

Ψηφιακή Σχεδίαση

Διάλεξη 6 – Διαγράμματα Καταστάσεων (FSMs), Μνήμες Μόνο Ανάγνωσης (ROMs)

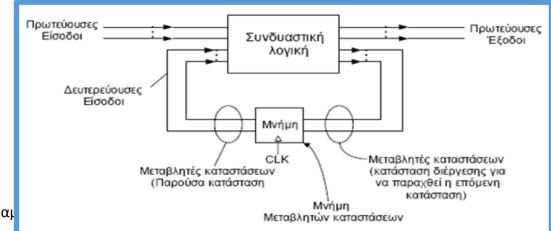
Γεώργιος Κεραμίδας, Επίκουρος Καθηγητής 2° Εξάμηνο, Τμήμα Πληροφορικής



Εισαγωγή στο σχεδιασμό ψηφιακών κυκλωμάτω με διαγράμματα καταστάσεων

- Χρησιμοποιούνται για την σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων
- Γενική μορφή ενός σύγχρονου ακολουθιακού κυκλώματος που χρησιμοποιείται σε διάγραμμα καταστάσεων

• Ένα σύστημα με Ν μεταβλητές χρειάζεται Ν flip-flops για μνήμη



18 May 2021

Γεώργιος Κεραι

Παράδειγμα



• Να σχεδιασθεί ψηφιακό σύστημα το οποίο να επιτρέπει την εκκίνηση της μηχανής του αυτοκινήτου μόνον εφόσον ο οδηγός καθίσει και δέσει την ζώνη ασφαλείας του καθίσματός του.

• Λάθος λύση: Μια πύλη AND από ένα sensor στη ζώνη και έναν

στο κάθισμα του αυτοκινήτου





Διαγράμματα Καταστάσεων



• Είναι η γραφική παράσταση της λειτουργίας ενός ακολουθιακού κυκλώματος

Στοιχείο Κατάστασης:



S_i: περιγραφή της κατάστασης i

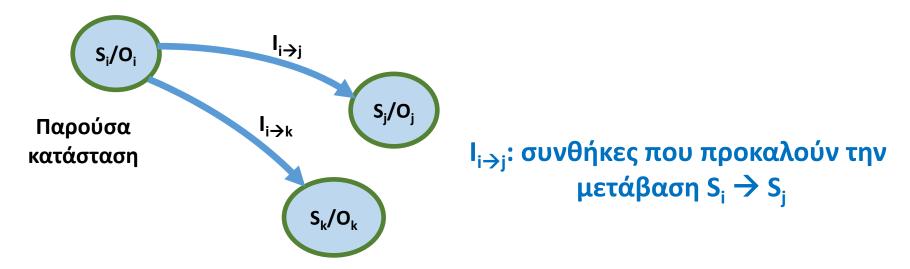
Ο_i: έξοδοι που σχετίζονται με την κατάσταση S_i

• Η παραπάνω γραφική αναπαράσταση των διαγραμμάτων είναι γνωστή και ως παράσταση Moore

Διαγράμματα Καταστάσεων



Μετάβαση Καταστάσεων (Διάγραμμα Καταστάσεων):



• Η παραπάνω γραφική αναπαράσταση των διαγραμμάτων είναι γνωστή και ως παράσταση Moore

Πίνακες Καταστάσεων



Είναι η παράσταση του διαγράμματος καταστάσεων υπό μορφή πίνακα. Σε αυτόν περιλαμβάνονται η παρούσα κατάσταση (PS), η επόμενη κατάσταση (NS), οι συνθήκες εισόδου που προκαλούν τη μετάβαση και οι αντίστοιχες τιμές εξόδου

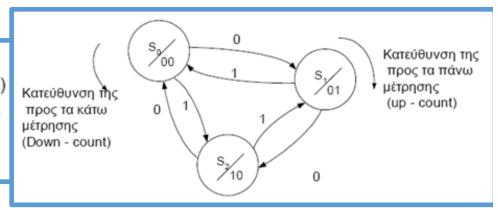
• Παράδειγμα:

Μετρητής MOD-3 (UP/DOWN)

είσοδοι: DIR = 1 : Μετράει μπρος τα κάτω (Down)

DIR = 0 : Μετράει μπρος τα επάνω (Up)

έξοδοι: Α,Β



Πίνακες Διέγερσης



- Πίνακας διέγερσης: έχει τις εισόδους για κάθε μετάβαση για ένα τύπο flip-flop
- Πίνακας-Διάγραμμα Καταστάσεων \rightarrow ανεξάρτητα του Flip-flop
- Όταν αποφασιστεί ο τύπος του flip-flop, κατασκευάζεται ο πίνακας διέγερσης

Είσοδοι (DIR)	PS	MS	Έξο Α	δοι Β
0	S ₀	S ₁	0	0
1	S ₁	S ₂	0	0
0	S ₁	S ₀	0	1
1	S ₂	S ₀	0	1
0	S ₂	S ₁	1	0

1η Μορφή

PS	NS DIR=0	NS DIR=1	Έξο Α	δοι Β
	Dire o	J		
S	S ₁	S ₂	0	0
S₁	S ₂	S ₀	0	1
91	02	0	•	'
S ₂	S_0	S ₁	1	0

2η Μορφή

18 May 2021

Γεώργι

Επιλογή Flip-Flop



• Συνήθως χρησιμοποιούμε είτε D-FFs, είτε JK-FFs

D-ffs

Απλούστερη επεξεργασία Κατά το σχεδιασμό JK-ffs

Λιγότερο σύνθετο τελικό κύκλωμα

Πρώτο Παράδειγμα



Να σχεδιαστεί ένα ακολουθιακό κύκλωμα που ανιχνεύει την ακολουθία 1 0 1 από ένα σύνολο δυαδικών στοιχείων που εισάγονται σειριακά με ρυθμό 1 bit ανά παλμό ρολογιού (να χρησιμοποιηθούν JK-FFs)

Πρώτο Παράδειγμα



Να σχεδιαστεί ένα ακολουθιακό κύκλωμα που ανιχνεύει την ακολουθία 1 0 1 από ένα σύνολο δυαδικών στοιχείων που εισάγονται σειριακά με ρυθμό 1 bit ανά παλμό ρολογιού (να χρησιμοποιηθούν JK-FFs)



Προσοχή: το 1-0-1-0-1 \rightarrow Είναι 1 ή 2 YESes ?

Λύση

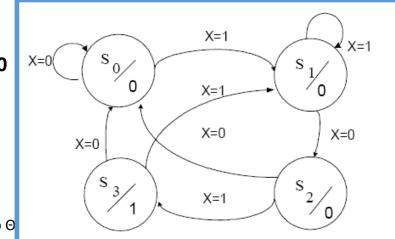


- Πρόβλημα: Δεν είναι φανερό πόσες είναι οι δυνατές
 καταστάσεις που θα βάλουμε στο διάγραμμα καταστάσεων
 - Πχ. η απάντηση 3 δεν είναι σωστή
- Έστω S₀ η αρχική κατάσταση κατά την οποία κανένα από τα bit εισόδου δεν έχουν φτάσει στην σωστή ακολουθία
- Στην S_0 αν X=0 μένουμε στην S_0 Αν X=1 τότε έχουμε το πρώτο bit της ακολουθίας και πάμε στην S_1
- Η έξοδος και στις δύο περιπτώσεις είναι μηδέν εφόσον δεν έχει ανιχνευτεί ακόμη η πλήρης ακολουθία

Λύση – Διάγραμμα Καταστάσεων



- Στην S_1 για X=1 μένουμε στην S_1 (θεωρούμε ότι το νέο 1 είναι το πρώτο bit της ακολουθίας)
- Στην S_1 για X=0 πηγαίνουμε στην S_2
- Στην S_2 για X=1 πηγαίνουμε στην S_3 (εδώ η έξοδος είναι Z=1)
- Στην S₂ για X=0 πηγαίνουμε στην S₀
- Στην S_3 για X=0 επιστρέφουμε στην S_0
- Στην S_3 για X=1 πηγαίνουμε στην S_1



18 May 2021

Γεώργιος Κεραμίδας / Αριστοτέλειο Πανεπιστήμιο Θ

Λύση – Πίνακας Καταστάσεων



• Διάγραμμα καταστάσεων \rightarrow 4 καταστάσεις \rightarrow 2 μεταβλητές \rightarrow 2 bits \rightarrow 2 FFs

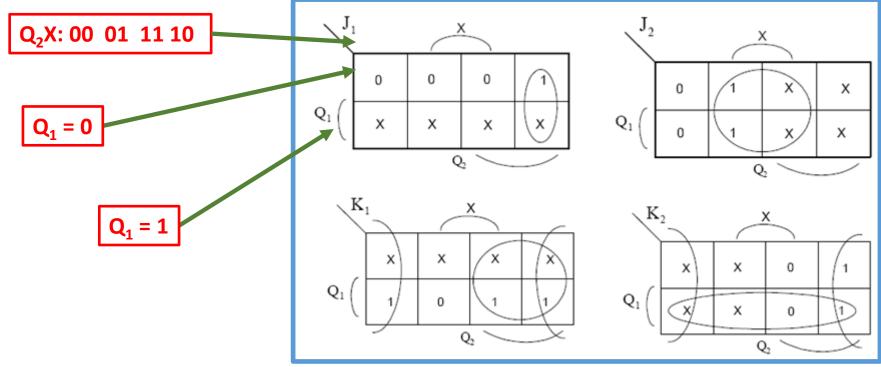
Λύση – Πίνακας Διέγερσης



		PS	NS X	= 0 EX	NS X		
		$Q_1 Q_2$	$Q_1 Q_2$	$J_1 K_1 J_2 K_2$	Q_1Q_2	$J_1 K_1 J_2 K_2$	Z
S	, o	0 0	$S_0:0 \ 0$	$0 X_{1}^{1} 0 X$	S ₁ : 0 1	$\begin{bmatrix} 0 & X & 1 & X \end{bmatrix}$	0
S	1	0 1	$S_2:1 0$	1 X X 1	S_1 : 0 1	0 X 1 0	0
	2	1 0	$S_0:0 \ 0$	$X \stackrel{!}{1} 0 X$	S_3 : 1 1	X 0 1 X	0
S	3	1 1	$S_0:0$ 0	$X1_{1}^{\dagger}X1$	S_1 : 0 1	X1 X 0	1

Λύση – Εξισώσεις Διέγερσης των JK-FFs

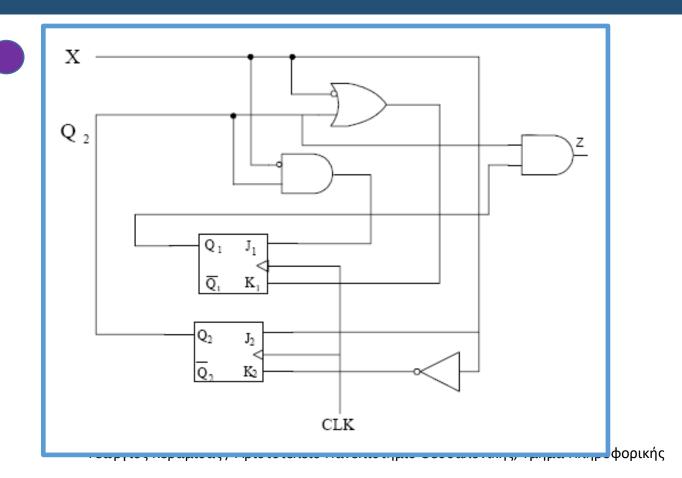




$$J_1 = \overline{X} Q_2$$
, $K_1 = \overline{X} + Q_2$, $J_2 = X$, $K_2 = \overline{X}$ kat $Z = Q_1 Q_2$

Λύση – Τελικό Κύκλωμα





18 May 2021

16

Προβλήματα από καταστάσεις που δεν χρησιμοποιούνται



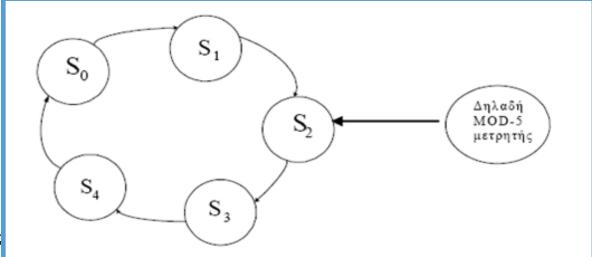
- Ένα ακολουθιακό κύκλωμα μπορεί να ξεκινήσει από μια κατάσταση που δεν χρησιμοποιείται ή να "μπει" σε μια τέτοια κατάσταση λόγω θορύβου
- Η συμπεριφορά του κυκλώματος δεν είναι πλέον προβλέψιμη

• Παράδειγμα (MOD-5):

3 καταστάσεις που δεν χρησιμοποιούνται. Μπορεί να οδηγήσουν το κύκλωμα να "κλειδώσει"

18 May 2021

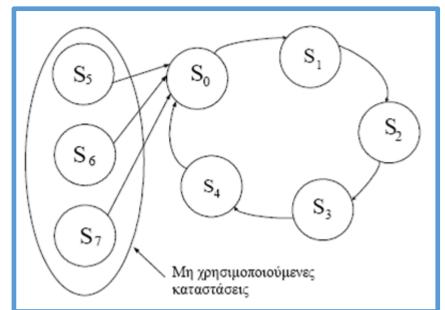
Γεώργιος Κεραμίδας



Προβλήματα από καταστάσεις που δεν χρησιμοποιούνται



- Βάζουμε στο διάγραμμα καταστάσεων και τις καταστάσεις που δεν χρησιμοποιούνται
- → Λογική επαναφοράς στην αρχική κατάσταση (Reset circuitry)
- Λύνει την περίπτωση που οι καταστάσεις 'παγίδα' εμφανιστούν π.χ. πριν την έναρξη του ρολογιού

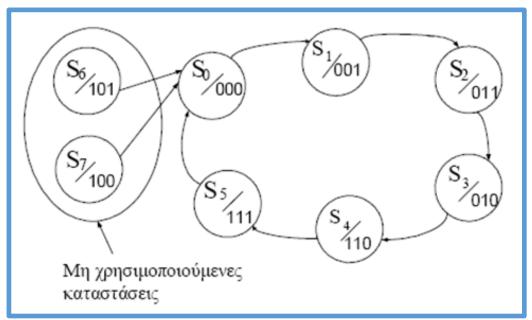


Δεύτερο Παράδειγμα



• Να σχεδιαστεί μετρητής MOD-6 Gray Code με το ακόλουθο διάγραμμα καταστάσεων

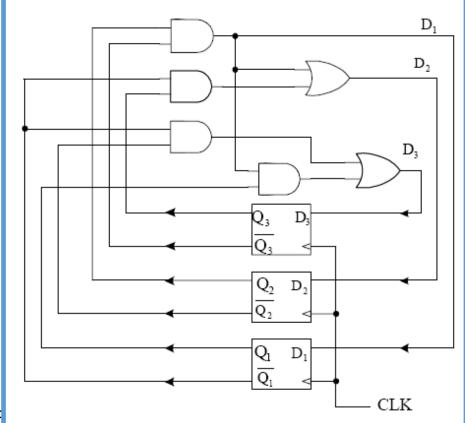
Θεωρούμε ότι οι καταστάσεις PS είναι ίδιες με τις καταστάσεις εξόδου



Λύση – Πίνακας Καταστάσεων & Τελικό Κύκλωμ 🔘

PS	PS+Έξοδοι				NS+Καταστάσεις διέγερσης				
	Q_1	Q_2	Q_3	018	ης D ₂	D_3			
S ₀ :	0	0	0	S 1 :	0	0	1		
S ₁ :	0	0	1	S 2 :	0	1	1		
S ₂ :	0	1	1	S 3 :	0	1	0		
S ₃ :	0	1	0	S 4 :	1	1	0		
S4:	1	1	0	S 5 :	1	1	1		
S ₅ :	1	1	1	S 0 :	0	0	0		
S ₆ :	1	0	1	S 0 :	0	0	0		
S ₇ :	1	0	0	S 0 :	0	0	0		

Τελικό κύκλωμα



18 May 2021

Γεώργιος Κεραμίδας / Αριστοτέλεις

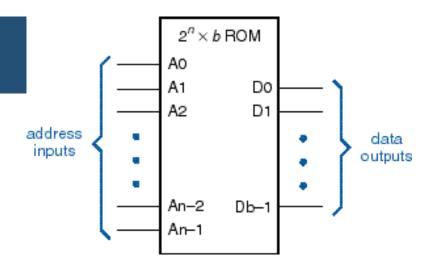
Διατάξεις ROM



- Η ROM είναι ένα προγραμματιζόμενο ολοκληρωμένο.
- Ο προγραμματισμός μπορεί να γίνει:
 - Μόνο μια φορά:
 - Στο εργοστάσιο (PROM)
 - Από το χρήστη (FPROM)
 - Περισσότερες φορές με σβήσιμο των ήδη υπαρχόντων δεδομένων, με διάφορους τρόπους :
 - Με ακτινοβολία (EPROM)
 - Με ηλεκτρικές διατάξεις (EEPROM E²PROM)
 - Τμηματικά
 - Flash
 - Paged flash

Βασικές δομές ROM

- Συνδυαστικό κύκλωμα Ν εισόδων και Β εξόδων
 - Είσοδοι = Διευθύνσεις (n-bits)
 - Εξοδοι = δεδομένα (b-bits)
- Η μνήμη ROM "αποθηκεύει" τον πίνακα αληθείας οποιασδήποτε συνάρτησης με εισόδους =< είσοδοι της ROM και εξόδους =< έξοδοι της ROM.
- Ο διπλανός πίνακας αληθείας μπορεί να αποθηκευτεί σε μια ROM (8x4)
- Η ROM τελικά είναι μνήμη ή συνδυαστικό κύκλωμα ?
- H ROM είναι non-volatile!



	Inputs		Outputs				
A2	A1	A0	D3	D2	D1	D0	
0	0	0	1	1	1	0	
0	0	1	1	1	0	1	
0	1	0	1	0	1	1	
0	1	1	0	1	1	1	
1	0	0	0	0	0	1	
1	0	1	0	0	1	0	
1	1	0	0	1	0	0	
1	1	1	1	0	0	0	

Υλοποίηση συναρτήσεων με μνήμη



- $f_1(x_1, x_2, x_3, ..., x_m)$, $f_2(x_1, x_2, x_3, ..., x_m)$, ..., $f_n(x_1, x_2, x_3, ..., x_m)$
- Η υπόθεση ότι κάθε συνάρτηση είναι των ίδιων μεταβλητών είναι απόλυτα λογική καθώς οποιαδήποτε λιγότερων μπορεί να μετατραπεί σε συνάρτηση m μεταβλητών, όπου οι επιπλέον είναι

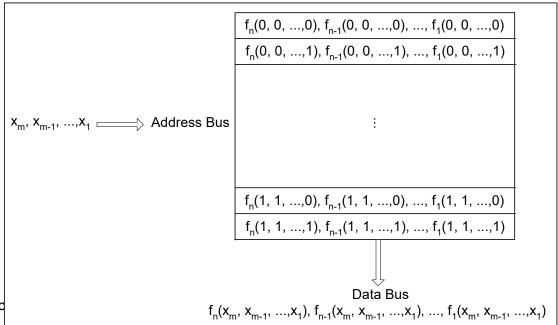
αδιάφοροι.

•
$$f(a,b,c) = ab + c$$

$$= (ab+c)$$

$$= (ab+c) (d + d')$$

$$=$$
 abd + cd + abd' + cd'



18 May 2021

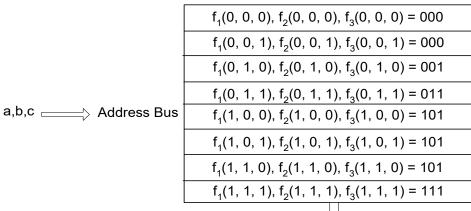
Γεώργιος Κεραμίδας / Αρισ

Παράδειγμα υλοποίησης συναρτήσεων με μνήμ

- f1(a) = a, f2(b, c) = bc, f3(a, b) = a + b
- Αριθμός μεταβλητών m = 3
- f1(a, b, c) = abc + abc' + a'bc + a'b'c,
- f2(a, b, c) = abc + a'bc
- f3(a, b, c) = abc + abc' + ab'c + ab'c' + a'bc + a'bc'

Στόχος: Να έχω αποθηκευμένα τα αποτελέσματα των συναρτήσεων και να τα χρησιμοποιώ όποτε θέλω χωρίς να χρειάζεται να κάνω κάθε φορά λογικές πράξεις

а	b	С	f1	f2	f3	
0	0	0	0	0	0	
0	0	1	0	0	0	
0	1	0	0	0	1	
0	1	1	0	1	1	
1	0	0	1	0	1	
1	0	1	1	0	1	
1	1	0	1	0	1	
1	1	1	1	1	1	
18 May 2021			ι εωρι	γιος Κεραμ	ιδας / Αρισ	τοτέλειο



Data Bus $f_1(a, b, c), f_2(a, b, c), f_3(a, b, c)$

Πολλαπλασιασμός με Αποθηκευμένες Τιμές



• Ονομάζονται Look up tables (ROMs)

	000	001	010	011	100	101	110	111
000	000000	000000	000000	000000	000000	000000	000000	000000
001	000000	000001	000010	000011	000100	000101	000110	000111
010	000000	000010	000100	000110	001000	001010	001100	001110
011	000000	000011	000110	001001	001100	001111	010010	010101
100	000000	000100	001000	001100	010000	010100	011000	011100
101	000000	000101	001010	001111	010100	011001	011110	100011
110	000000	000110	001100	010010	011000	011110	100100	101010
111	000000	000111	001110	010101	011100	100011	101010	110001

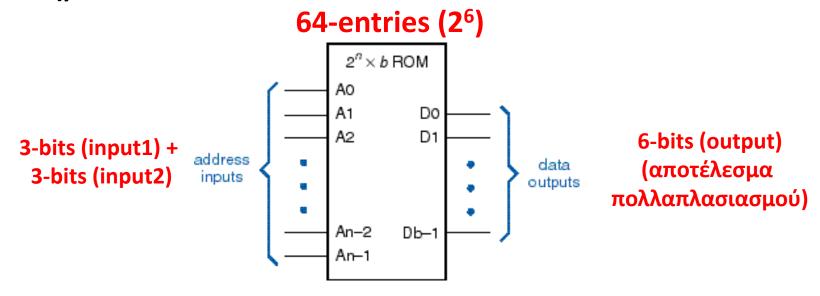
18 May 2021

Γεώργιος Κεραμίδας / Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Πληροφορικής

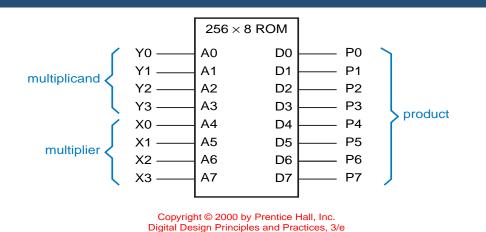
Πολλαπλασιασμός με Αποθηκευμένες Τιμές

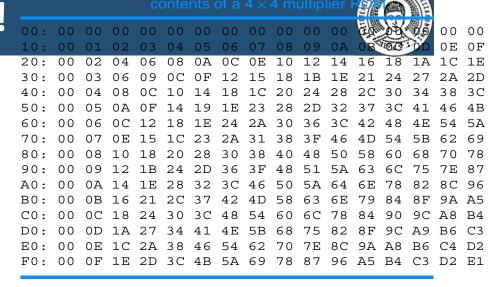


• Ονομάζονται Look up tables (Μικρή καθυστέρηση, Εκθετικό μέγεθος)



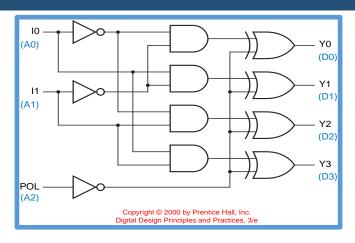
Πολλαπλασιασμός με ROM!

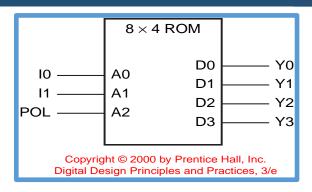




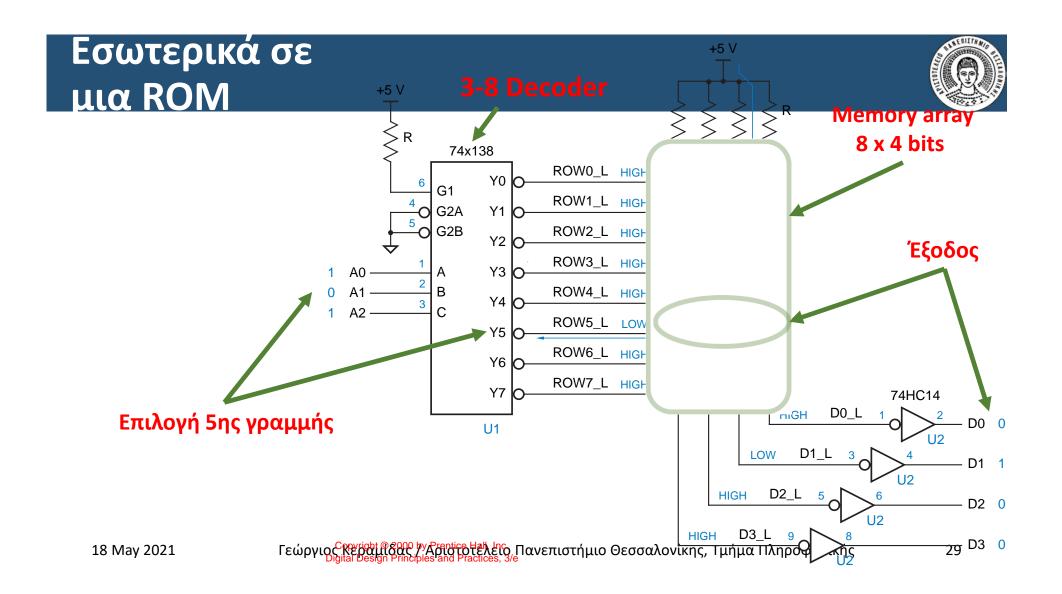
- Πολλαπλασιασμός 2 x 4-bit αριθμών -> ROM 256x8
- Διεύθυνση -> συνένωση πολλαπλασιαστή & ολλαπλασιαστέου
- Αποτέλεσμα → 8-bits (έξοδος δεδομένων της ROM)
- Θέση {Y, X} στη ROM → αποθηκευμένη η τιμή Y x X.
- Ο προγραμματισμός γίνεται με ειδική συσκευή στην οποία δίνουμε τα δεδομένα με τη μορφή πίνακα.

Οι δύο υλοποιήσεις ενός συνδυαστικού κυκλώμ

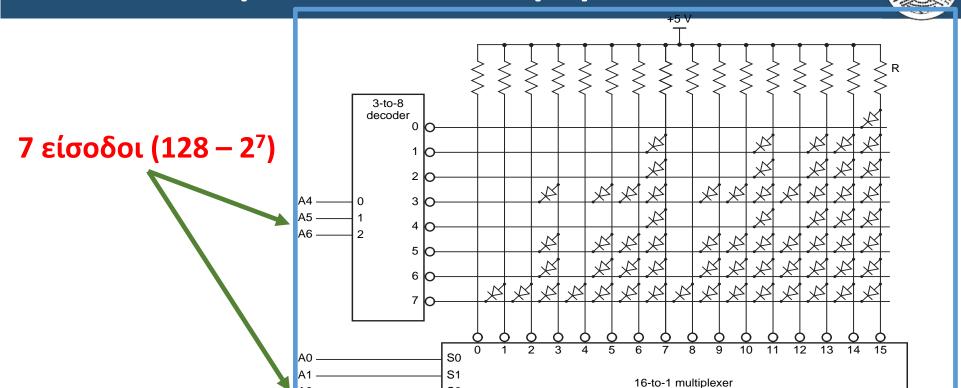




- Οι δύο υλοποιήσεις είναι ισοδύναμες μόνο από λογικής πλευράς
- Από χρονικής πλευράς η ROM είναι πιο αργή, αλλά παρέχει τον ίδιο χρόνο απόκρισης ανεξάρτητα των εισόδων POL, I1 και I0
- Για κύκλωμα 4 μεταβλητών χρειαζόμαστε ROM 16 x Y. Για ένα 20 μεταβλητών 1M x Y !!!
- Υλοποίηση με ROM συμφέρει :
 - Για μικρά κυκλώματα λίγων εισόδων
 - Για να κρύψουμε τη πραγματική υλοποίηση



Δισδιάστατη Αποκωδικοποίηση: 128 x 1 ROM



S2 S3

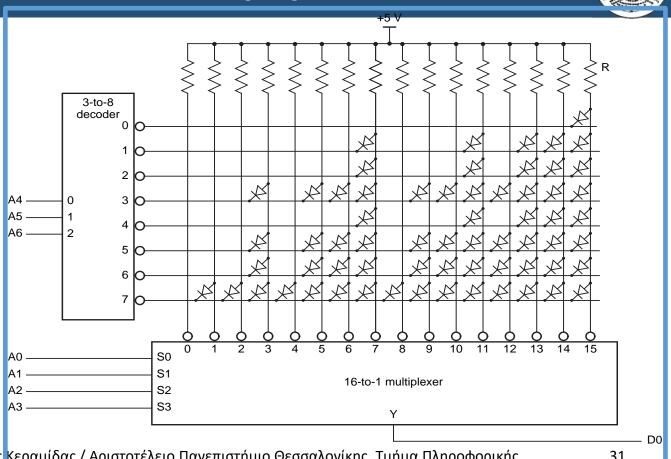
18 May 2021

Γεώργιος Κεραμίδας / Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Πληροφορικής 30 Cdpyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Δισδιάστατη Αποκωδικοποίηση : 128 x 1 ROM



Υπάρχουν και άλλες επιλογές Π.χ. χρήση 4-16 decoder και 8-1 multiplexer



18 May 2021

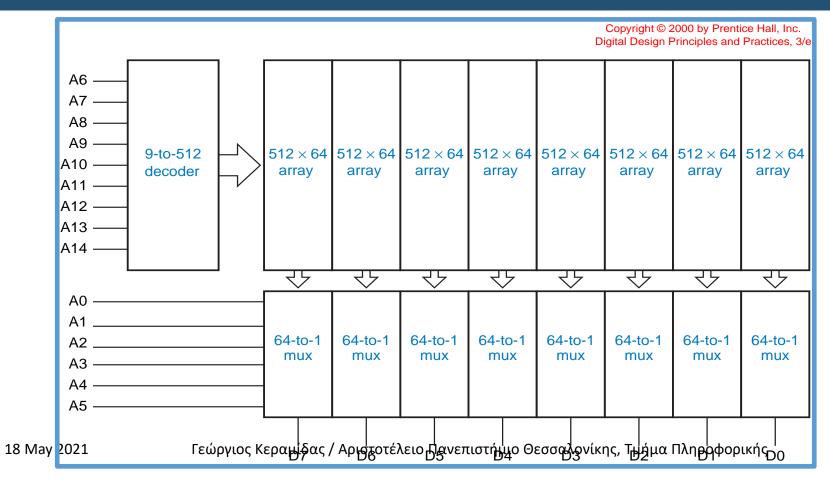
Γεώργιος Κεραμίδας / Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Πληροφορικής

ight © 2000 by Prentice Hall, Inc.

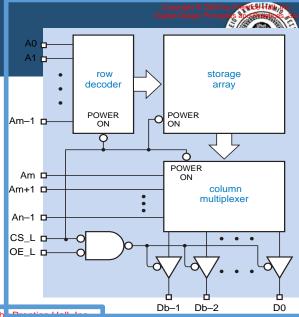
Digital Design Principles and Practices, 3/e

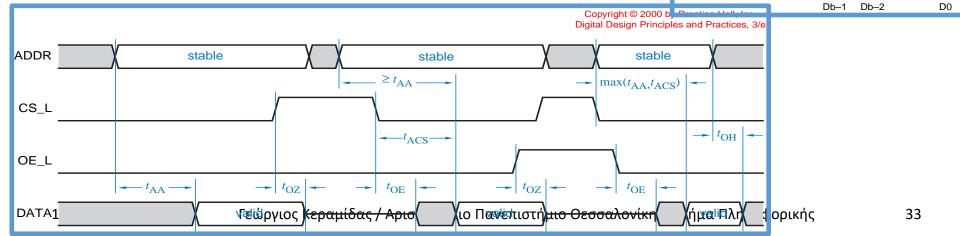
Πιθανή διάταξη μιας 32K x 8 ROM





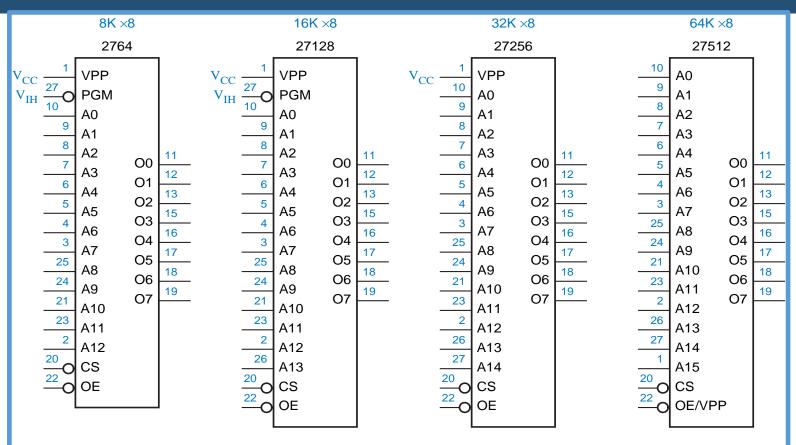
Γενική αρχιτεκτονική και κρίσιμοι χρόνοι μιας ROM





LSI EPROMs





18 May 2021

Γεώργιος Κεραμίδας / Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης Τμήμος Πληροφορμκής Inc.

Digital Design Principles and Practices, 3

LSI EPROMs

4 chips 32K x 8 ROMs



Μικροεπεξεργαστής -

Address Bus: 20-bits

Συνολική Μνήμη: 1Mbyte

Διαθέσιμη Μνήμη: 128Κ

Για την διαθέσιμη Μνήμη: 17-bits

- 15-bits για κάθε 32K (chip)

- 2-bits για επιλογή chip

3-bits από το address bus δεν χρησιμοποιούνται

Συνέχεια στο μάθημα Μικροεπεξεργαστες (7ο εξάμηνο)

