdot 1 \cdot D) = \\ &= \boxed{\bar{A}\bar{C}(B) + \bar{A}C(0) + A\bar{C}(B) + AC(B+D)} \end{aligned}$$

Καταλήγουμε έτσι σε μία βέλτιστη υλοποίηση με 1 πολυπλέκτη 4σε1 και 1 πύλη OR ως εξής:



Εύκολα μπορεί να διαπιστώσει κανείς ότι και τα τρία κυκλώματα είναι ισοδύναμα με το αρχικό (δοθείσα συνάρτηση F), καθώς οι αντίστοιχες συναρτήσεις έχουν την ίδια αποτίμηση στον ΠΑ:

A	B	C	D	F	Fdec	F2tol	F4tol
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	1	1	1	1
0	1	0	1	1	1	1	1
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1

Έξυπνα και Εύκολα!

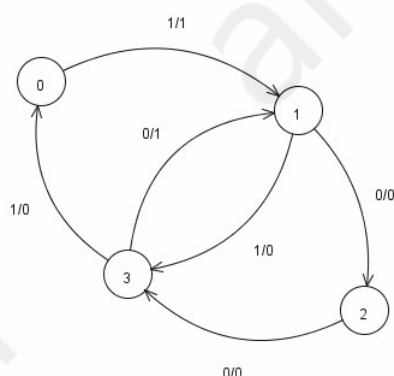
Άσκηση 3^η (15 μονάδες)

ερώτημα (α)

Μας ζητείται να υλοποιήσουμε την προδιαγραφή της εκφώνησης ως μηχανή Mealy. Οι μηχανές Mealy «κρυφοκοιτάζουν» την είσοδό τους και έτσι μπορούν ασύγχρονα να αλλάζουν την έξοδό τους, η οποία ουσιαστικά εξαρτάται τόσο από την παρούσα κατάσταση όσο και από την είσοδο.

Σύμφωνα με αυτό κι ακολουθώντας «πιστά» την εκφώνηση, η έξοδος θα γίνεται λογικό 1 μόνο όταν η είσοδος είναι 1 ενώ βρισκόμαστε στην κατάσταση 0, ή όταν η είσοδος είναι 0 ενώ βρισκόμαστε στην κατάσταση 3, σε όλες τις άλλες περιπτώσεις η έξοδος θα παίρνει 0.

Έτσι το Διάγραμμα Καταστάσεων που θα προκύψει θα είναι:



ερώτημα (β)

Πριν συμπληρώσουμε τον Πίνακα Αλήθειας-Καταστάσεων, θα πρέπει να σκεφτούμε ότι στην προδιαγραφή της εκφώνησης δεν αναφέρεται τι θα συνέβαινε όταν η είσοδος γίνει 1 ενώ βρισκόμαστε στην κατάσταση 2 ή όταν η είσοδος γίνει 0 ενώ βρισκόμαστε στην κατάσταση 0. Αυτές τις 2 μεταβάσεις λοιπόν (που απουσιάζουν από την προδιαγραφή αλλά και από το ΔΚ), θα τις θεωρήσουμε προς επόμενες καταστάσεις αδιάφορες (X X), ακόμα όμως και εδώ πρέπει εμείς να φροντίσουμε ώστε η έξοδος να παραμένει σε λογικό 0, αλλιώς θα παραβαίναμε την προδιαγραφή που απαιτεί λογικό 1 μόνο στις 2 αναφερθείσες περιπτώσεις του ερωτήματος (α).

παρούσα κατάσταση				επόμενη κατάσταση		είσοδοι flip-flops	
Q ₁	Q ₀	X	Y	Q' ₁	Q' ₀	D ₁	D ₀
0	0	0	0	X	X	X	X
0	0	1	1	0	1	0	1
0	1	0	0	1	0	1	0
0	1	1	0	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	X	X	X	X
1	1	0	1	0	1	0	1
1	1	1	0	0	0	0	0

Σημειώστε πάλι ότι χρησιμοποιώ την ' για το συμπλήρωμα και τον ' για την επόμενη κατάσταση

Έξυπνα και Εύκολα!

ερώτημα (γ)

		Q0, X			
		00	01	11	10
Q1	D1	x	0	1	1
	Q1	1	x	0	0

$$D_1 = Q_1 \bar{Q}_0 + \bar{Q}_1 Q_0 = Q_1 \oplus Q_0 \quad \text{που υλοποιείται με μία XOR}$$

		Q0, X			
		00	01	11	10
Q1	D0	x	1	1	0
	Q1	1	x	0	1

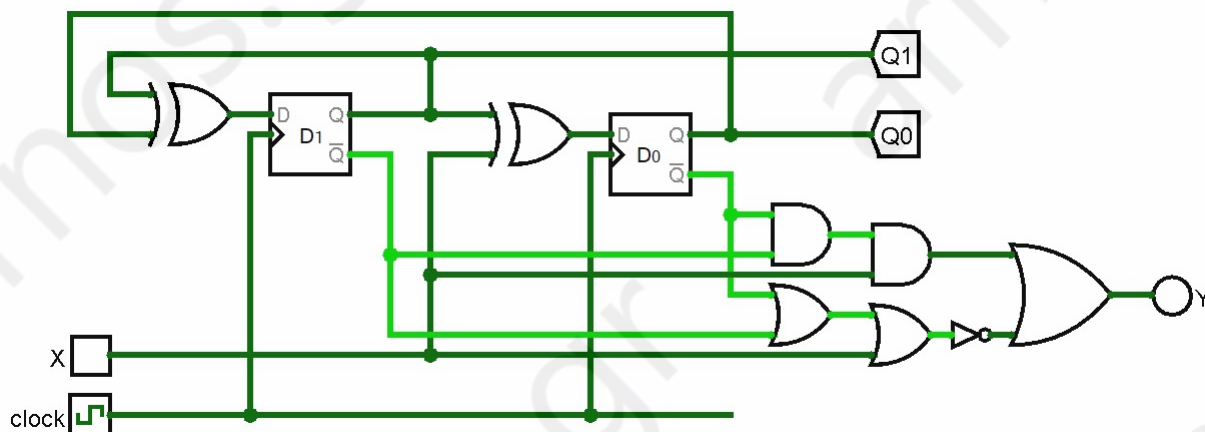
$$D_0 = Q_1 \bar{X} + \bar{Q}_1 X = Q_1 \oplus X \quad \text{που υλοποιείται ομοίως με μία XOR}$$

		Q0, X			
		00	01	11	10
Q1	Y	0	1	0	0
	Q1	0	0	0	1

$$\begin{aligned} D_1 &= \bar{Q}_1 \bar{Q}_0 X + Q_1 Q_0 \bar{X} = (\bar{Q}_1 \bar{Q}_0) X + (Q_1 Q_0) \bar{X} = (\bar{Q}_1 \bar{Q}_0) X + \overline{(\bar{Q}_1 \bar{Q}_0)} \bar{X} \\ &= (\bar{Q}_1 \bar{Q}_0) X + \overline{(\bar{Q}_1 \bar{Q}_0)} + X = (\bar{Q}_1 \bar{Q}_0) X + (\bar{Q}_1 + \bar{Q}_0) + X \end{aligned}$$

που θα την υλοποιήσουμε με 2 πύλες AND, 3 πύλες OR και μία NOT

Έτσι θα προκύψει το ζητούμενο κύκλωμα, απόλυτα συνεπές με την προδιαγραφή όσον αφορά τη λειτουργία και την υλοποίηση, με 7 πύλες AND, OR, XOR μόνο 2 εισόδων και μία NOT:



Έξυπνα και Εύκολα!

Άσκηση 4^η (15 μονάδες)

ερώτημα (α)

Πρόκειται για μηχανή Moore, η έξοδος του κυκλώματος σημειώνεται εντός των καταστάσεων δηλαδή η έξοδος δεν εξαρτάται από την είσοδο αλλά μόνο από την παρούσα κατάσταση. Από το ΔΚ παρατηρούμε ότι για $X=0$ διαγράφει τον κύκλο 0-1-3-2-0... ενώ για $X=1$ 0-3-1-2-1-2...

ερώτημα (β)

state	παρούσα κατάσταση				είσοδοι flip-flops				επόμενη κατάσταση		state
	Q_1	Q_0	X	Y	R_1	S_1	R_0	S_0	Q'_1	Q'_0	
0	0	0	0	0	0	X	1	0	0	1	1
0	0	0	1	0	1	0	1	0	1	1	3
1	0	1	0	1	1	0	X	0	1	1	3
1	0	1	1	1	1	0	0	1	1	0	2
2	1	0	0	1	0	1	0	X	0	0	0
2	1	0	1	1	0	1	1	0	0	1	1
3	1	1	0	0	X	0	0	1	1	0	2
3	1	1	1	0	0	1	X	0	0	0	0

ερώτημα (γ)

		Q_0, X			
Q_1	Y	00	01	11	10
	0	0	0	1	1
	1	1	1	0	0

$$Y = Q_1 \bar{Q}_0 + \bar{Q}_1 Q_0 = Q_1 \oplus Q_0$$

		Q_0, X			
Q_1	R_1	00	01	11	10
	0	x	0	0	0
	1	1	1	1	0

$$R_1 = Q_1 \bar{Q}_0 + Q_1 X = Q_1 (\bar{Q}_0 + X)$$

		Q_0, X			
Q_1	S_1	00	01	11	10
	0	0	1	1	1
	1	0	0	0	x

$$S_1 = \bar{Q}_1 X + \bar{Q}_1 Q_0 = \bar{Q}_1 (Q_0 + X)$$

		Q_0, X			
Q_1	R_0	00	01	11	10
	0	0	0	1	0
	1	x	0	0	1

$$R_0 = Q_1 \bar{X} + \bar{Q}_1 Q_0 X = Q_1 Q_0 \bar{X} + \bar{Q}_1 Q_0 X = Q_0 (Q_1 \bar{X} + \bar{Q}_1 X) = Q_0 (Q_1 \oplus X)$$

		Q_0, X			
Q_1	S_0	00	01	11	10
	0	1	1	0	x
	1	0	1	x	0

$$S_0 = \bar{Q}_1 \bar{Q}_0 + \bar{Q}_0 X = \bar{Q}_0 (\bar{Q}_1 + X)$$

Έξυπνα και εύκολα!

Άσκηση 5^η (20 μονάδες)
ερώτημα (α)

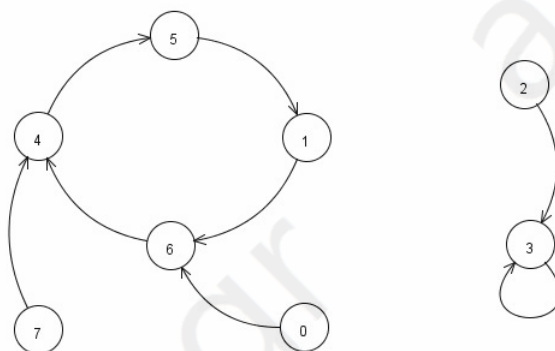
$$\underline{J = \overline{Q}_B}, \underline{K = \overline{Q}_B Q_C}, \underline{R = Q_A}, \underline{S = \overline{Q}_A} \text{ και } \underline{D = Q_A \oplus Q_B = Q_A \overline{Q}_B + \overline{Q}_A Q_B}$$

Υπενθυμίζω ότι χρησιμοποιώ την $\overline{}$ για το συμπλήρωμα και τον $'$ για την επόμενη κατάσταση

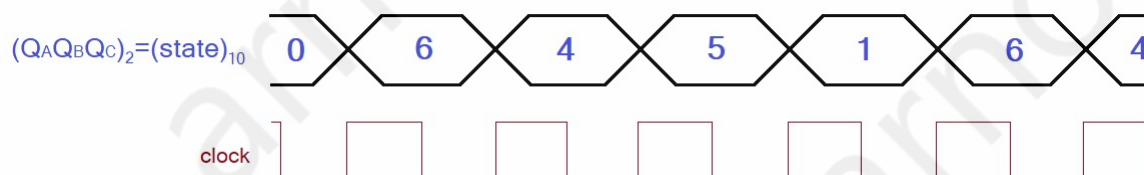
ερώτημα (β)

state	παρούσα κατάσταση			είσοδοι flip-flops					επόμενη κατάσταση			state
	Q_A	Q_B	Q_C	J	K	R	S	D	Q'_A	Q'_B	Q'_C	
0	0	0	0	1	0	0	1	0	1	1	0	6
1	0	0	1	1	1	0	1	0	1	1	0	6
2	0	1	0	0	0	0	1	1	0	1	1	3
3	0	1	1	0	0	0	1	1	0	1	1	3
4	1	0	0	1	0	1	0	1	1	0	1	5
5	1	0	1	1	1	1	0	1	0	0	1	1
6	1	1	0	0	0	1	0	0	1	0	0	4
7	1	1	1	0	0	1	0	0	1	0	0	4

Παρατηρείστε την αποκωδικοποιημένη παρούσα και επόμενη κατάσταση του κυκλώματος εκατέρωθεν του ΠΑ-Καταστάσεων για ευκολότερο σχεδιασμό του Διαγράμματος Καταστάσεων:


ερώτημα (γ)

Από το κύκλωμα παρατηρούμε ότι χρησιμοποιήθηκαν rising edge Flip-Flops, οπότε:

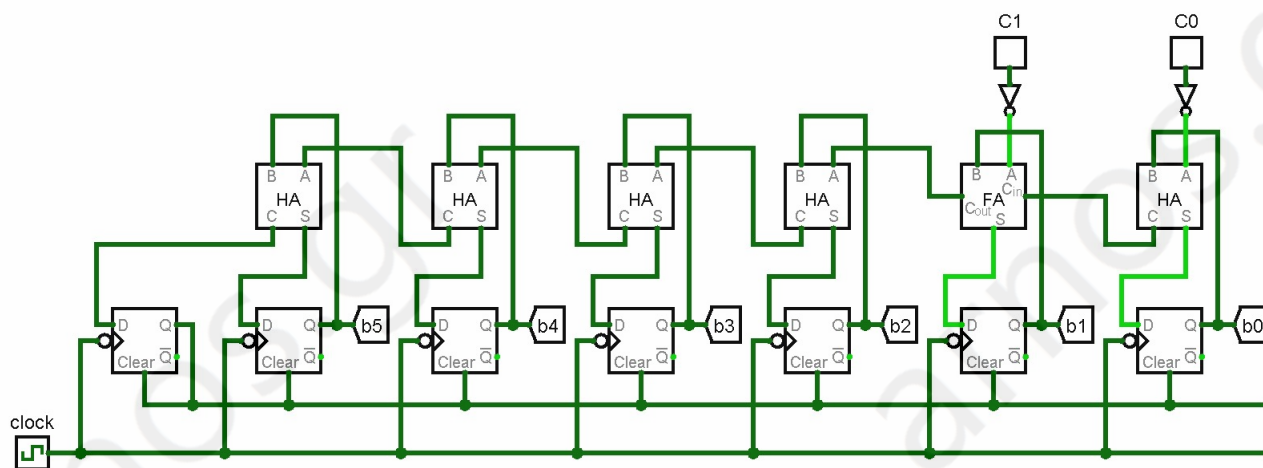


Έξυπνα και Εύκολα!

Άσκηση 6^η (20 μονάδες)

C_1C_0	αύξηση μετρητή	$\overline{C_1}\overline{C_0}$
00	3	11
01	2	10
10	1	01
11	0	00

Παρατηρούμε ότι η επόμενη κατάσταση μέτρησης προκύπτει αθροίζοντας στην παρούσα κατάσταση τον διψήφιο δυαδικό C_1C_0 συμπληρωμένο ως προς 1, δηλαδή τον $\overline{C_1}\overline{C_0}$. Θα υλοποιήσουμε λοιπόν έναν τυπικό καταχωρητή 6 ψηφίων με D flip-flops (εφοδιασμένα με ασύγχρονη είσοδο reset), στον οποίο θα προσθέτουμε τον διψήφιο $\overline{C_1}\overline{C_0}$ σε κάθε tick του ρολογιού, μέσω ενός αθροιστή διάδοσης κρατούμενου που θα υλοποιηθεί με 6 FAs. Στην έξοδο $b_5b_4b_3b_2b_1b_0$ του κυκλώματος θα διαβάζουμε την τρέχουσα κατάσταση μέτρησης.



Στην πραγματικότητα θα χρειαστούν μόνο 5 HAs και 1 FA, καθώς στα 4 Most Significant ψηφία $b_5b_4b_3b_2$ του μετρητή προστίθεται πάντα το 0 και στο b_0 υπάρχει 0 ως κρατούμενο εισόδου, επομένως οι FAs αντικαθίστανται με HAs και ο FA χρησιμοποιείται μόνο στο b_1 .

ΠΡΟΣΕΧΤΕ ΟΜΩΣ ΠΑΡΑ ΠΟΛΥ, διότι για την ενεργοποίηση του ασύγχρονου reset στα FFs θα πρέπει να χρησιμοποιήσουμε ένα επιπλέον D flip-flop προκειμένου να κάνουμε σύγχρονα το reset στα bits του μετρητή, διαφορετικά ενδέχεται να χάνουμε μετρήσεις στο όριο υπερχείλισης του μετρητή. Ένα εύκολο παράδειγμα για να καταλάβετε το πρόβλημα: Έστω ότι $b_5b_4b_3b_2b_1b_0 = (111100)_2 = (60)_{10}$ και $C_1C_0 = 00$, στο επόμενο tick του ρολογιού ο μετρητής θα πάει στιγμιαία στο +3 άρα $(63)_{10}$ αλλά ο αθροιστής στο $(2)_{10}$ δημιουργώντας κρατούμενο, αν χρησιμοποιούσαμε τούτο το κρατούμενο για reset δεν θα προλάβουμε καν να δούμε αυτό το $(63)_{10}$! Χρησιμοποιώντας ένα 7^ο «κρυφό» FF, θα πραγματοποιήσουμε το reset εκεί ακριβώς που πρέπει δηλαδή όταν εσωτερικά $Q_6b_5b_4b_3b_2b_1b_0 \geq (1000000)_2 = (64)_{10}$.

Έξυπνα και Εύκολα!