



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ

ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

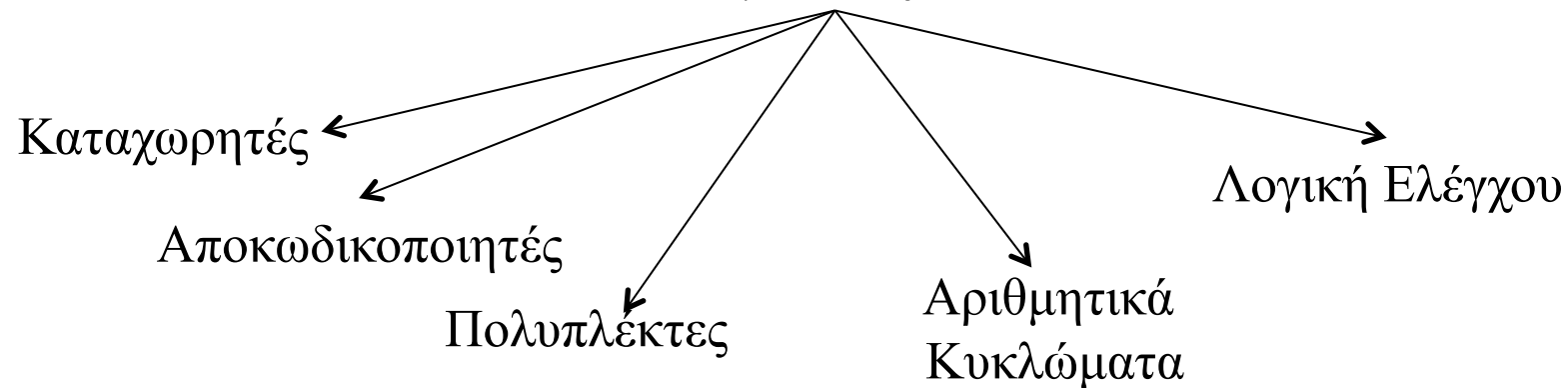
6^η Θεματική Ενότητα : Σχεδίαση Συστημάτων σε Επίπεδο Καταχωρητή

Εισαγωγή

Η σχεδίαση ενός ψηφιακού συστήματος
ως ακολουθιακή μηχανή είναι εξαιρετικά
δύσκολη

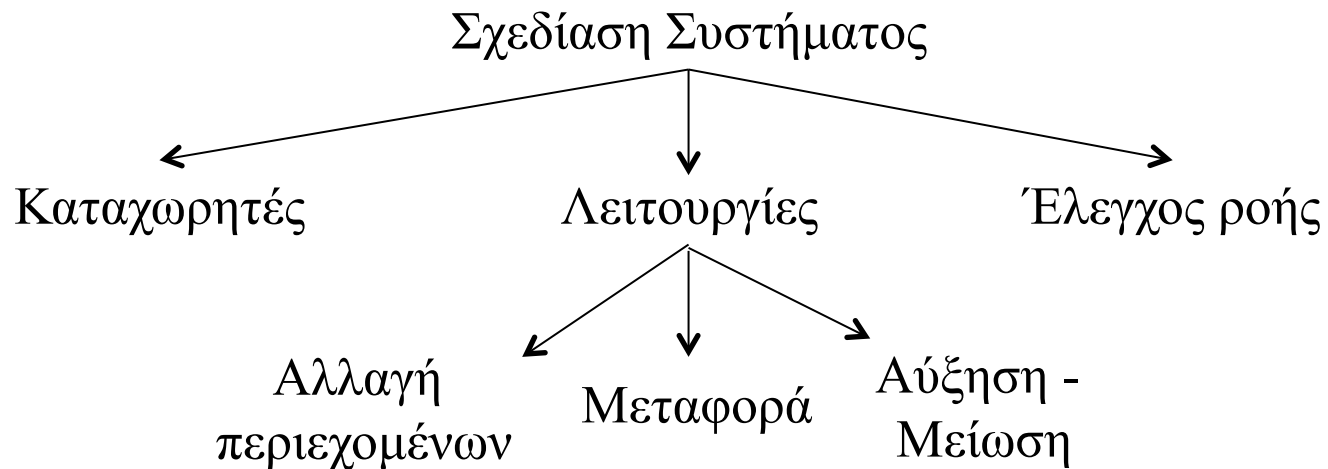
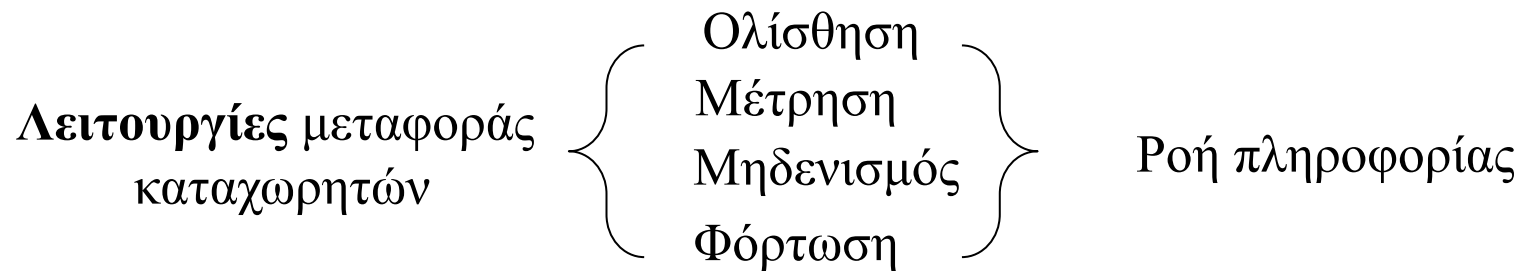


Τμηματοποίηση σε
υποσυστήματα με δομικές
μονάδες:



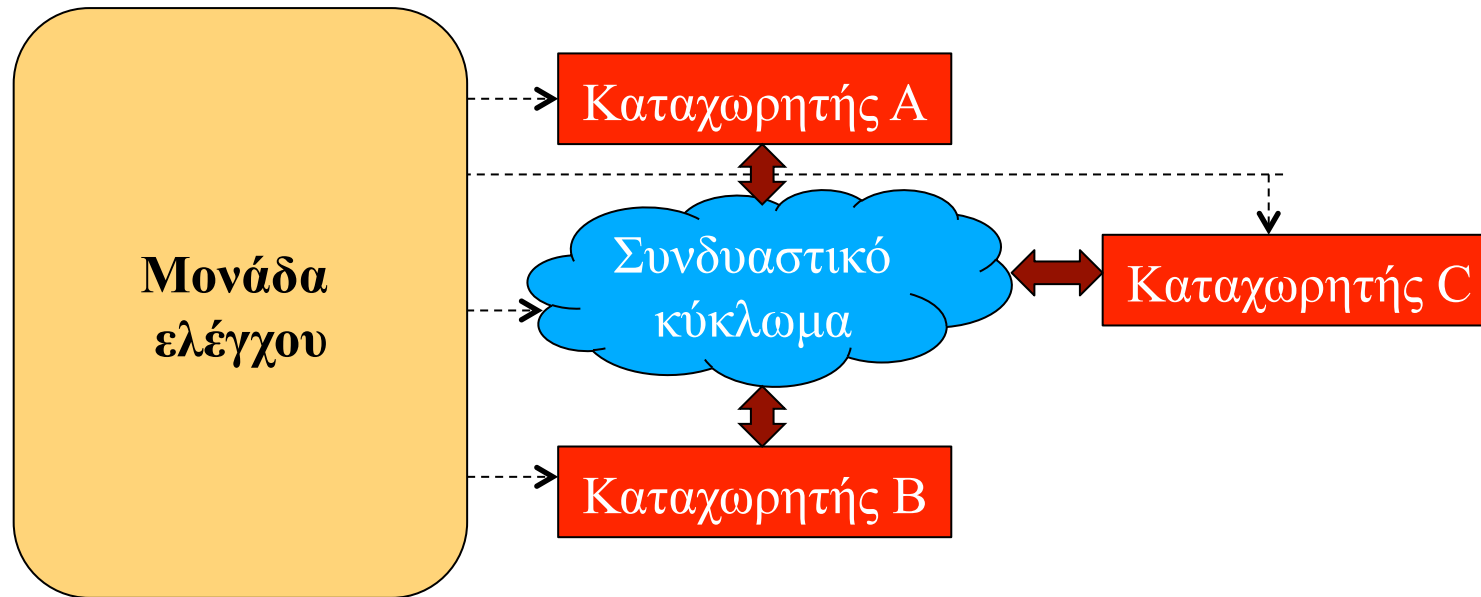
Σημειογραφία

Οι καταχωρητές θεωρούνται οι βασικές δομικές μονάδες ενός συστήματος σε επίπεδο μεταφοράς καταχωρητών RTL



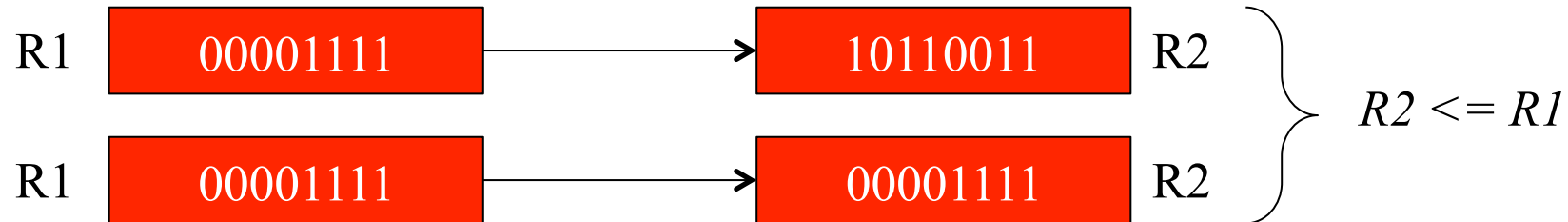
Λειτουργίες

Οι λειτουργίες ελέγχονται από **σήματα ελέγχου** που καθορίζουν την ακριβή **ακολουθία** τους και που **εξαρτώνται** από προηγούμενα αποτελέσματα



Η μονάδα ελέγχου είναι μία ακολουθιακή μηχανή καταστάσεων με πολλές εξόδους

Μεταφορά Πληροφορίας



Όλες οι μεταφορές γίνονται σε ακμές ρολογιού οι οποίες δεν συμπεριλαμβάνονται στους συμβολισμούς.

Υπό συνθήκη μεταφορά: "If ($T1=1$) then $R2 \leq R1$ "

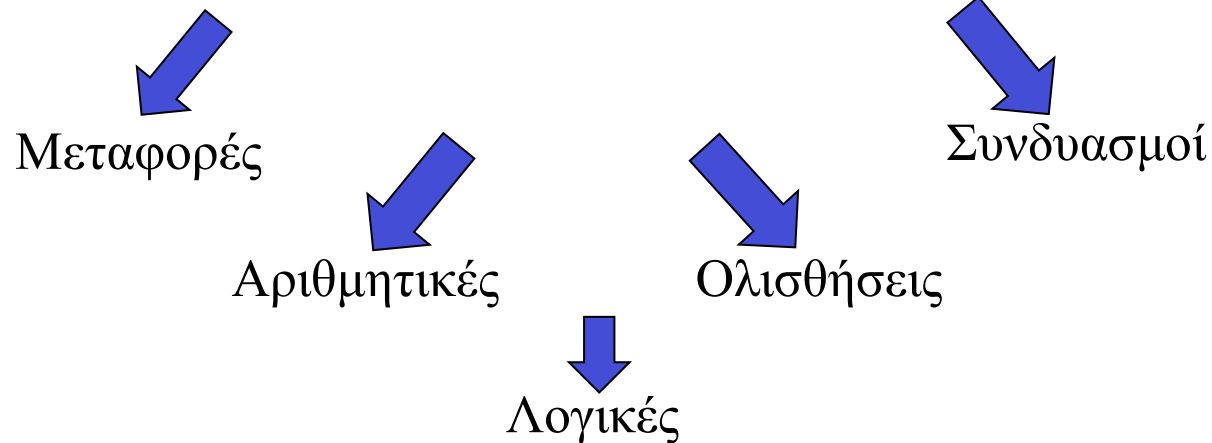
Η συνθήκη ελέγχεται από την μονάδα ελέγχου

Η μεταφορά συμβαίνει στην επόμενη ακμή ρολογιού

Υπό συνθήκη πολλαπλή μεταφορά: "If ($T1=1$) then ($R2 \leq R1, R1 \leq R2$)"

Λειτουργίες

$R1 \leftarrow R1 + R2$	Add contents of $R2$ to $R1$ ($R1$ gets $R1 + R2$)
$R3 \leftarrow R3 + 1$	Increment $R3$ by 1 (count upwards)
$R4 \leftarrow \text{shr } R4$	Shift right $R4$
$R5 \leftarrow 0$	Clear $R5$ to 0



Πράξεις Καταχωρητών

Ένα ψηφιακό κύκλωμα μπορεί να οριστεί από τους καταχωρητές που περιέχει και τις πράξεις που εκτελούνται πάνω στα δεδομένα.

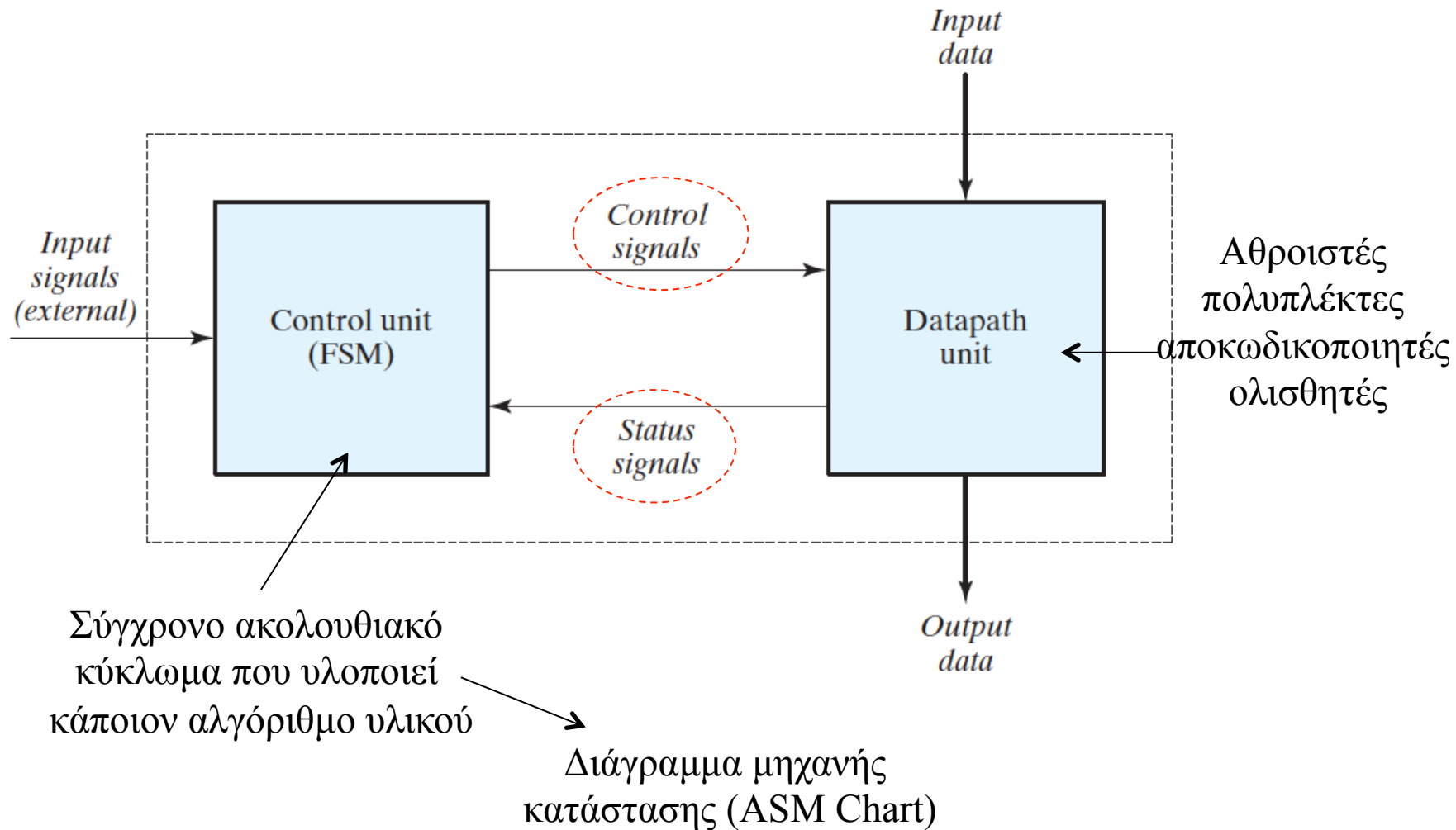
ΠΙΝΑΚΑΣ 8 -1

Συμβολισμοί Πράξεων Καταχωρητών

Συμβολισμός	Περιγραφή
$A \leftarrow B$	Μεταφορά του περιεχόμενου του καταχωρητή B στον καταχωρητή A
$R \leftarrow 0$	Μηδένιση του καταχωρητή R .
$F \leftarrow 1$	Θέση του flip-flop F στην κατάσταση 1
$A \leftarrow A + 1$	Αύξηση του καταχωρητή A κατά 1 (μέτρηση προς-τα-πάνω).
$A \leftarrow A - 1$	Μείωση του καταχωρητή A κατά 1 (μέτρηση προς-τα-κάτω).
$A \leftarrow A + B$	Πρόσθεση του περιεχομένου του καταχωρητή B στον καταχωρητή A .

Ο όρος καταχωρητής με την ευρύτερη του έννοια περιλαμβάνει καταχωρητές αποθήκευσης, καταχωρητές ολίσθησης, μετρητές και μεμονωμένα flip flops.

Αλγοριθμικές Μηχανές κατάστασης

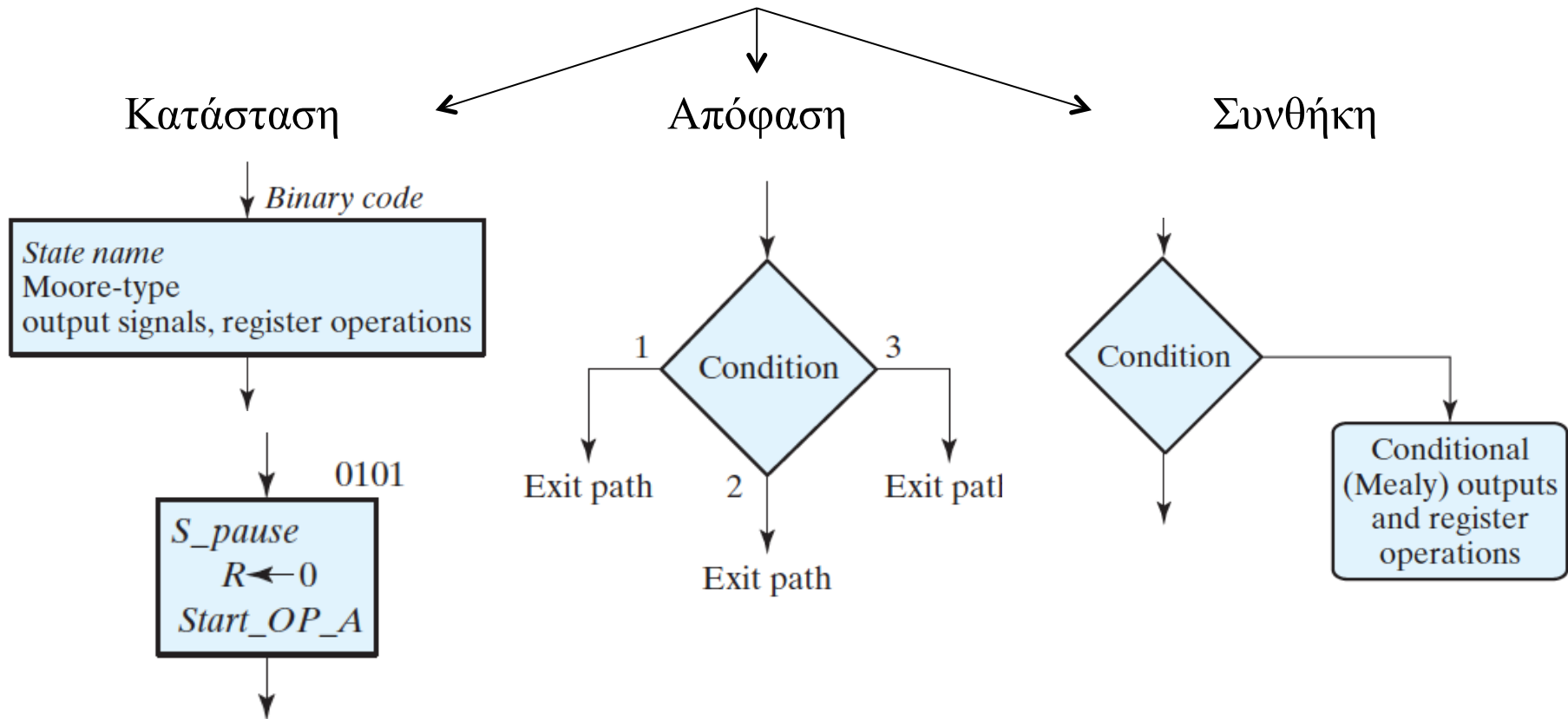


Διαγράμματα ASM

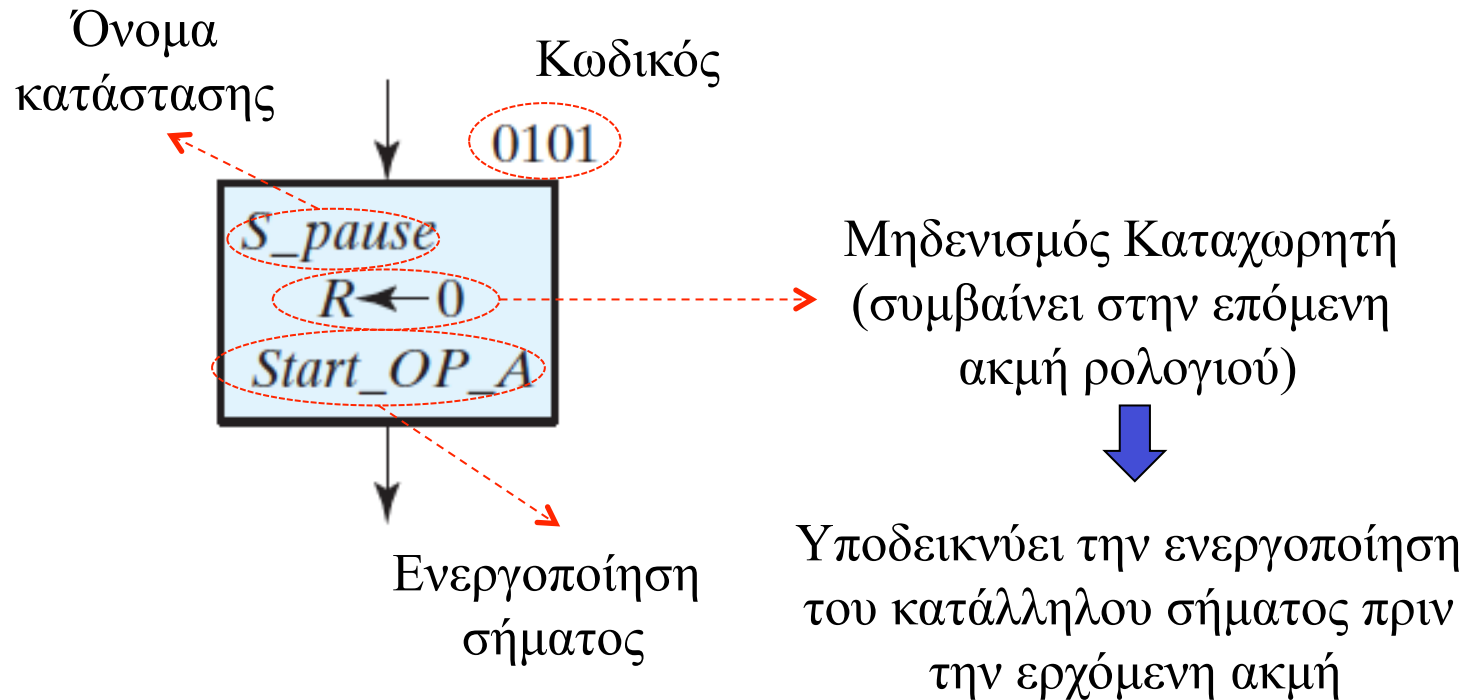
Περιγράφουν την αλληλουχία γεγονότων στον χρόνο



Μοιάζουν με συμβατικά διαγράμματα καταστάσεων

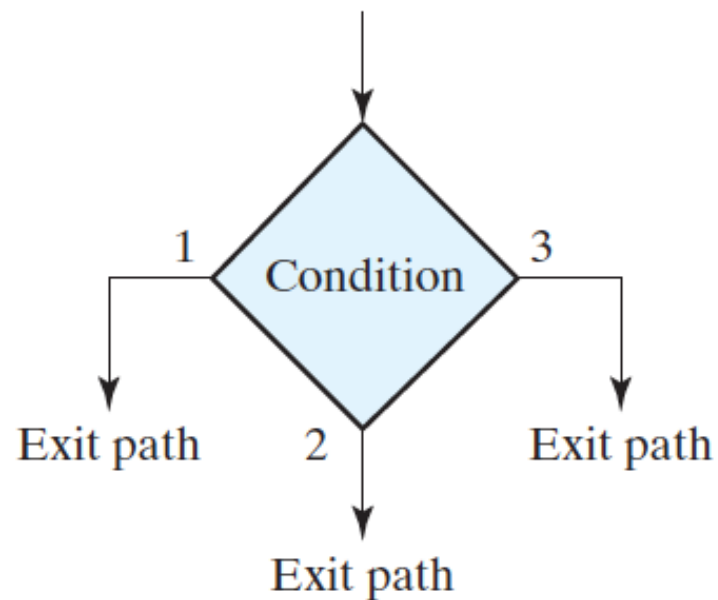


Κουτί κατάστασης



Κουτί Απόφασης

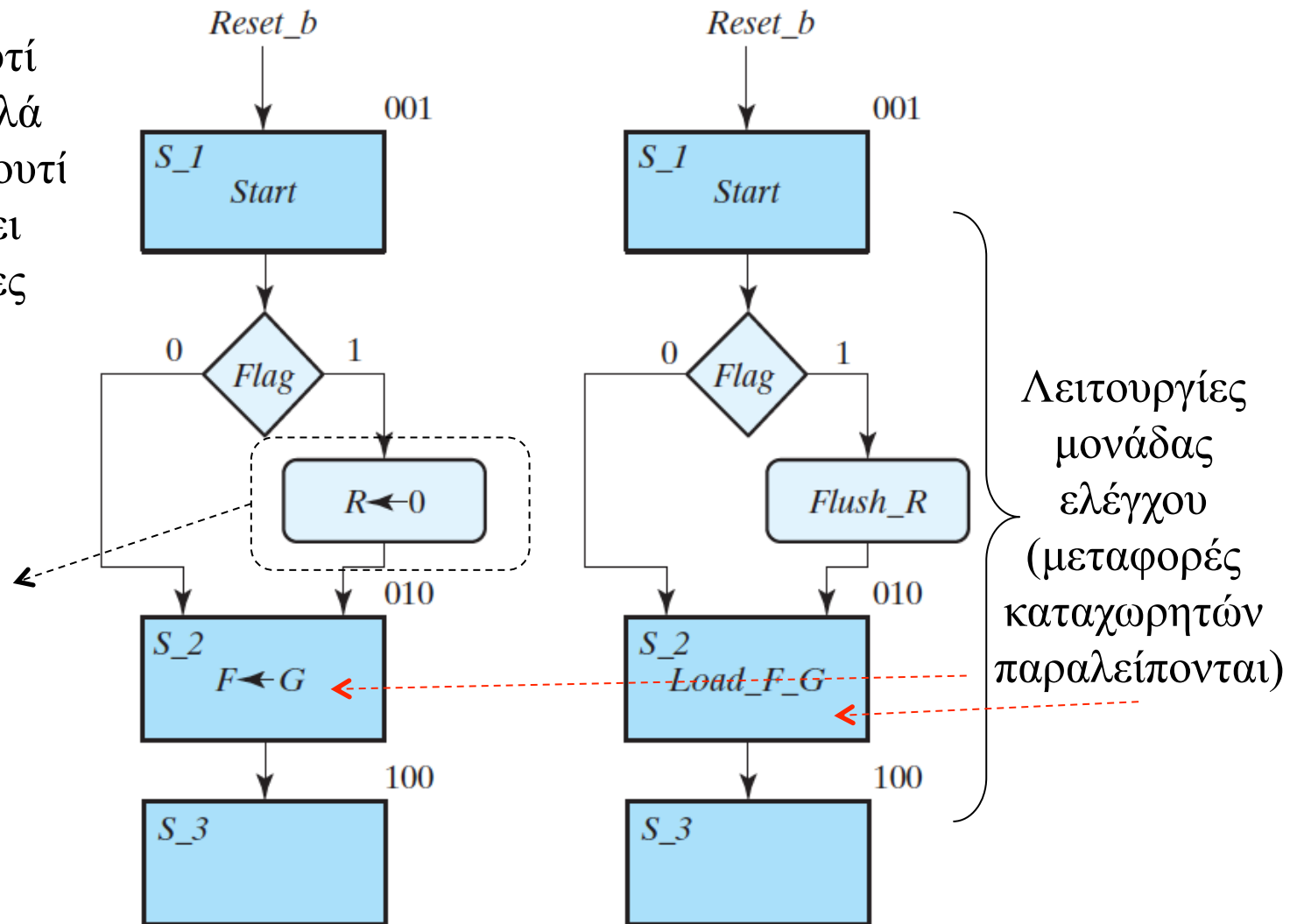
Περιγράφει την επίδραση μίας εισόδου στο υποσύστημα ελέγχου



Κουτί Συνθήκης

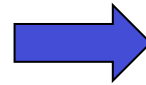
Μοιάζει με κουτί
κατάστασης αλλά
ακολουθεί ένα κουτί
απόφασης (έχει
στρογγυλεμένες
γωνίες)

Ο μηδενισμός
θα συμβεί στην
επόμενη ακμή
του ρολογιού

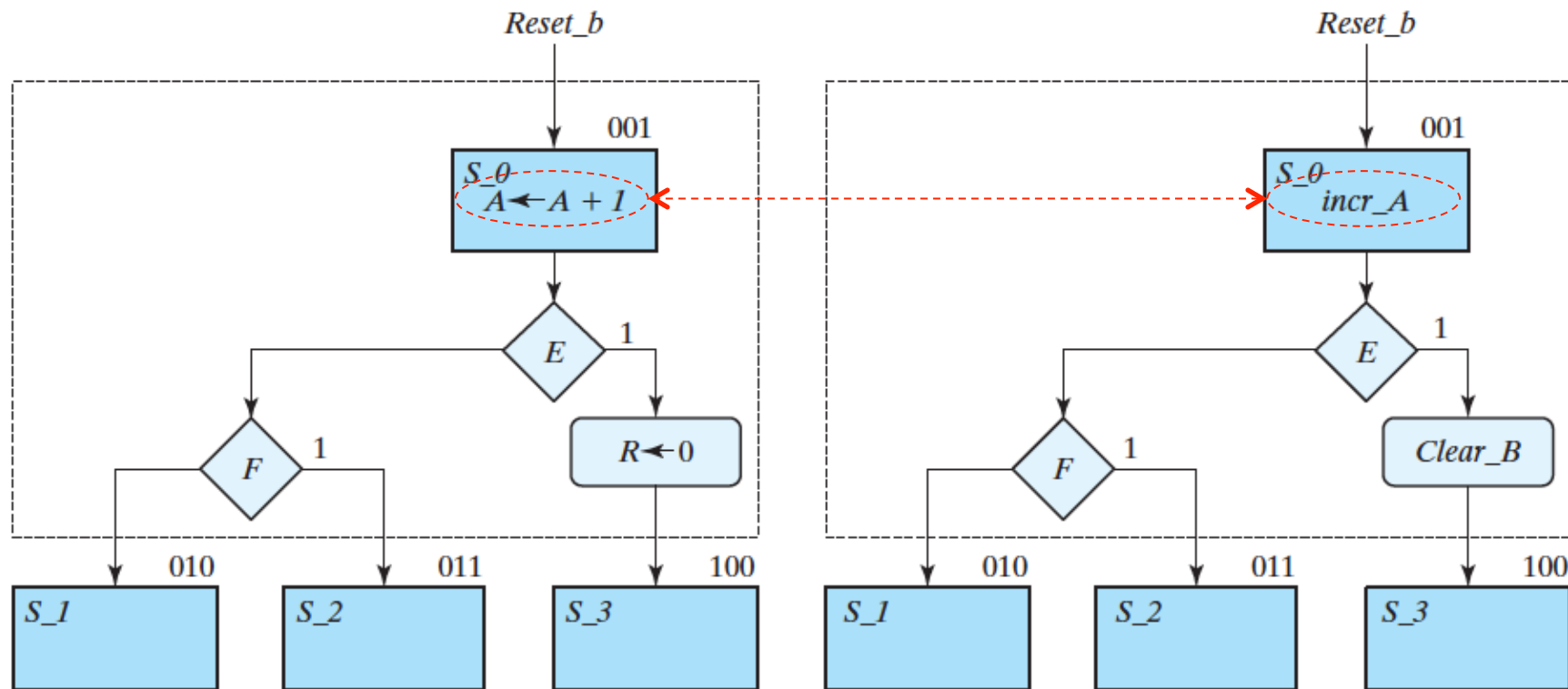


ASM Block

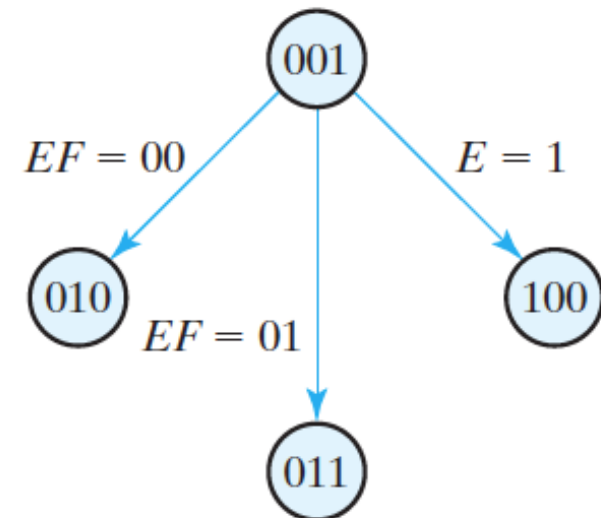
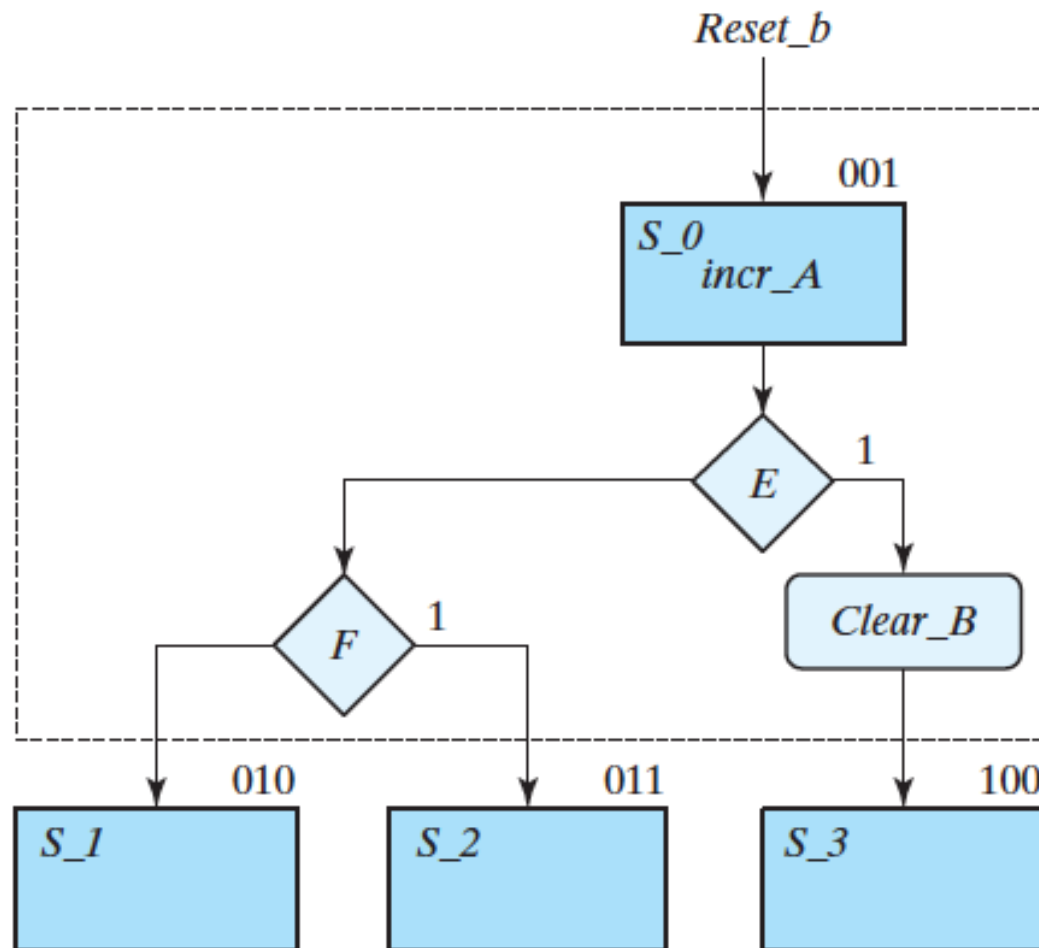
Αποτελείται από ένα κουτί
κατάστασης και όσα κουτιά
απόφασης/συνθήκης απαιτούνται
στις εξόδους του



Περιγράφει μία κατάσταση της
μηχανής στην διάρκεια ενός
παλμού ρολογιού



Αντιστοίχιση με Διάγραμμα Καταστάσεων

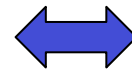


Θέματα Χρονισμού

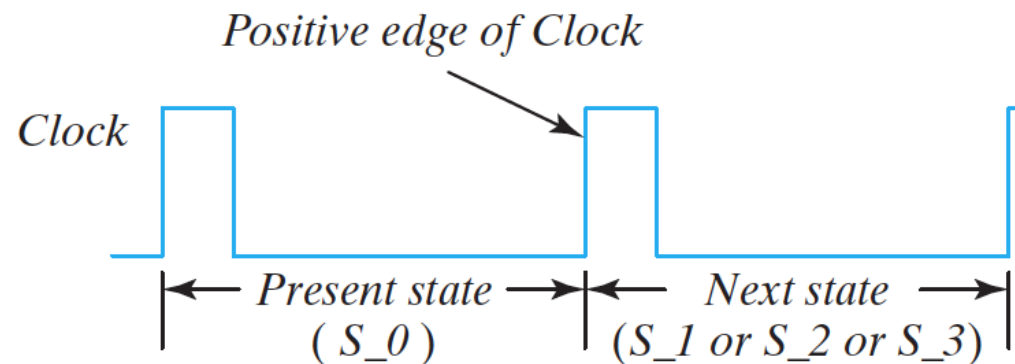
Τα ffs της Μονάδας Επεξεργασίας Δεδομένων και της Μονάδας Ελέγχου χρονίζονται από το ίδιο ρολόι.

Οι είσοδοι μπορούν να είναι **σύγχρονοι** ή **ασύγχρονοι**

Η αλγοριθμική μηχανή
κατάστασης εμπεριέχει και
στοιχεία χρονισμού

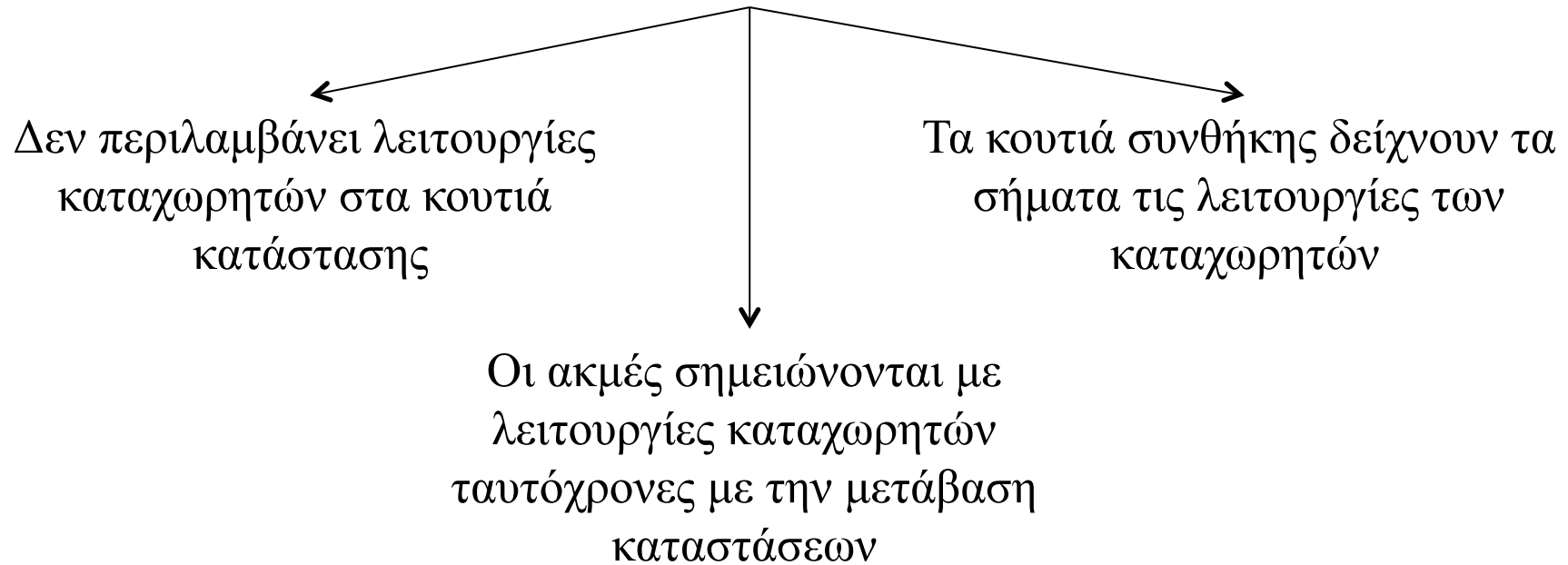


Κάθε μπλοκ αντιστοιχεί σε ένα
κύκλο ρολογιού



Διάγραμμα ASMD

Algorithmic State Machine and DataPath

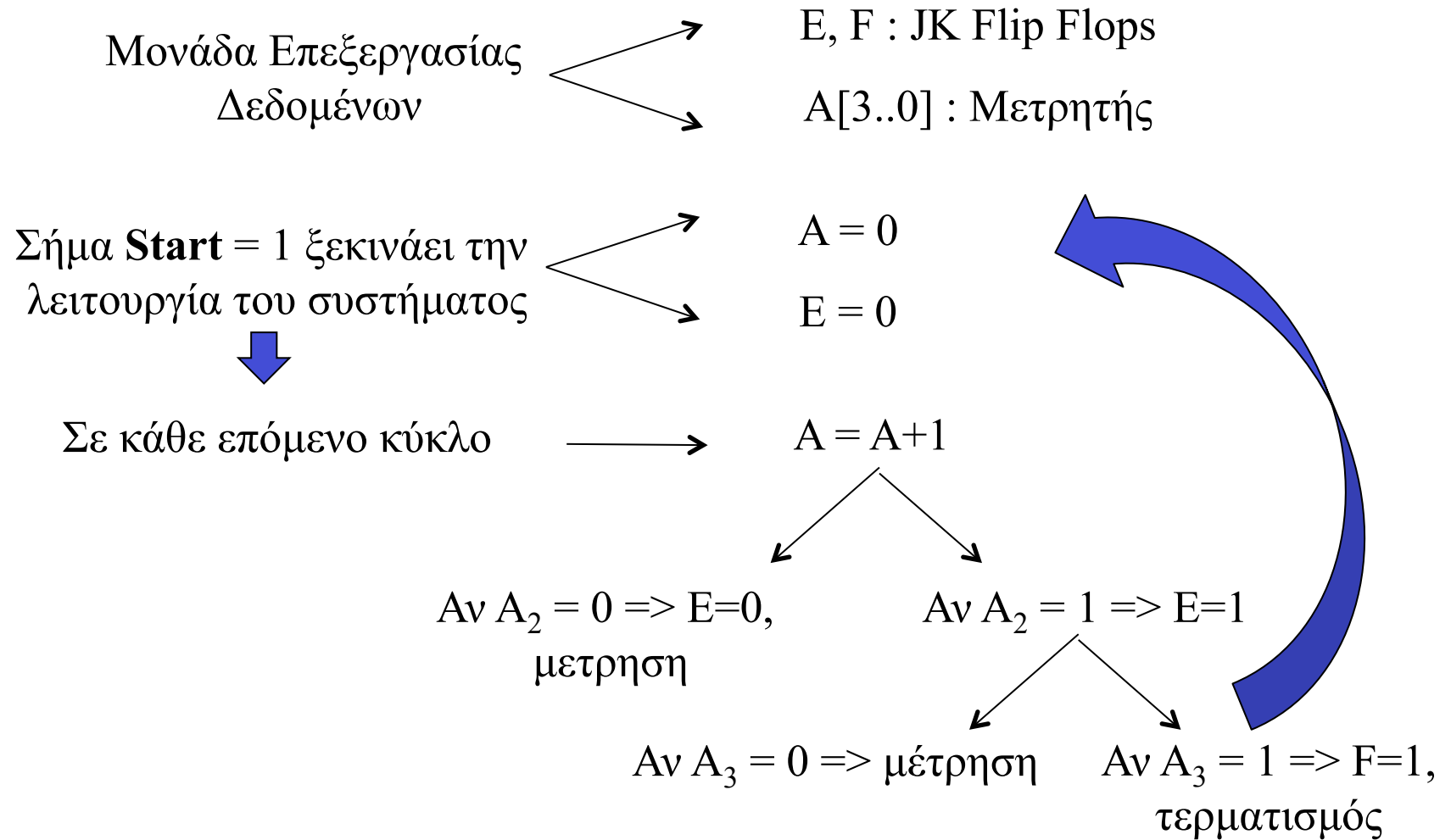


Το διάγραμμα ASMD διαμερίζει ένα σύστημα σε Επεξεργασία Δεδομένων και Μονάδα Ελέγχου

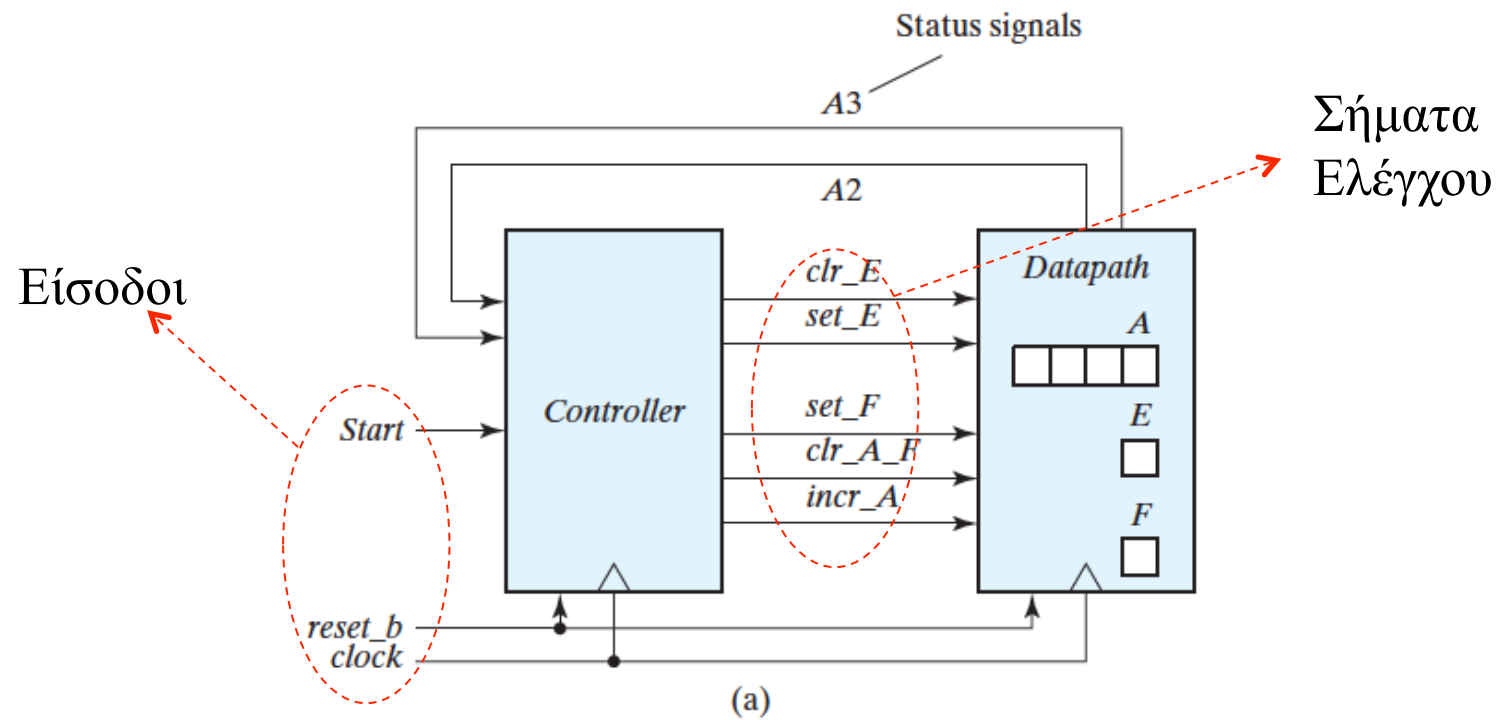
Δημιουργία Διαγράμματος ASMD

1. Δημιουργία διαγράμματος ASM με τις καταστάσεις μόνο και τα σήματα εισόδου που προκαλούν μεταβάσεις.
2. Μετατροπή του διαγράμματος ASM σε ASMD σημειώνοντας στις ακμές του ASM διαγράμματος τις λειτουργίες των καταχωρητών.
3. Μετατροπή του διαγράμματος ASMD με προσθήκη των σημάτων ελέγχου που παράγονται από τον ελεγκτή και προκαλούν τις λειτουργίες των καταχωρητών.

Παράδειγμα ASMD

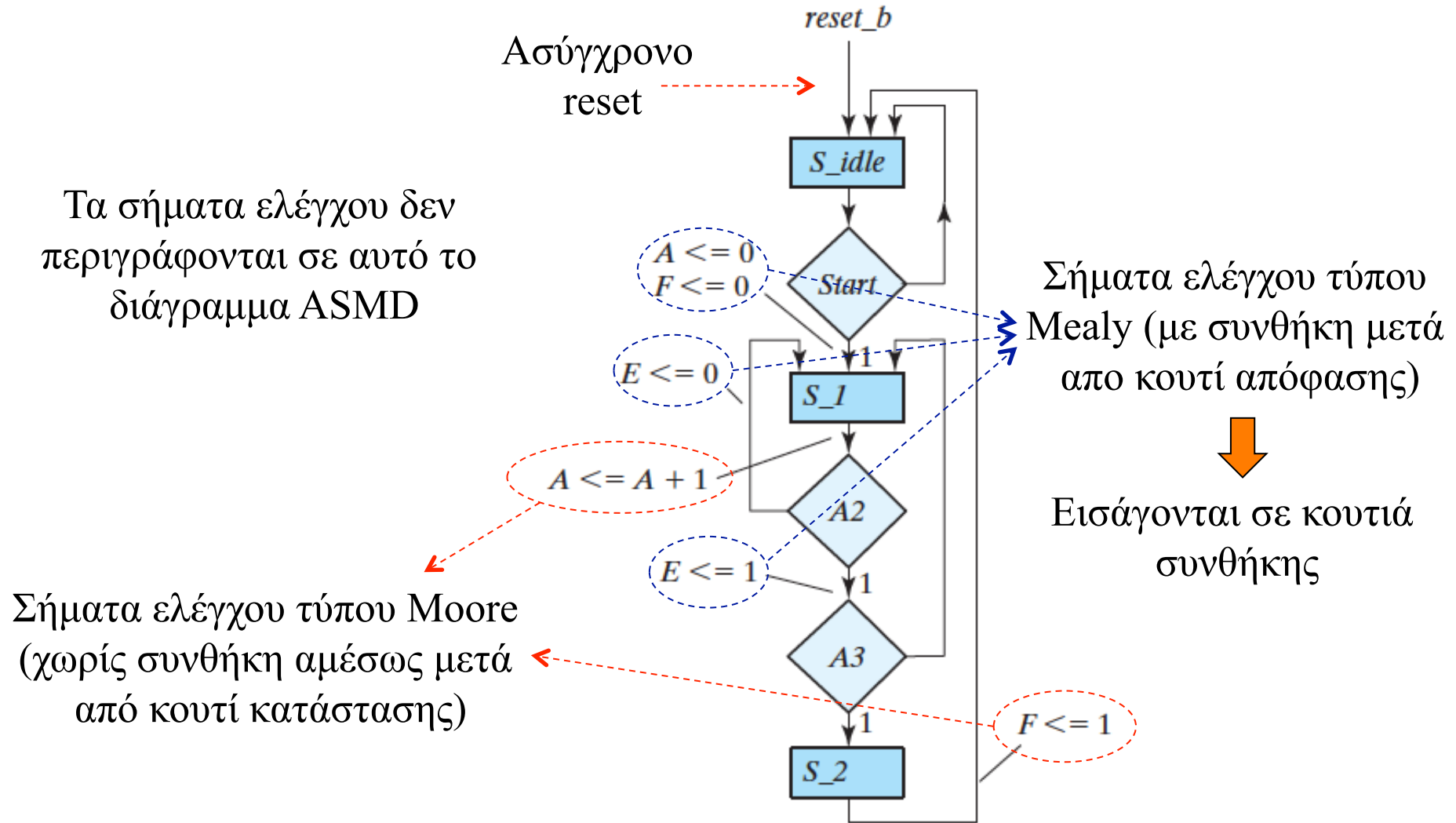


Παράδειγμα ASMD

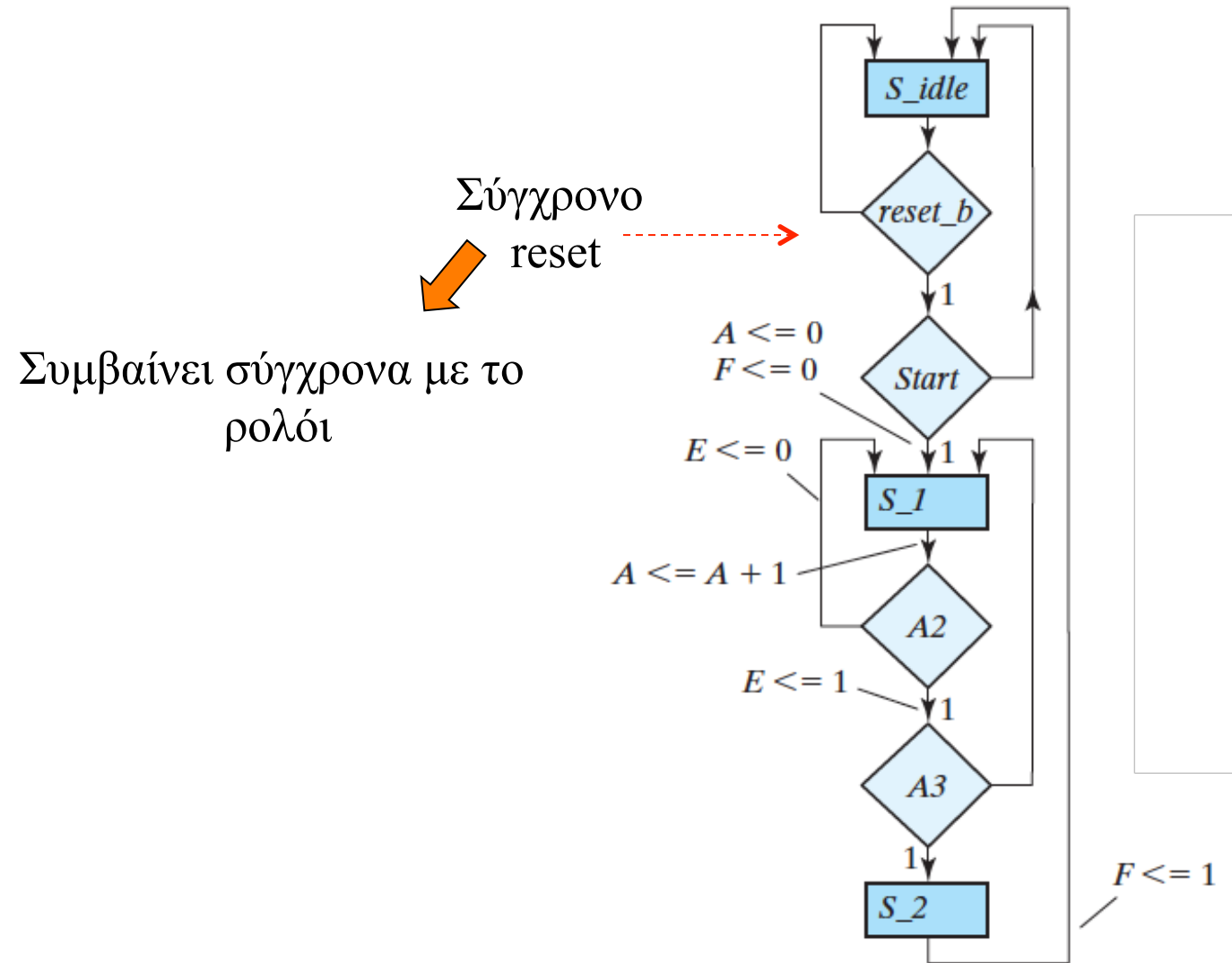


Μπλοκ Διάγραμμα

Παράδειγμα ASMD

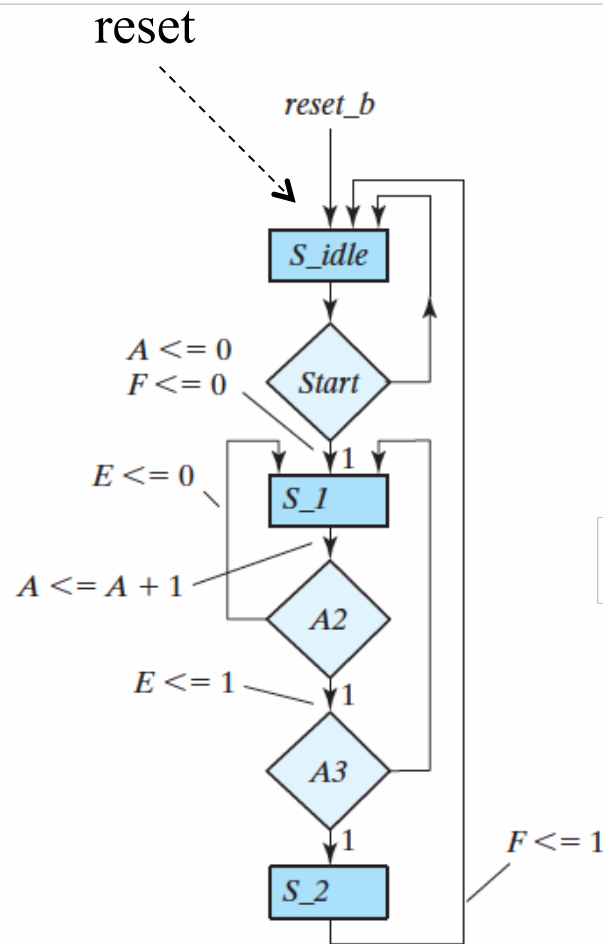


Παράδειγμα ASMD



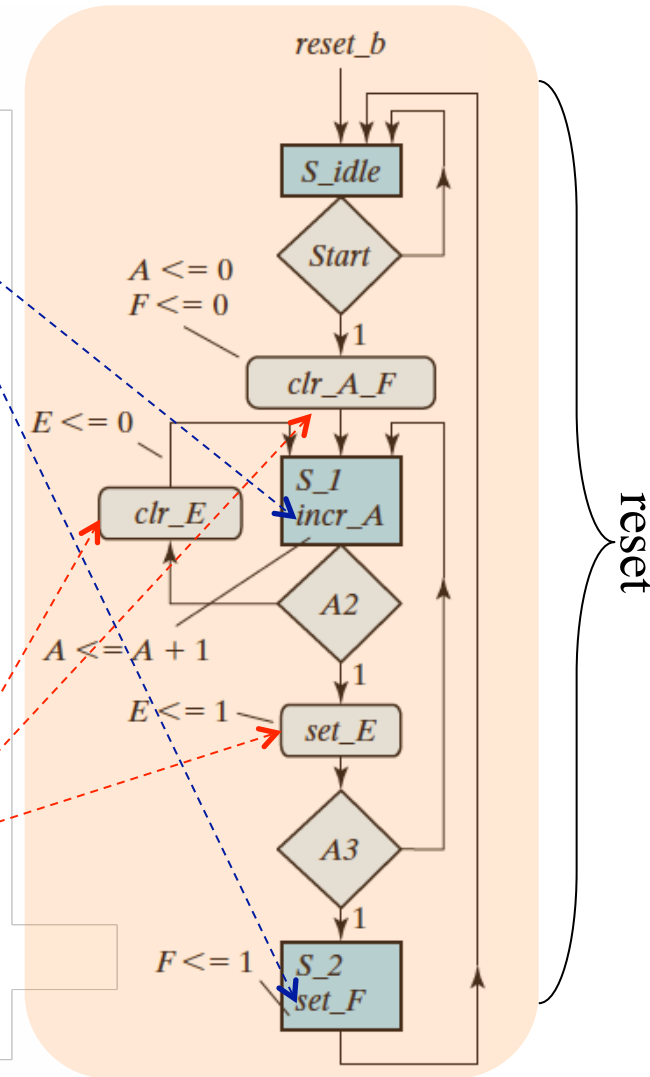
Παράδειγμα ASMD

Ασύγχρονο



Τα σήματα
ελέγχου τύπου
Moore μπαίνουν
στα κουτιά
κατάστασης

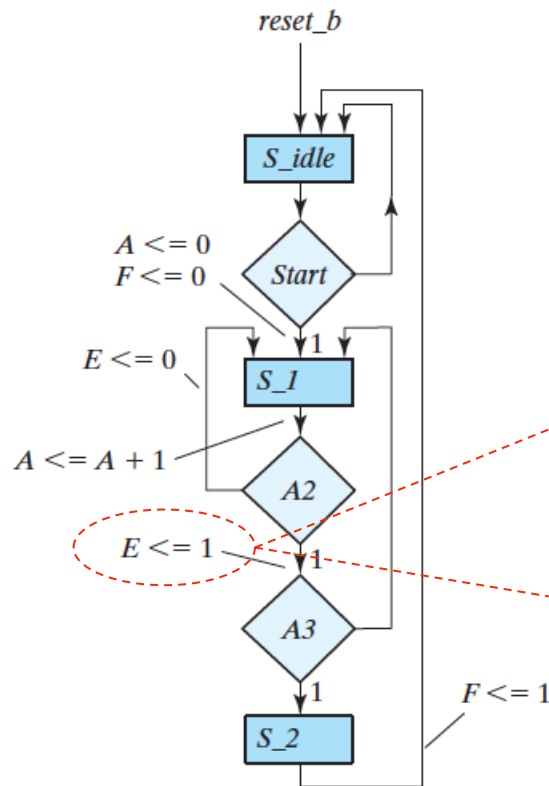
Τα σήματα
ελέγχου τύπου
Mealy μπαίνουν
στα κουτιά
συνθήκης



Ολοκληρωμένη ASMD με Ασύγχρονο
reset

Παράδειγμα ASMD

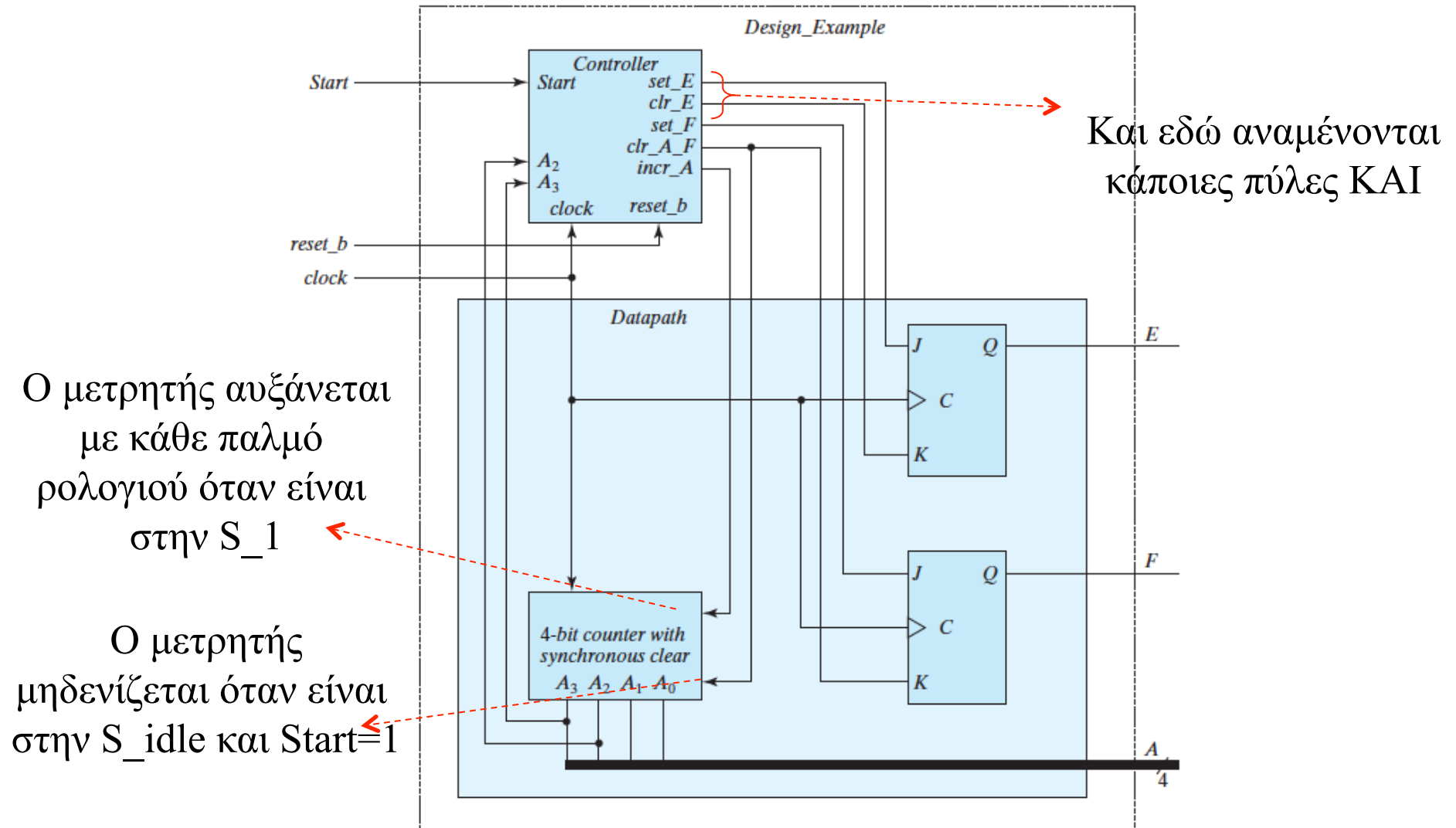
Το σύστημα ήταν σε λειτουργία πριν την S_1 και δέχθηκε το σήμα Start



Counter				Flip-Flops		Conditions	State
A ₃	A ₂	A ₁	A ₀	E	F		
0	0	0	0	1	0	A ₂ = 0, A ₃ = 0	S ₁
0	0	0	1	0	0		
0	0	1	0	0	0		
0	0	1	1	0	0		
0	1	0	0	0	0	A ₂ = 1, A ₃ = 0	
0	1	0	1	1	0		
0	1	1	0	1	0		
0	1	1	1	1	0		
1	0	0	0	1	0	A ₂ = 0, A ₃ = 1	
1	0	0	1	0	0		
1	0	1	0	0	0		
1	0	1	1	0	0		
1	1	0	0	0	0	A ₂ = 1, A ₃ = 1	
1	1	0	1	1	0		S ₂
1	1	0	1	1	1		S _{idle}

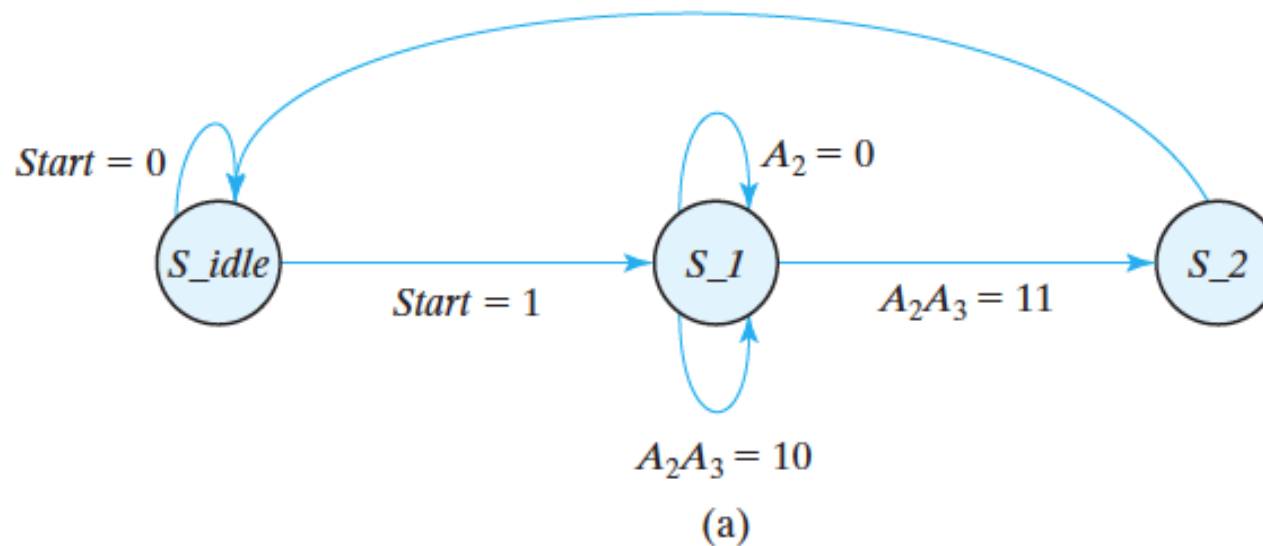
Οι αλλαγές γίνονται ένα κύκλο μετά

Παράδειγμα ASMD



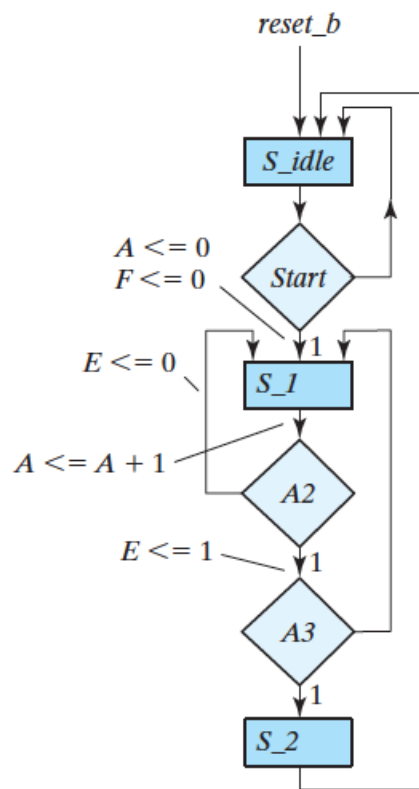
Παράδειγμα ASMD

Εναλλακτική Αναπαράσταση Μεταφοράς Καταχωρητή



$S_idle \longrightarrow S_1, clr_A_F:$	$A \longleftarrow 0, F \longleftarrow 0$
$S_1 \longrightarrow S_1, incr_A:$	$A \longleftarrow A + 1$
$if (A_2 = 1) then set_E:$	$E \longleftarrow 1$
$if (A_2 = 0) then clr_E:$	$E \longleftarrow 0$
$S_2 \longrightarrow S_idle, set_F:$	$F \longleftarrow 1$

Παράδειγμα ASMD



Πίνακας Καταστάσεων

Present-State Symbol	Present State		Inputs			Next State		Outputs				
	G ₁	G ₀	Start	A ₂	A ₃	G ₁	G ₀	set_E	clr_E	set_F	clr_A_F	incr_A
S_idle	0	0	0	X	X	0	0	0	0	0	0	0
S_idle	0	0	1	X	X	0	1	0	0	0	1	0
S_1	0	1	X	0	X	0	1	0	1	0	0	1
S_1	0	1	X	1	0	0	1	1	0	0	0	1
S_1	0	1	X	1	1	1	1	1	0	0	0	1
S_2	1	1	X	X	X	0	0	0	0	1	0	0

Κωδικοποίηση Καταστάσεων με 2 bits

$$\left\{ \begin{array}{ll} S_idle & \leq 00 \\ S_1 & \leq 01 \\ S_2 & \leq 11 \end{array} \right\} \text{ Χρήση D-FFs}$$

Η σχεδίαση μπορεί να γίνει με τις τεχνικές σχεδίασης ακολουθιακών κυκλωμάτων

Παράδειγμα ASMD

	Present State		Inputs			Next State		Outputs				
Present-State Symbol	G ₁	G ₀	Start	A ₂	A ₃	G ₁	G ₀	set_E	clr_E	set_F	clr_A_F	incr_A
<i>S_idle</i>	0	0	0	X	X	0	0	0	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	0	0	0	1	0
<i>S_1</i>	0	1	X	0	X	0	1	0	1	0	0	1
<i>S_1</i>	0	1	X	1	0	0	1	1	0	0	0	1
<i>S_1</i>	0	1	X	1	1	1	1	1	0	0	0	1
<i>S_2</i>	1	1	X	X	X	0	0	0	0	1	0	0

Απαιτούνται χάρτες 5
μεταβλητών



Οι συναρτήσεις μπορούν να
εξαχθούν από τον πίνακα απευθείας

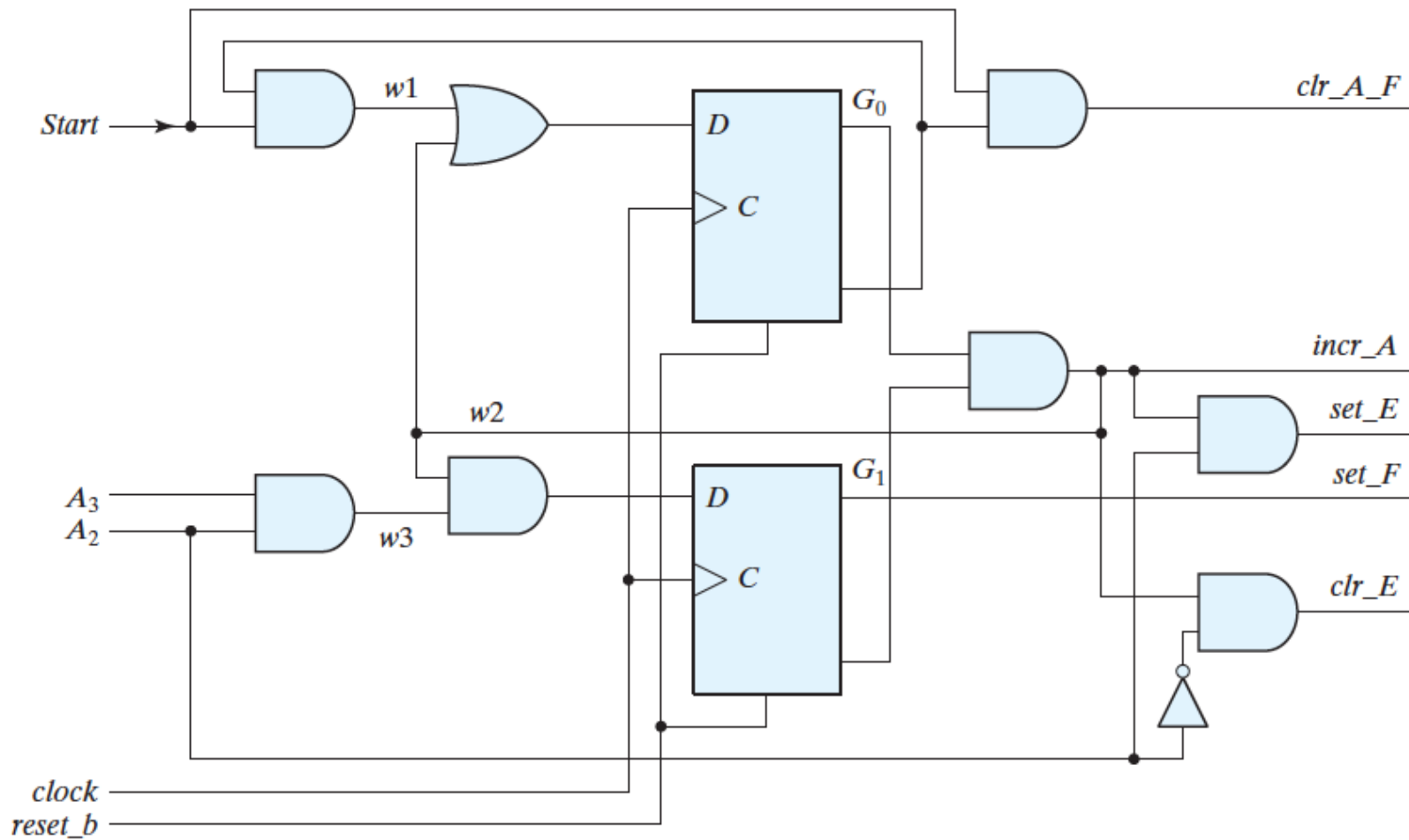
$$DG1 = S_1 \cdot A_2 \cdot A_3$$

$$DG0 = Start \cdot S_idle + S_1$$

$$\begin{aligned} set_E &= S_1 A_2 \\ clr_E &= S_1 A_2' \\ set_F &= S_2 \\ clr_A_F &= Start S_idle \\ incr_A &= S_1 \end{aligned}$$

Παράδειγμα ASMD

Μονάδα Ελέγχου Παραδείγματος



Παράδειγμα 2: Πολλαπλασιαστής

Ο ακολουθιακός πολλαπλασιαστής
βασίζεται σε έναν ολισθητή και
έναν αθροιστή

Έχει μικρό κόστος σε υλικό

Απαιτεί πολλούς κύκλους



Διαδοχικές
προσθέσεις και
ολισθήσεις

23 10111 multiplican

19 10011 multiplier

10111

10111

00000

00000

10111

437

110110101 product

Προσθέσεις

Αριστερές
Ολισθήσεις

Μήκος $n+n$

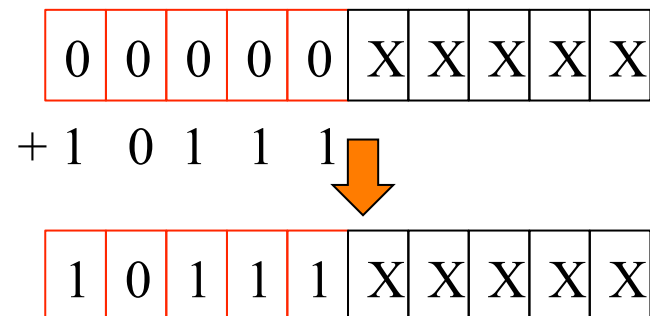
Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Βήμα 1:

1^ο μερικό γινόμενο και
πρόσθεση στα 5 πιο
σημαντικά ψηφία του
αποτελέσματος

Ολισθητής Αποτελέσματος



Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Βήμα 2:

Ολίσθηση μερικού
αποτελέσματος προς τα δεξιά

Ολισθητής Αποτελέσματος

1	0	1	1	1	X	X	X	X	X
---	---	---	---	---	---	---	---	---	---



0	1	0	1	1	1	X	X	X	X
---	---	---	---	---	---	---	---	---	---

Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Βήμα 3:

2^ο μερικό γινόμενο και
πρόσθεση στα 5 πιο
σημαντικά ψηφία του
αποτελέσματος

Ολισθητής Αποτελέσματος

0	1	0	1	1	1	X	X	X	X
---	---	---	---	---	---	---	---	---	---

+ 1 0 1 1 1



Κρατούμενο

← 1

0	0	0	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---

Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Βήμα 4:

Ολίσθηση μερικού
αποτελέσματος προς τα δεξιά

Ολισθητής Αποτελέσματος

1 0 0 0 1 0 1 X X X X



1 0 0 0 1 0 1 X X X

Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Βήμα 5-6:

Το επόμενο μερικό γινόμενο είναι μηδενικό οπότε έχουμε μόνο ολίσθηση μερικού αποτελέσματος προς τα δεξιά

Ολισθητής Αποτελέσματος

1	0	0	0	1	0	1	X	X	X
---	---	---	---	---	---	---	---	---	---



0	1	0	0	0	1	0	1	X	X
---	---	---	---	---	---	---	---	---	---

Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Βήμα 7-8:

Το επόμενο μερικό γινόμενο είναι μηδενικό οπότε έχουμε μόνο ολίσθηση μερικού αποτελέσματος προς τα δεξιά

Ολισθητής Αποτελέσματος

0	1	0	0	0	1	0	1	X	X
---	---	---	---	---	---	---	---	---	---



0	0	1	0	0	0	1	0	1	X
---	---	---	---	---	---	---	---	---	---

Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Βήμα 9:

5^ο μερικό γινόμενο και
πρόσθεση στα 5 πιο
σημαντικά ψηφία του
αποτελέσματος

Ολισθητής Αποτελέσματος

0 0 1 0 0 0 1 0 1 X
+ 1 0 1 1 1
↓
Κρατούμενο ← 0 1 1 0 1 1 0 1 0 1 X

Παράδειγμα 2: Πολλαπλασιαστής

10111 multiplican
10011 multiplier
10111
10111
00000
00000
10111
110110101 product

Τελικό
αποτέλεσμα

Βήμα 10:

Ολίσθηση μερικού
αποτελέσματος προς τα δεξιά

Ολισθητής Αποτελέσματος

1	1	0	1	1	0	1	0	1	X
---	---	---	---	---	---	---	---	---	---



0	1	1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---

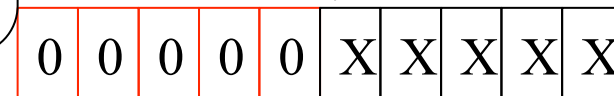
Παράδειγμα 2: Πολλαπλασιαστής

Παρατήρηση 1^η: Κάθε φορά μόνο ένα δυαδικό ψηφίο του πολλαπλασιαστή μας χρειάζεται.

Παρατήρηση 2^η: Το λιγότερο σημαντικό μέρος του αποτελέσματος γεμίζει σταδιακά.

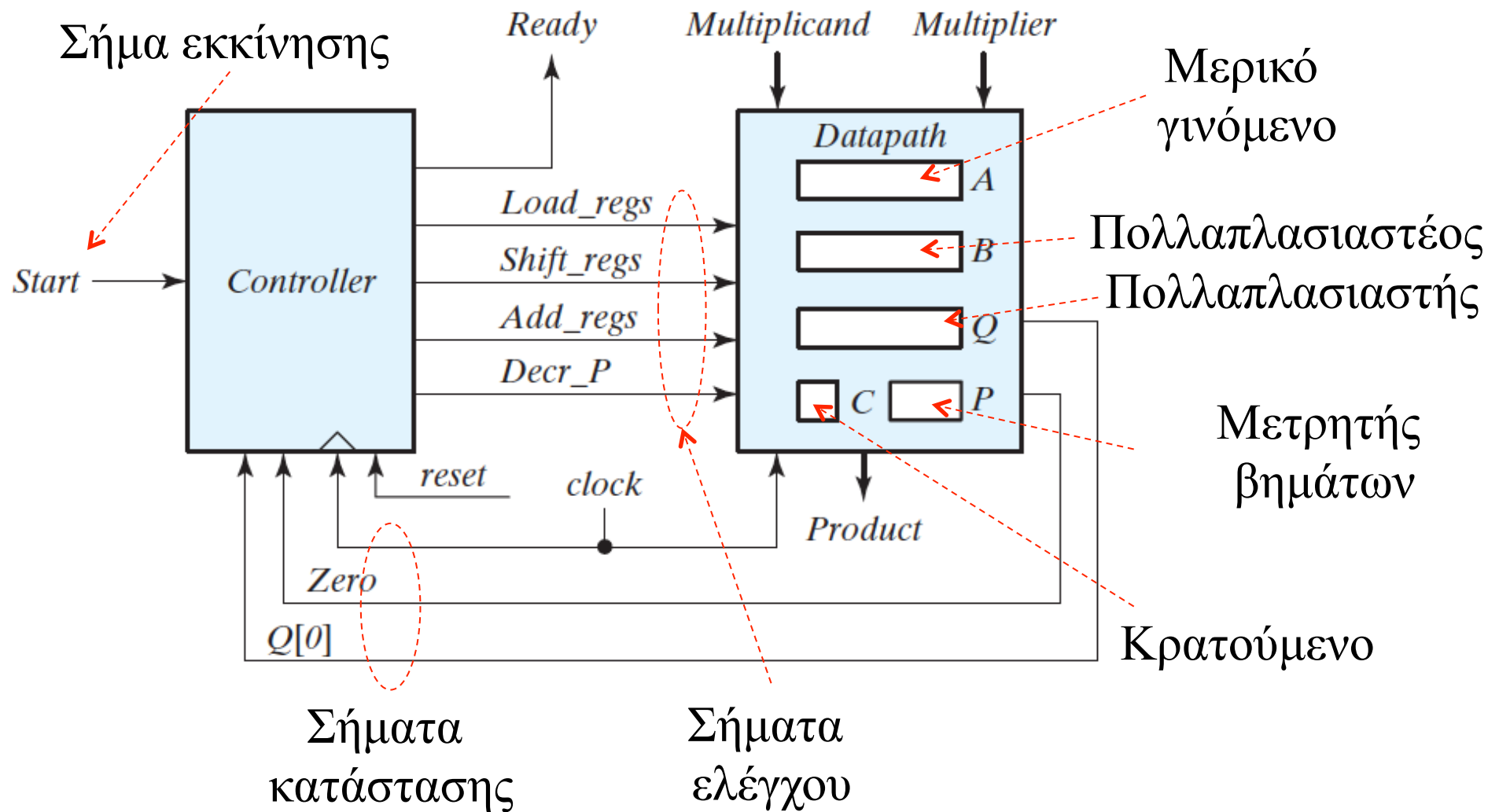
Παρατήρηση 3^η: Μετά από κάθε 2^ο βήμα το λιγότερο σημαντικό ψηφίο του πολλαπλασιαστή είναι άχρηστο

Αποθήκευση του πολλαπλασιαστή στο λιγότερο σημαντικό τμήμα του αποτελέσματος



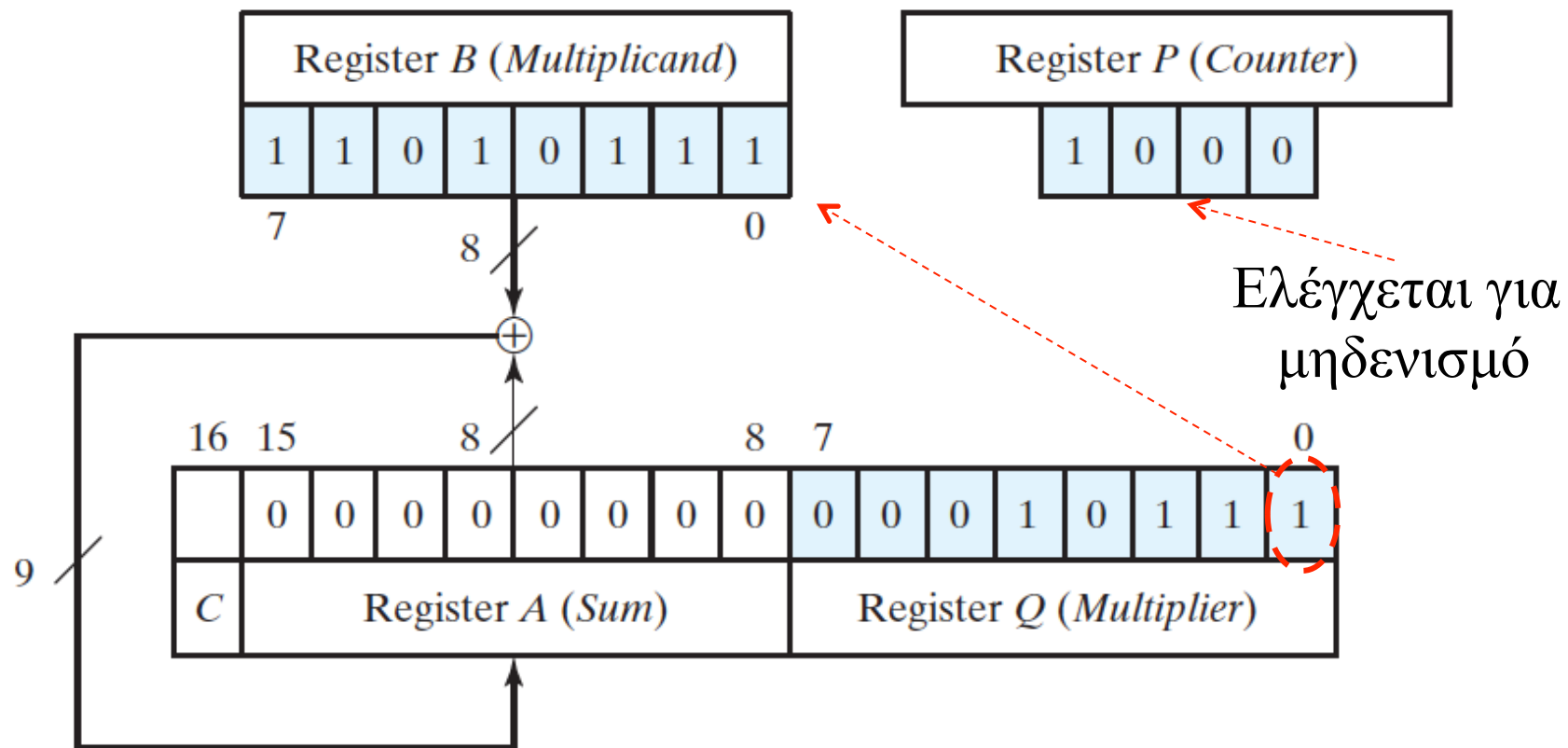
**Ολισθητής
Αποτελέσματος**

Παράδειγμα 2: Πολλαπλασιαστής



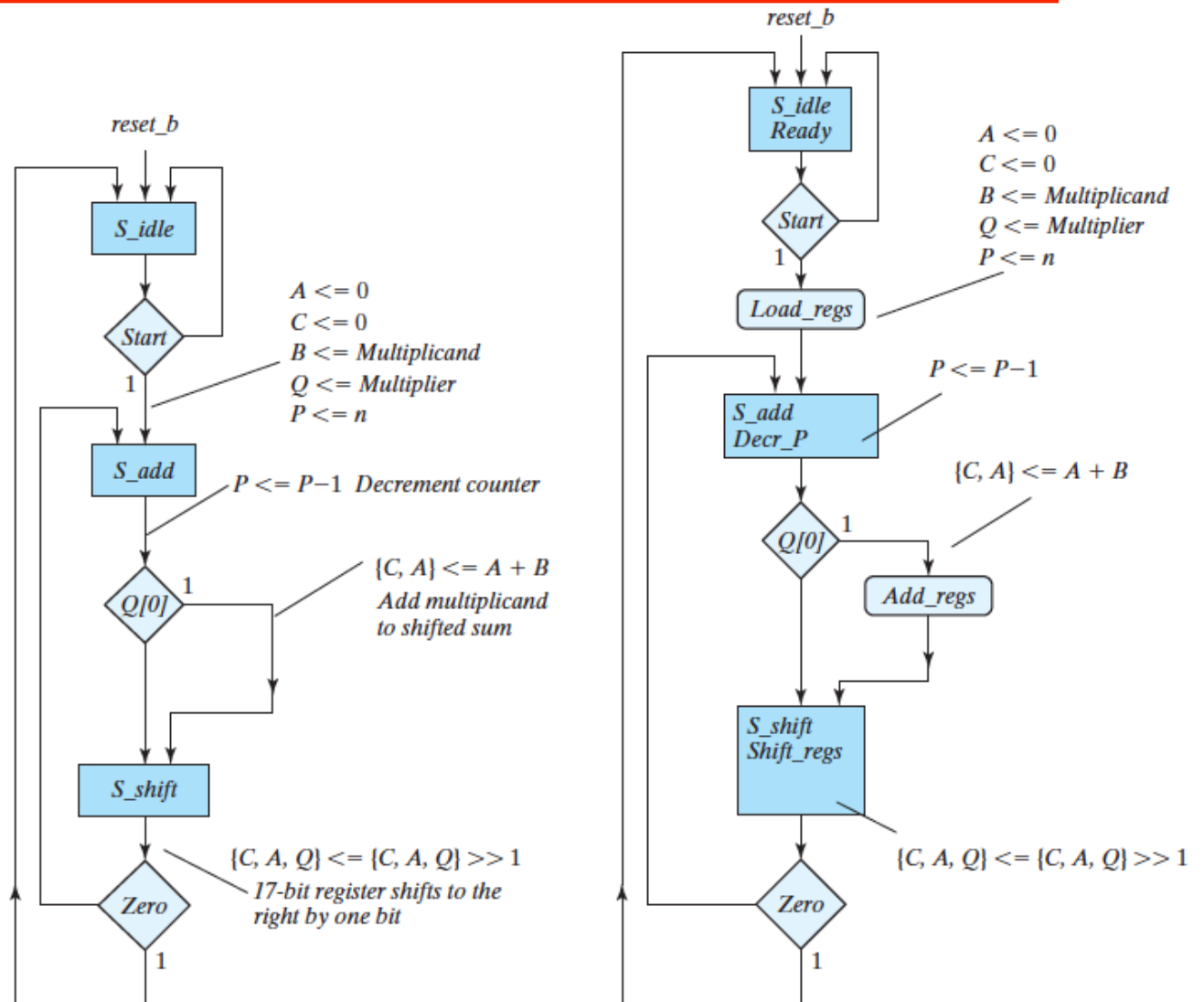
Παράδειγμα 2: Πολλαπλασιαστής

Μονάδα Επεξεργασίας Δεδομένων



Παράδειγμα 2: Πολλαπλασιαστής

Μονάδα Επεξεργασίας Δεδομένων



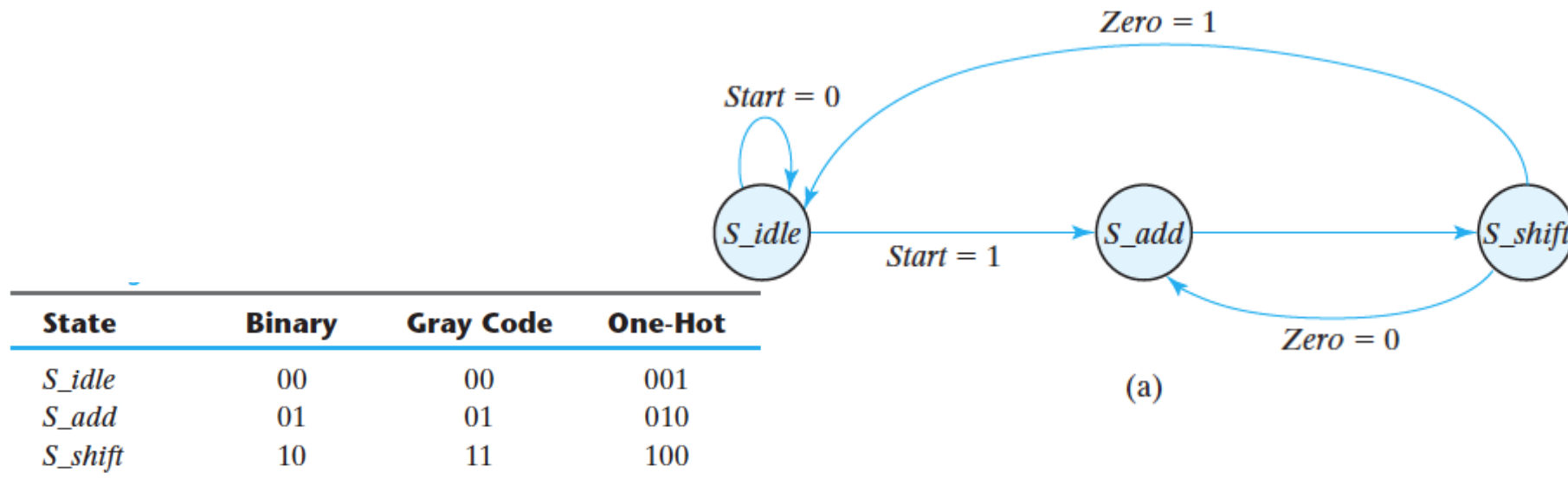
Παράδειγμα 2: Πολλαπλασιαστής

Multiplicand $B = 10111_2 = 17_H = 23_{10}$

Multiplier $Q = 10011_2 = 13_H = 19_{10}$

	C	A	Q	P
Multiplier in Q	0	00000	10011	101
$Q_0 = 1$; add B		<u>10111</u>		
First partial product	0	10111		100
Shift right CAQ	0	01011	11001	
$Q_0 = 1$; add B		<u>10111</u>		
Second partial product	1	00010		011
Shift right CAQ	0	10001	01100	
$Q_0 = 0$; shift right CAQ	0	01000	10110	010
$Q_0 = 0$; shift right CAQ	0	00100	01011	001
$Q_0 = 1$; add B		<u>10111</u>		
Fifth partial product	0	11011		
Shift right CAQ	0	01101	10101	000
Final product in $AQ = 0110110101_2 = 1b5_H$				

Παράδειγμα 2: Πολλαπλασιαστής



Κωδικοποίηση
καταστάσεων

State Transition		Register Operations
<u>From</u>	<u>To</u>	
<i>S_idle</i>		Initial state
<i>S_idle</i>	<i>S_add</i>	$A \leq 0, C \leq 0, P \leq dp_width$
<i>S_add</i>	<i>S_shift</i>	$P \leq P - 1$
<i>S_shift</i>		if ($Q[0]$) then ($A \leq A + B, C \leq C_{out}$)
		shift right {CAQ}, $C \leq 0$

Παράδειγμα 2: Πολλαπλασιαστής

Σχεδίαση με **καταχωρητή** ακολουθίας και
αποκωδικοποιητή

←
Χρήση **καταχωρητή** για τις
καταστάσεις



Συμπεριλαμβάνει και πύλες
για την μετάβαση των
καταστάσεων

→
Χρήση **αποκωδικοποιητή** για
παραγωγή ενός σήματος για
κάθε κατάσταση (όχι για on-hot)

Παράδειγμα 2: Πολλαπλασιαστής

Πίνακας Καταστάσεων

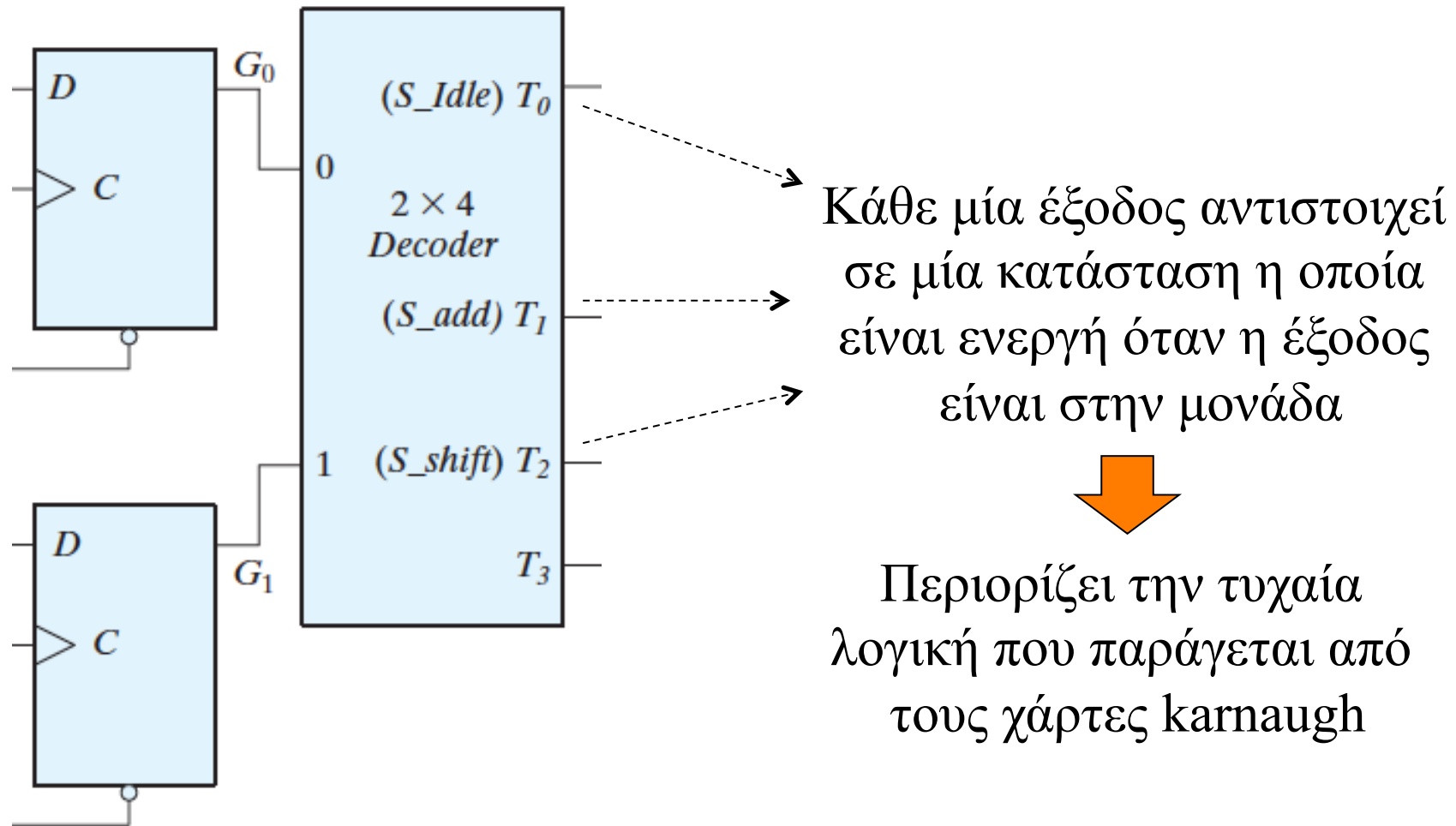
	Present State		Inputs			Next State						
Present-State Symbol	G_1	G_0	<i>Start</i>	<i>Q[0]</i>	<i>Zero</i>	G_1	G_0	<i>Ready</i>	<i>Load_regs</i>	<i>Decr_P</i>	<i>Add_regs</i>	<i>Shift_regs</i>
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

Δυαδική
Κωδικοποίηση



Αντί για χάρτες χρησιμοποιούμε
αποκωδικοποιητή

Παράδειγμα 2: Πολλαπλασιαστής

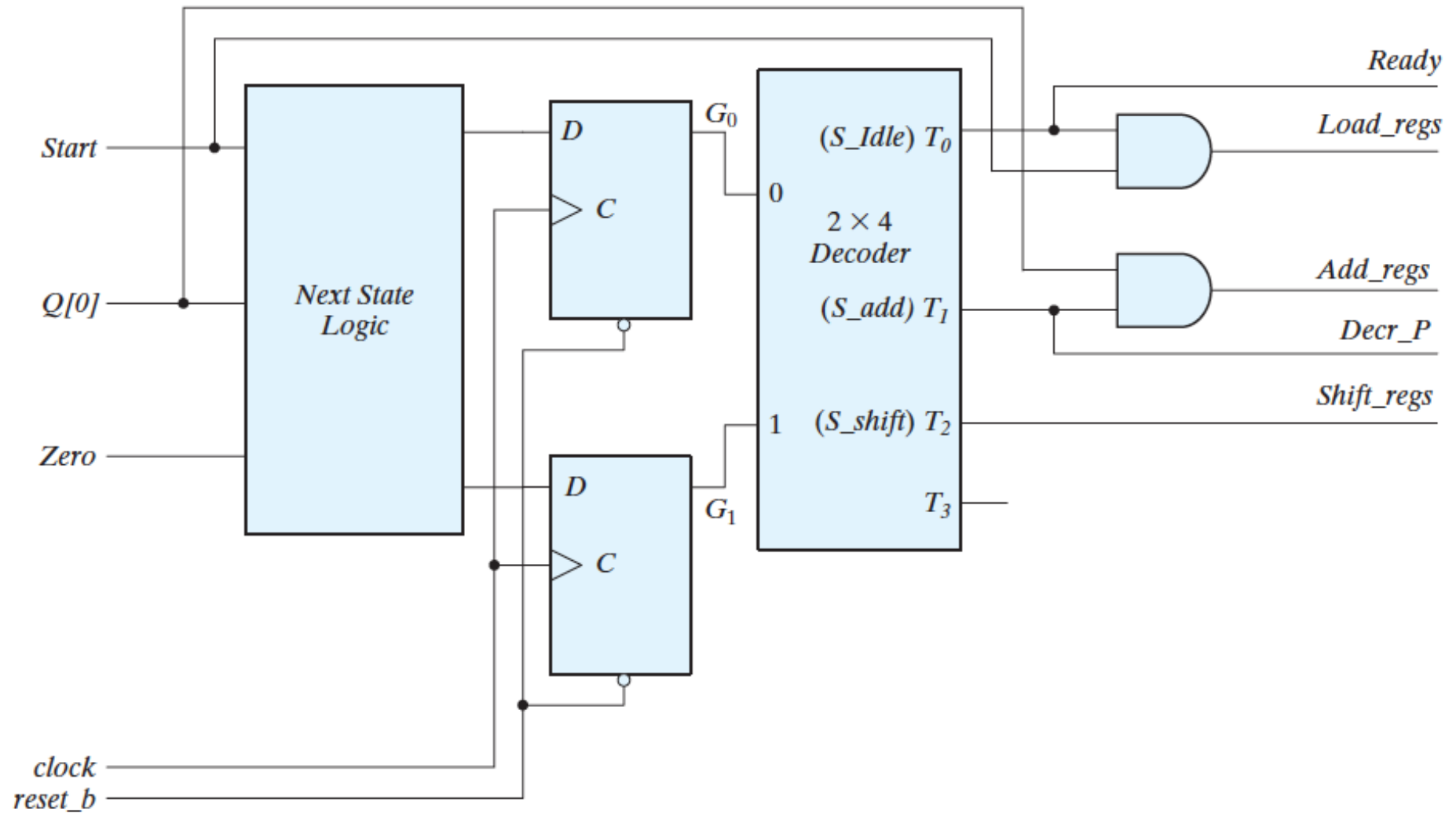


Παράδειγμα 2: Πολλαπλασιαστής

	Present State		Inputs			Next State		Ready	Load_regs	Decr_P	Add_regs	Shift_regs
	G ₁	G ₀	Start	Q[0]	Zero	G ₁	G ₀					
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

$$\text{Ready} = T_0$$

Παράδειγμα 2: Πολλαπλασιαστής



Παράδειγμα 2: Πολλαπλασιαστής

	Present State		Inputs			Next State						
Present-State Symbol	G ₁	G ₀	Start	Q[0]	Zero	G ₁	G ₀	Ready	Load_regs	Decr_P	Add_regs	Shift_regs
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

Load_regs = Start • T₀

Παράδειγμα 2: Πολλαπλασιαστής

Present-State Symbol	Present State		Inputs			Next State		Ready	Load_regs	Decr_P	Add_regs	Shift_regs
	G ₁	G ₀	Start	Q[0]	Zero	G ₁	G ₀					
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

Decr_P = T₁

Παράδειγμα 2: Πολλαπλασιαστής

Present-State Symbol	Present State		Inputs			Next State		Ready	Load_regs	Decr_P	Add_regs	Shift_regs
	G ₁	G ₀	Start	Q[0]	Zero	G ₁	G ₀					
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

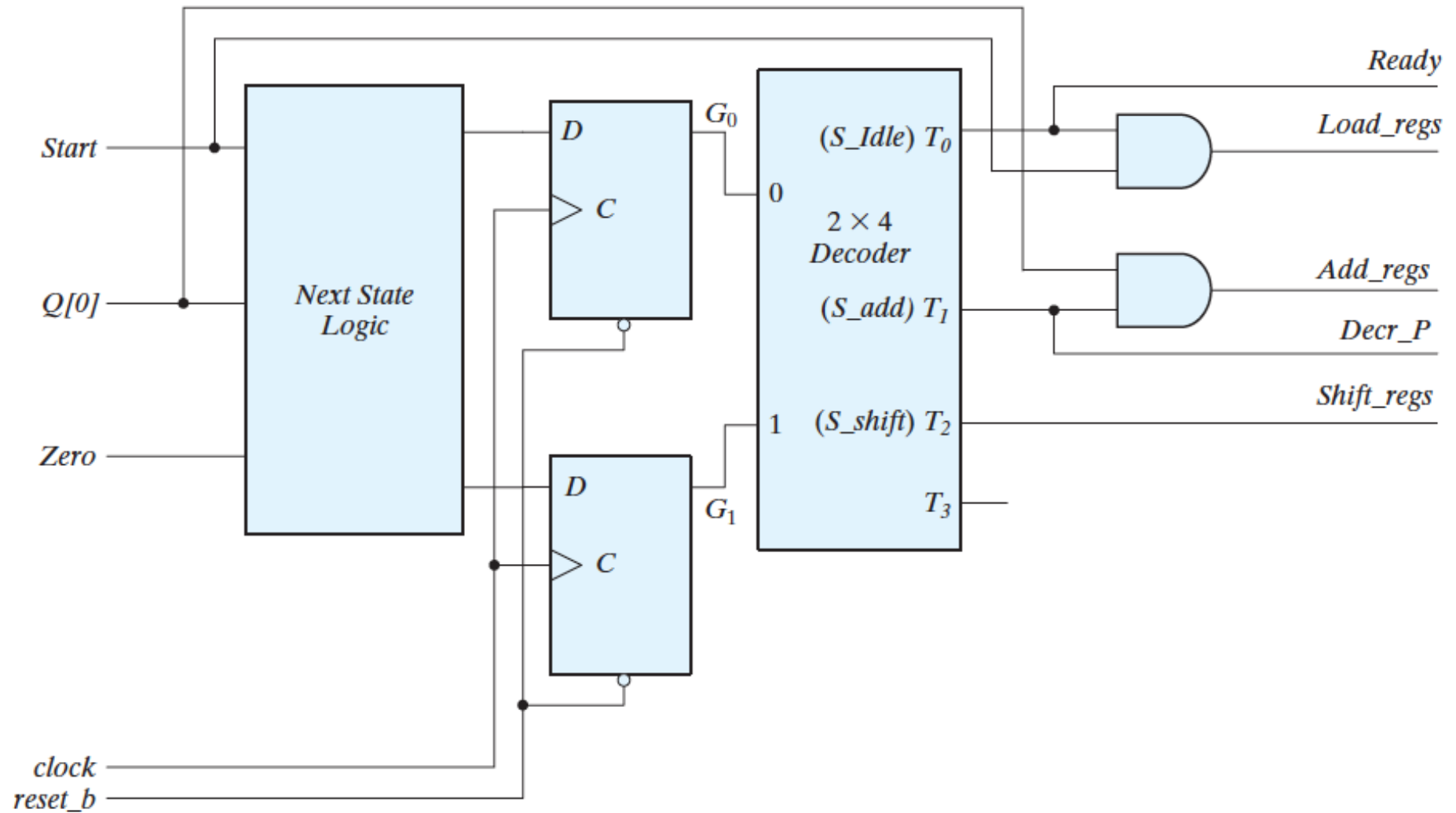
$$\text{Add_regs} = Q_0 \bullet T_1$$

Παράδειγμα 2: Πολλαπλασιαστής

	Present State		Inputs			Next State						
	G_1	G_0	<i>Start</i>	<i>Q[0]</i>	<i>Zero</i>	G_1	G_0	<i>Ready</i>	<i>Load_regs</i>	<i>Decr_P</i>	<i>Add_regs</i>	<i>Shift_regs</i>
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

Shift_regs = T_2

Παράδειγμα 2: Πολλαπλασιαστής



Παράδειγμα 2: Πολλαπλασιαστής

	Present State		Inputs			Next State						
Present-State Symbol	G ₁	G ₀	Start	Q[0]	Zero	G ₁	G ₀	Ready	Load_regs	Decr_P	Add_regs	Shift_regs
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

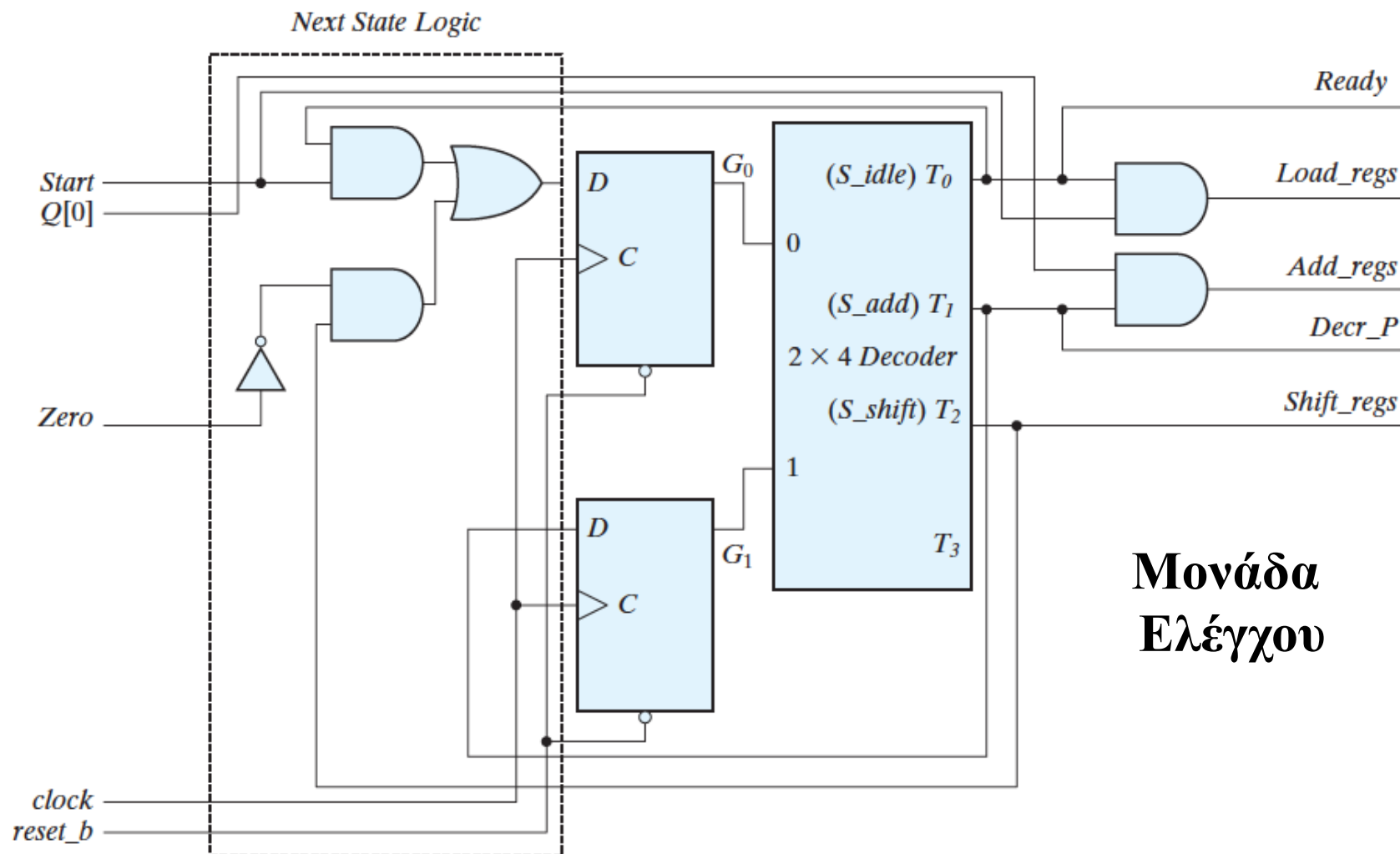
$D_{G1} = T_1$

Παράδειγμα 2: Πολλαπλασιαστής

	Present State		Inputs			Next State						
Present-State Symbol	G ₁	G ₀	Start	Q[0]	Zero	G ₁	G ₀	Ready	Load_regs	Decr_P	Add_regs	Shift_regs
<i>S_idle</i>	0	0	0	X	X	0	0	1	0	0	0	0
<i>S_idle</i>	0	0	1	X	X	0	1	1	1	0	0	0
<i>S_add</i>	0	1	X	0	X	1	0	0	0	1	0	0
<i>S_add</i>	0	1	X	1	X	1	0	0	0	1	1	0
<i>S_shift</i>	1	0	X	X	0	0	1	0	0	0	0	1
<i>S_shift</i>	1	0	X	X	1	0	0	0	0	0	0	1

$$D_{G0} = T_0 \text{Start} + T_2 \text{Zero}'$$

Παράδειγμα 2: Πολλαπλασιαστής



**Μονάδα
Ελέγχου**

Παράδειγμα 2: Πολλαπλασιαστής

Εναλλακτική Σχεδίαση: Κωδικοποίηση One-Hot



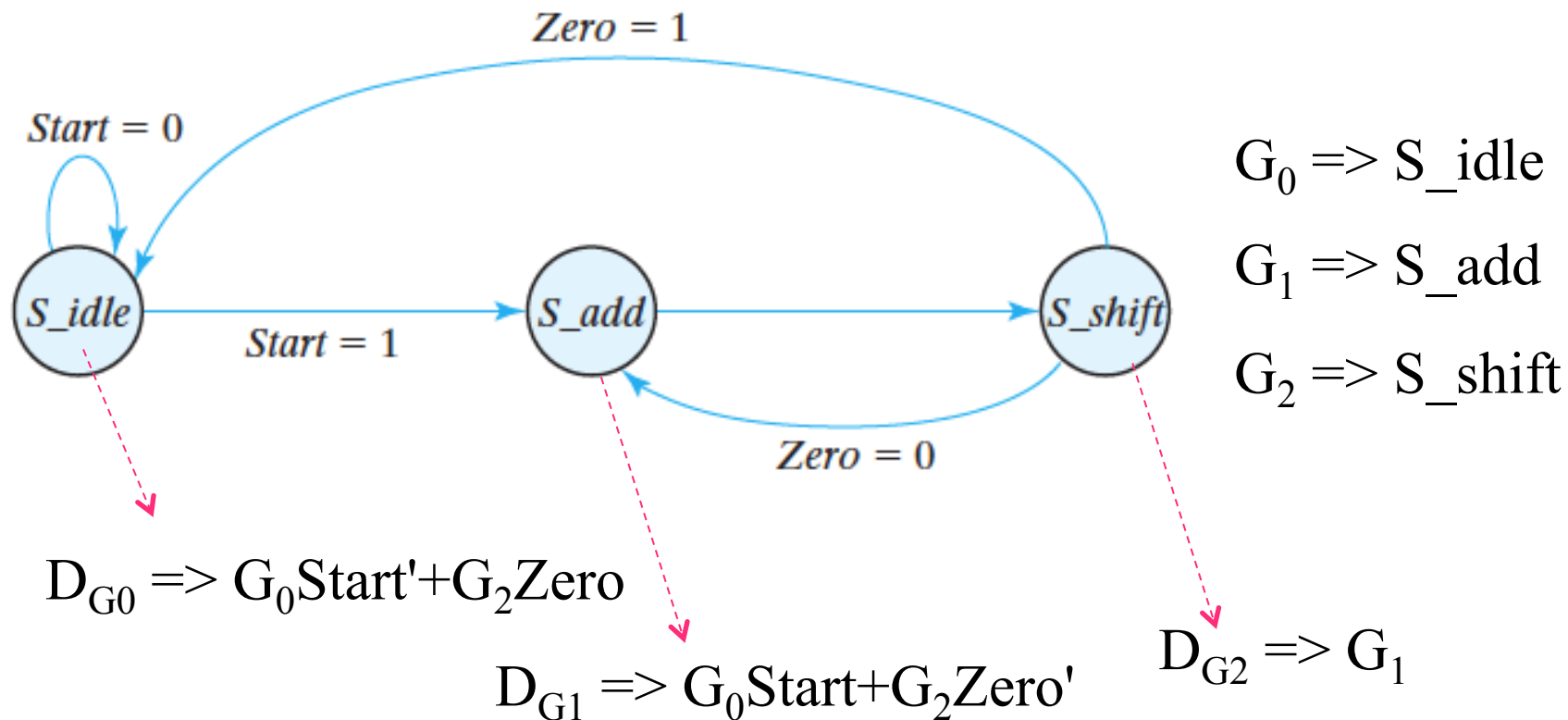
Μόνο ένα flip flop είναι στην μονάδα σε οποιαδήποτε κατάσταση και χρονική στιγμή

Απλή σχεδίαση

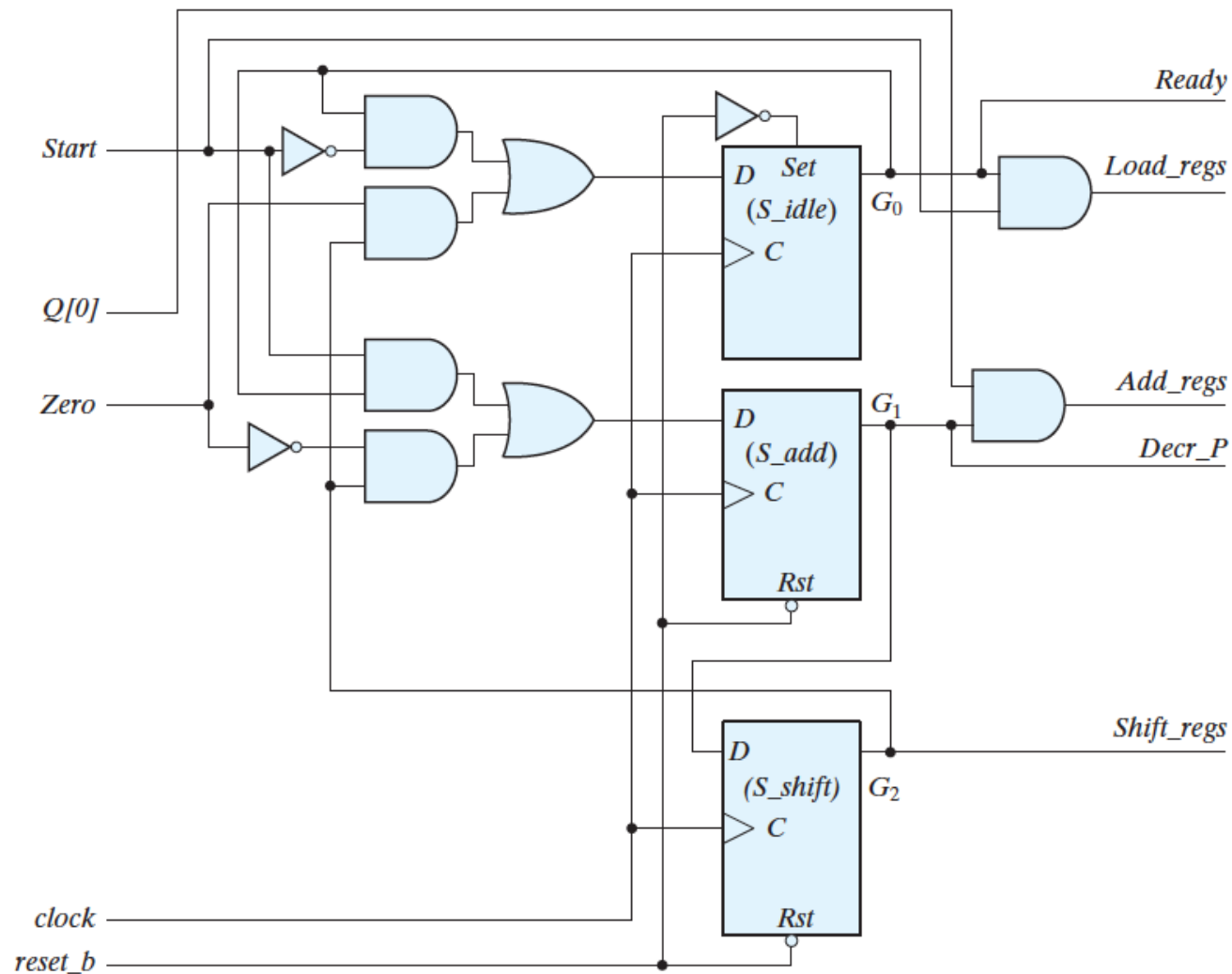
Μειωμένος φόρτος
σχεδίασης

Δεν απαιτείται
αποκωδικοποιητής

Παράδειγμα 2: Πολλαπλασιαστής (One-Hot)

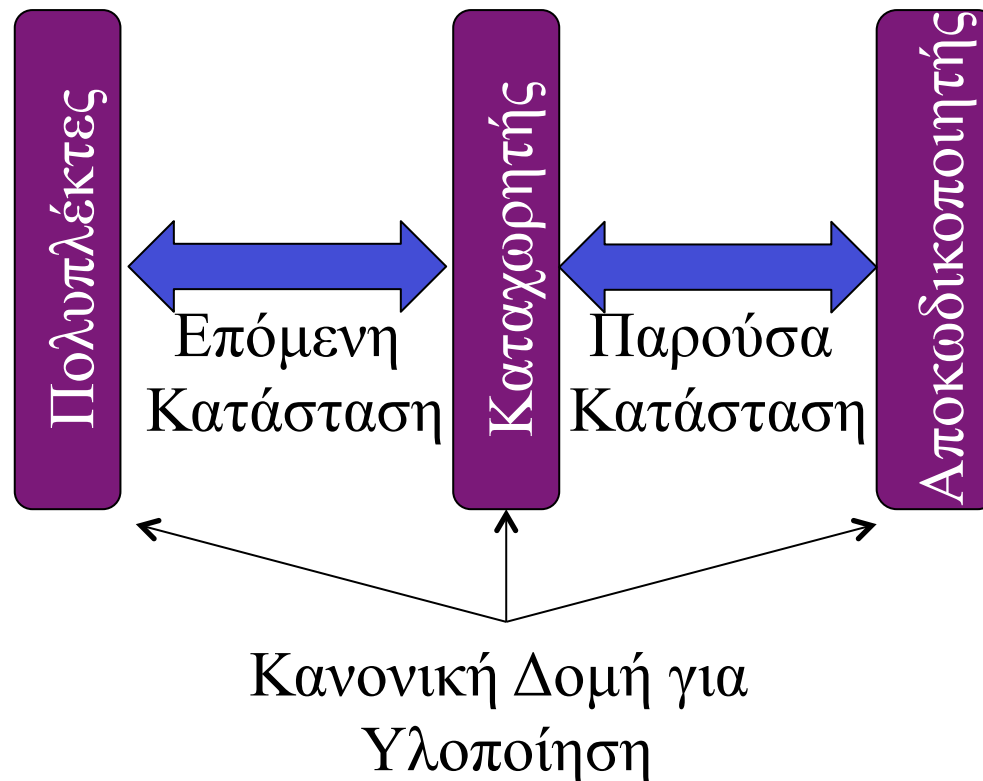


Παράδειγμα 2: Πολλαπλασιαστής (One-Hot)



Σχεδίαση με Πολυπλέκτες

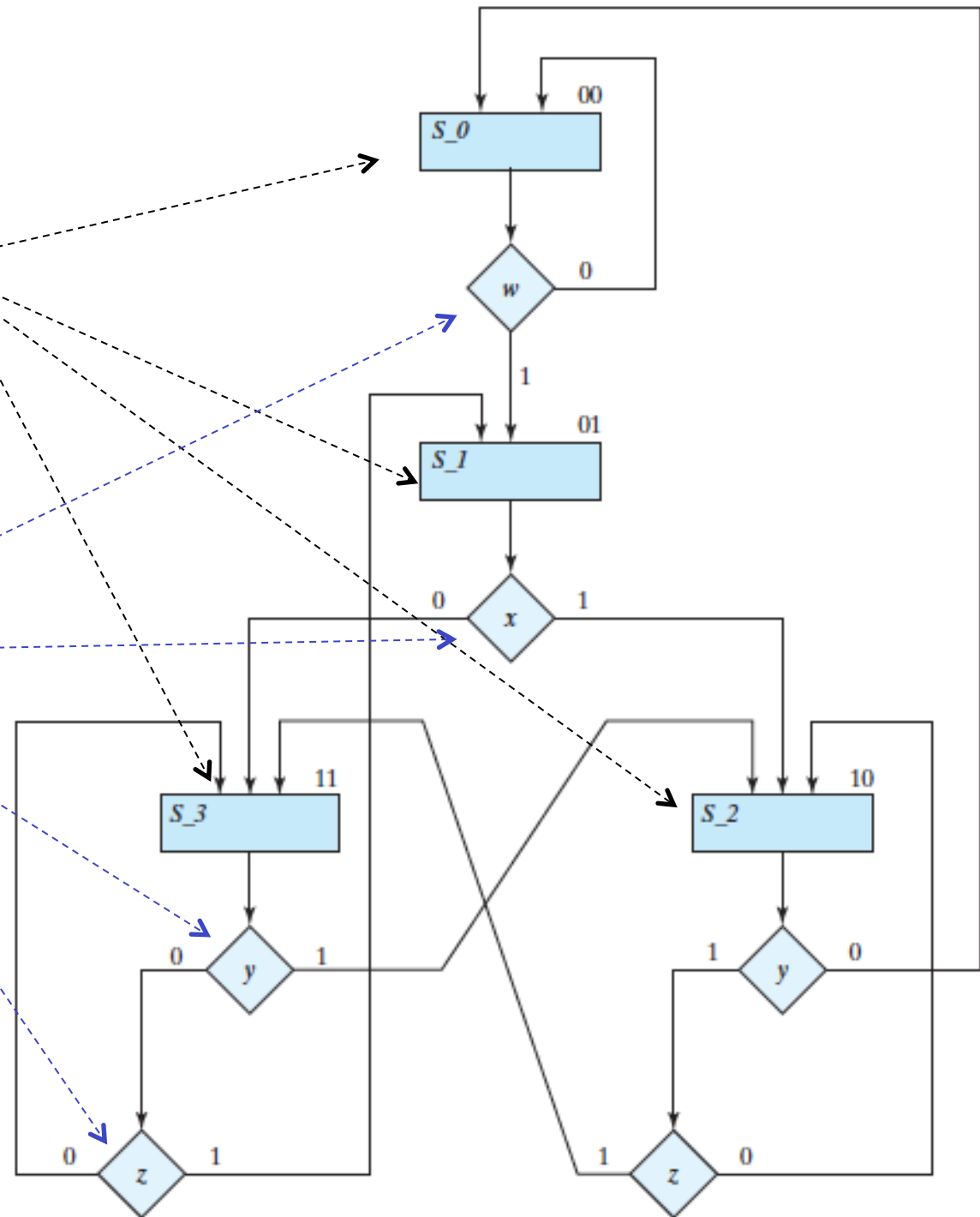
Οι λογικές πύλες που καθορίζουν την επόμενη κατάσταση μπορούν να αντικατασταθούν από πολυπλέκτες



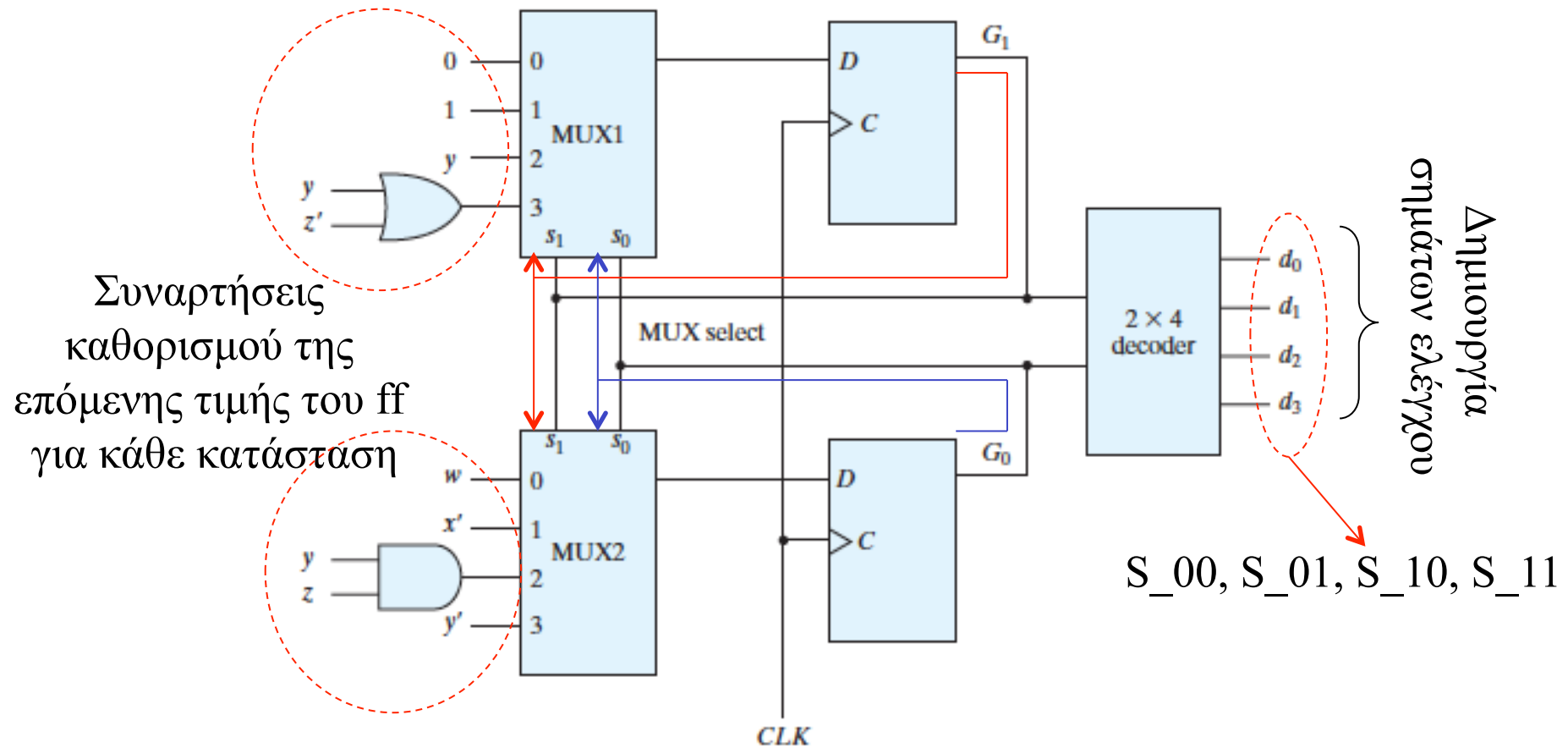
Παράδειγμα

Καταστάσεις

Σήματα
Ελέγχου

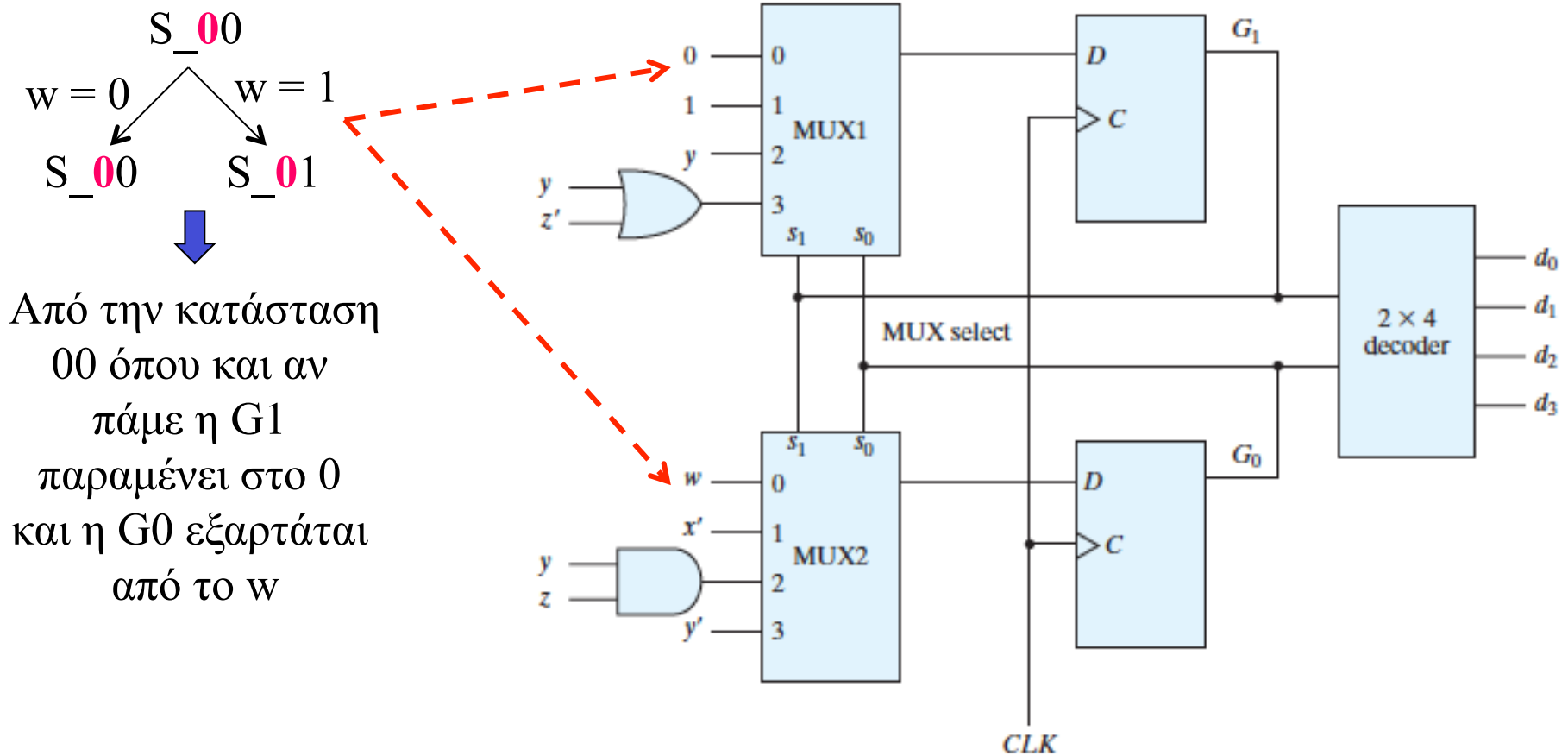


Παράδειγμα



Η επόμενη κατάσταση επιλέγεται κατευθείαν από τους πολυπλέκτες και από τις εισόδους ελέγχου

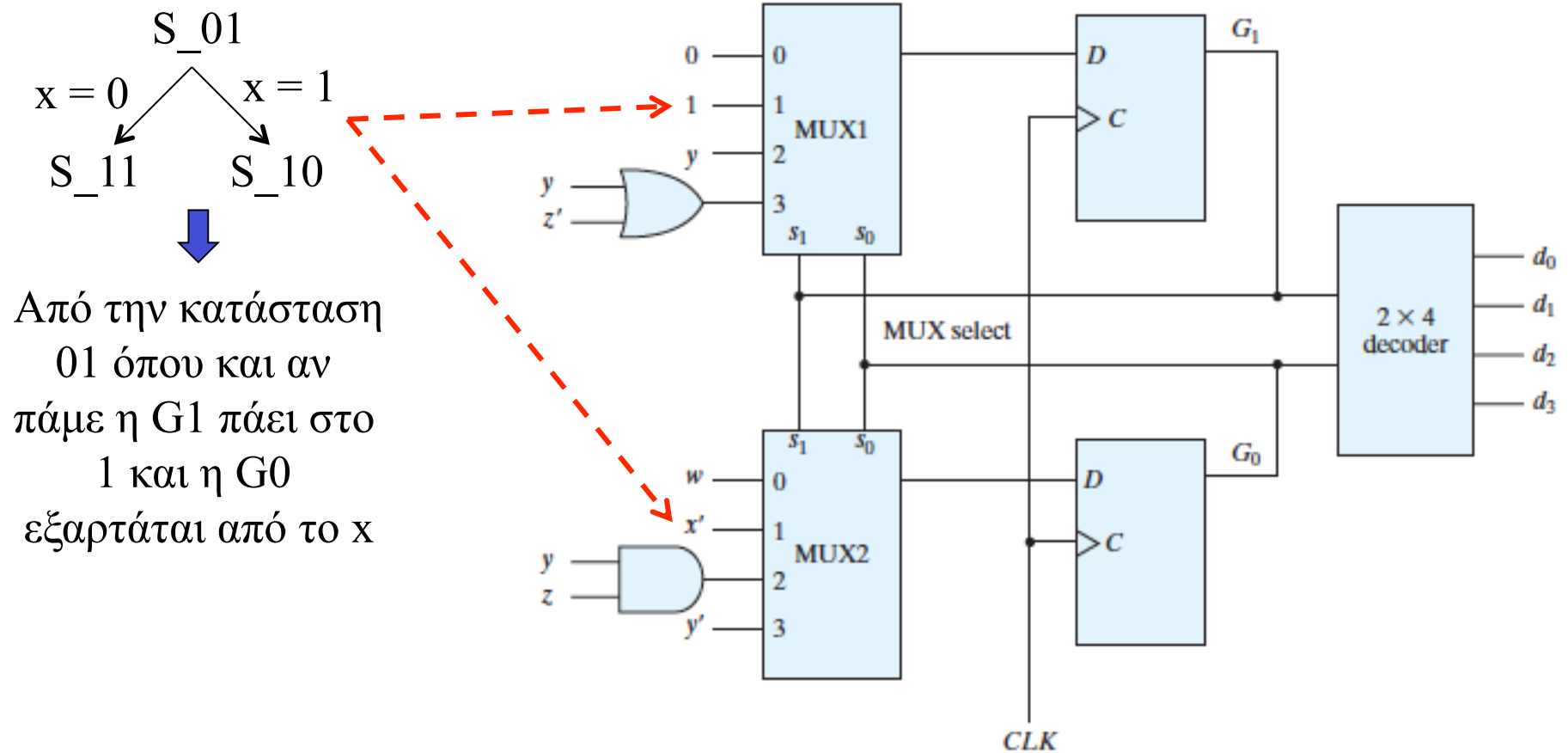
Παράδειγμα



Από την κατάσταση
00 όπου και αν
πάμε η G_1
παράμένει στο 0
και η G_0 εξαρτάται
από το w

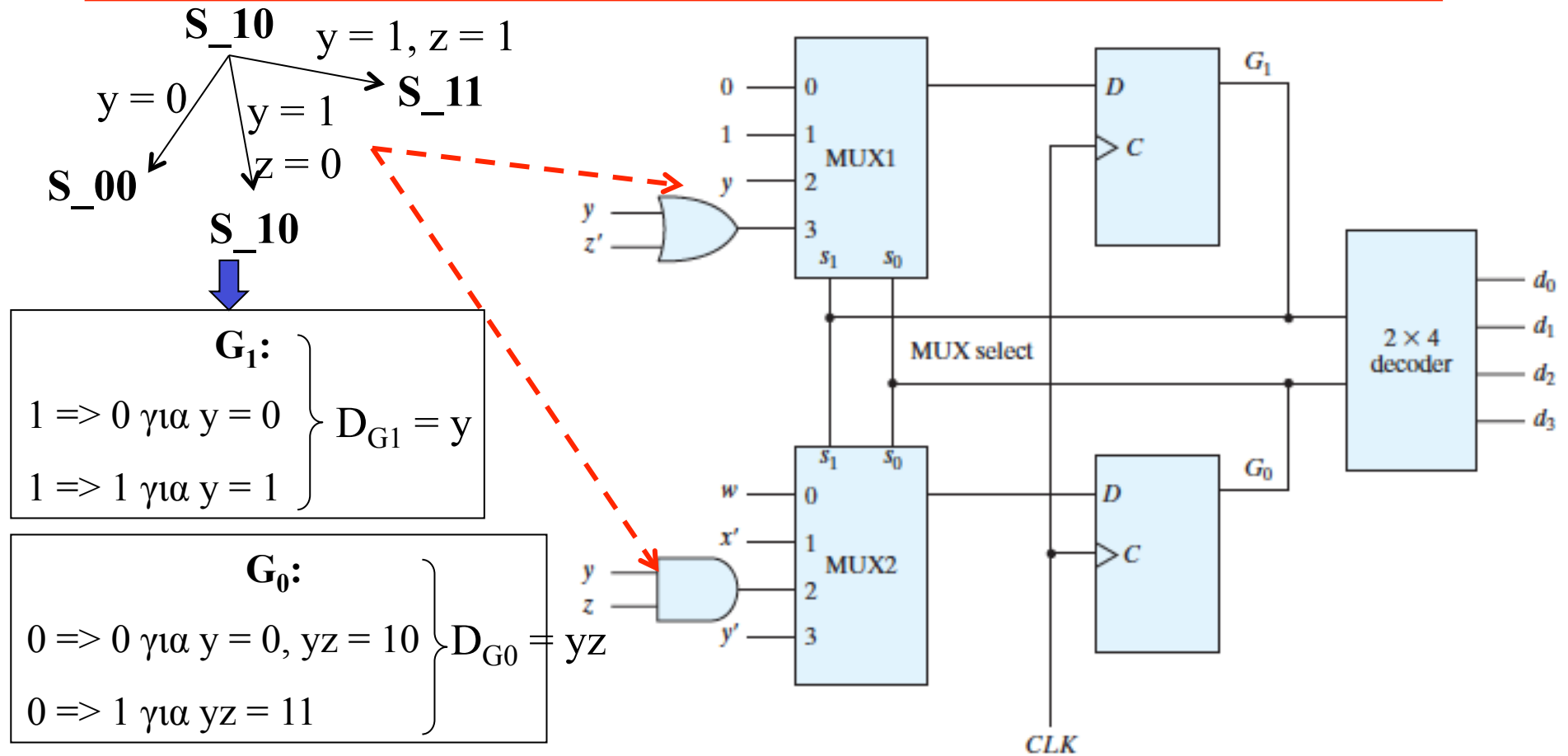
Η μετάβαση των καταστάσεων καθορίζει τις εισόδους των
πολυπλεκτών

Παράδειγμα



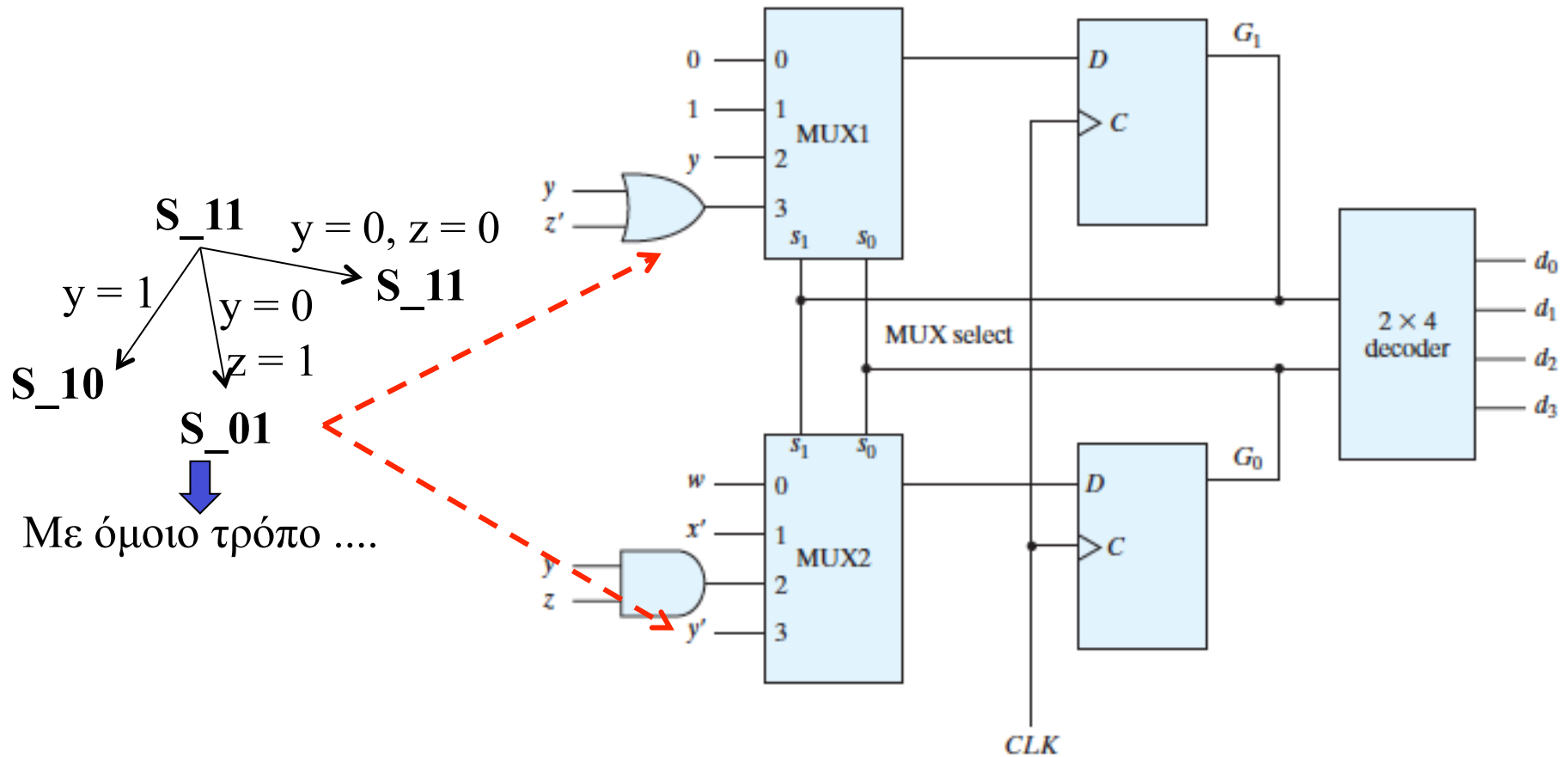
Η μετάβαση των καταστάσεων καθορίζει τις εισόδους των πολυπλεκτών

Παράδειγμα



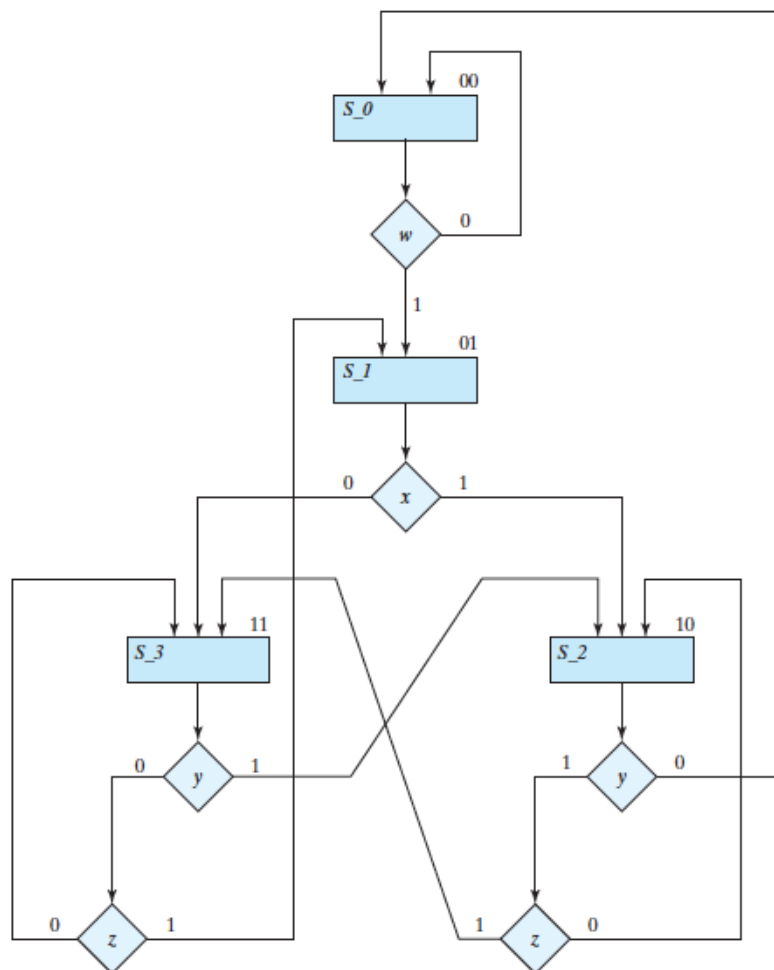
Η μετάβαση των καταστάσεων καθορίζει τις εισόδους των πολυπλεκτών

Παράδειγμα



Με συστηματικό τρόπο μπορούμε να καθορίσουμε οποιαδήποτε μετάβαση. ➡ Πίνακας συνθηκών

Παράδειγμα



Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Πίνακας συνθηκών

Παράδειγμα

Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Μία ομάδα για κάθε είσοδο του/των πολυπλεκτών

Παράδειγμα

Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	→ 0	w
0	1	1	0	x		
0	1	1	1	x'	→ 1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Περίπτωση (α): Μετάβαση σταθερή και ανεξάρτητη συνθήκης

Παράδειγμα

Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Περίπτωση (β): Τοποθετούμε στην είσοδο του πολυπλέκτη την συνάρτηση που δίνει μονάδα (αλλιώς θα είναι 0)

Παράδειγμα

Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz		
					$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Περίπτωση (β): Τοποθετούμε στην είσοδο του πολυπλέκτη την συνάρτηση που δίνει μονάδα (αλλιώς θα είναι 0)

Παράδειγμα

Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Περίπτωση (β): Τοποθετούμε στην είσοδο του πολυπλέκτη την συνάρτηση που δίνει μονάδα (αλλιώς θα είναι 0)

Παράδειγμα

Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Περίπτωση (β): Τοποθετούμε στην είσοδο του πολυπλέκτη την συνάρτηση που δίνει μονάδα (αλλιώς θα είναι 0)

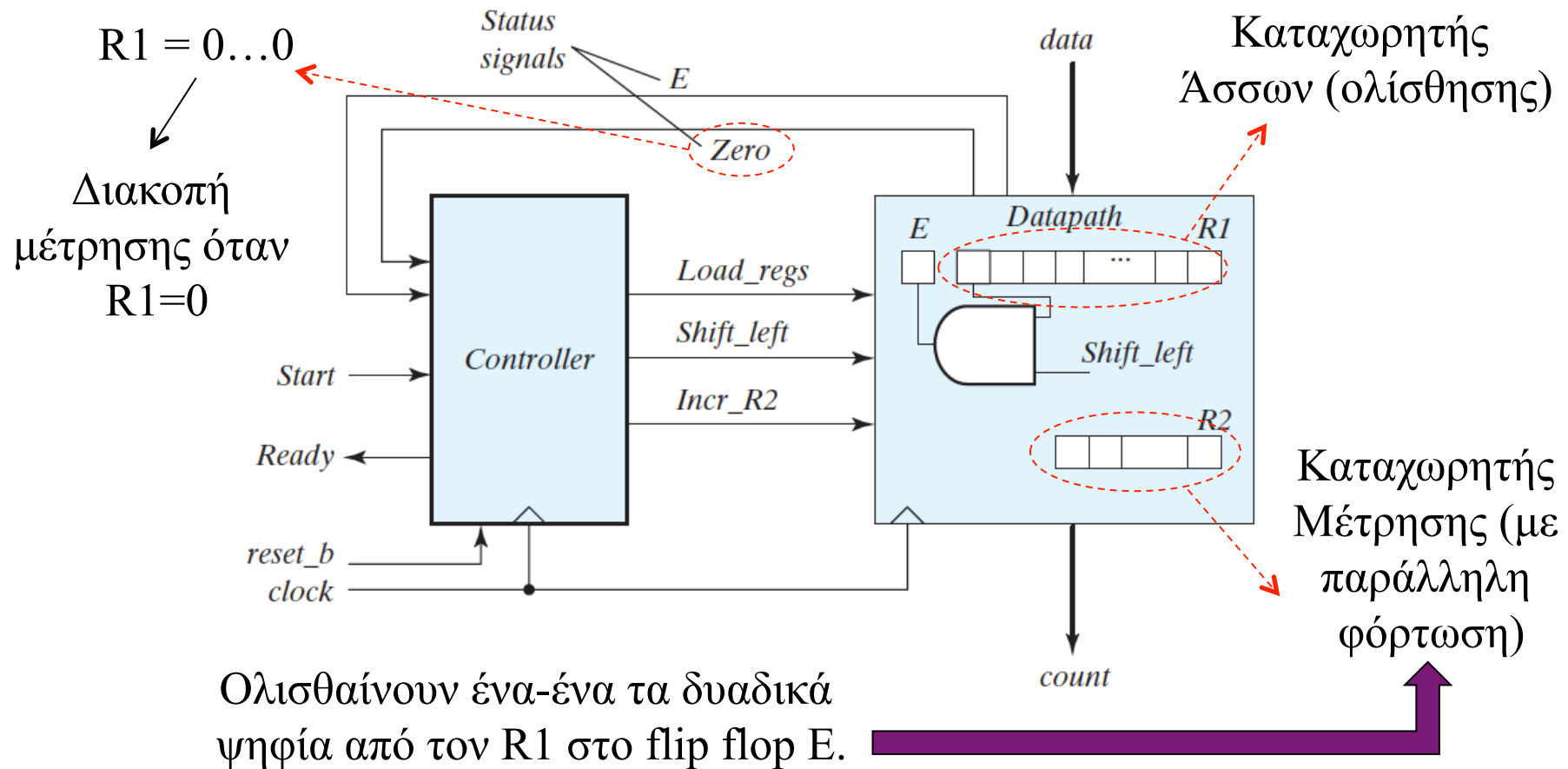
Παράδειγμα

Present State		Next State		Input Condition	Inputs	
G_1	G_0	G_1	G_0	s	MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

Περίπτωση (β): Τοποθετούμε στην είσοδο του πολυπλέκτη την συνάρτηση που δίνει μονάδα (αλλιώς θα είναι 0)

Παράδειγμα (2)

Προδιαγραφές: Σχεδίαση ενός κυκλώματος που θα μετράει τους άσους σε έναν καταχωρητή. Θα γίνει χρήση πολυπλεκτών.

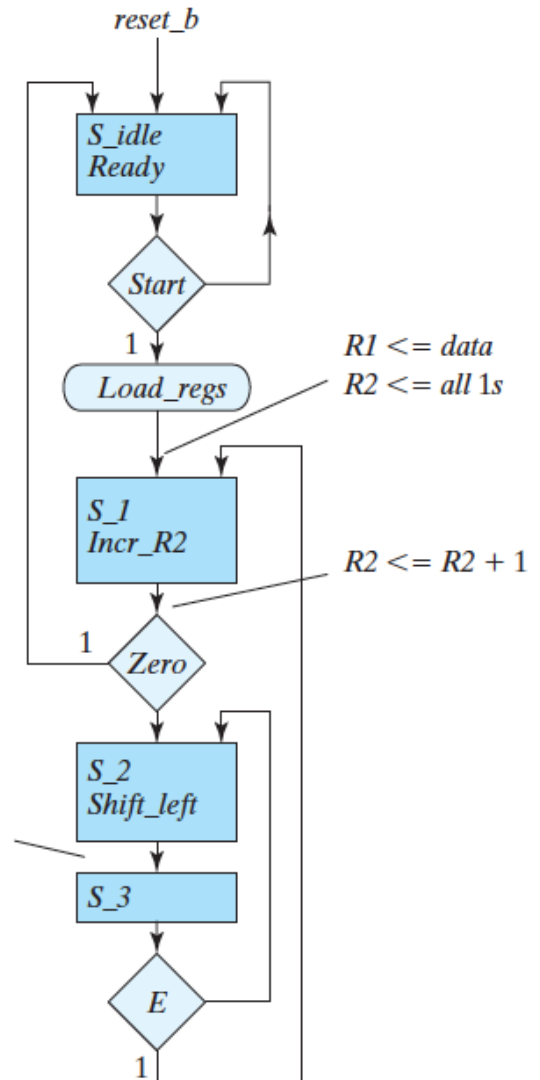
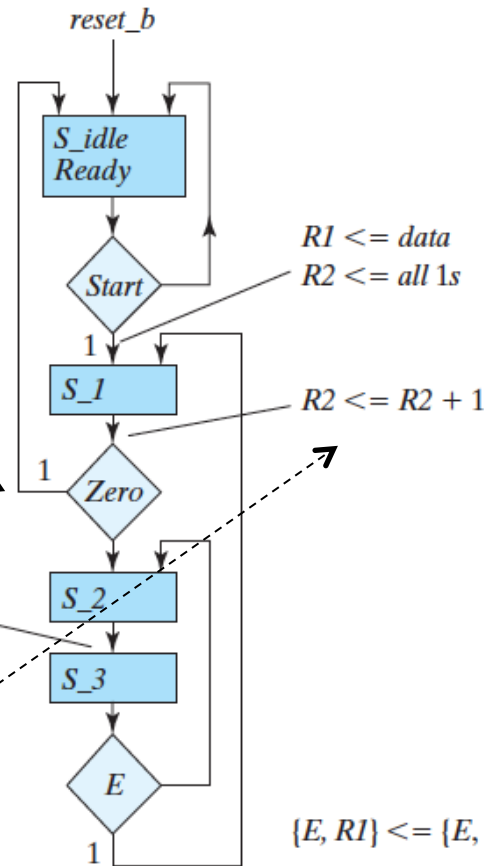


Παράδειγμα (2)

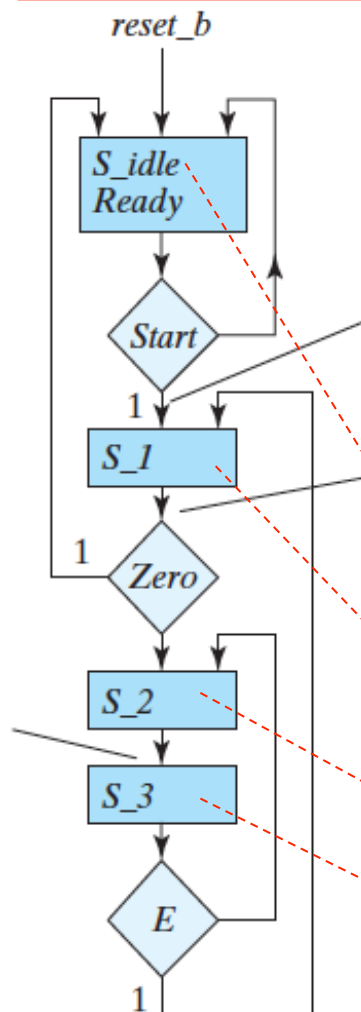
Όταν μείνουν μηδενικά η διαδικασία διακόπτεται

$\{E, RI\} \leq \{E, RI\} \ll 1$

Αρχικά $R2=0$



Παράδειγμα (2)



Πίνακας Συνθηκών
Πολυπλεκτών

Present State		Next State		Input Conditions	Multiplexer Inputs	
G_1	G_0	G_1	G_0		MUX1	MUX2
0	0	0	0	$Start'$		
0	0	0	1	$Start$	0	$Start$
0	1	0	0	$Zero$		
0	1	1	0	$Zero'$	$Zero'$	0
1	0	1	1	$None$	1	1
1	1	1	0	E'		
1	1	0	1	E	E'	E

Παράδειγμα (2)

