



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Ψηφιακή Λογική Σχεδίαση

Επιμέλεια:

Γεώργιος Θεοδωρίδης, Επίκουρος Καθηγητής

Ανδρέας Εμερετλής, Υποψήφιος Διδάκτορας

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη Δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Το παρόν εκπαιδευτικό υλικό αναπτύχθηκε στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Σχεδίαση στο Επίπεδο Μεταφοράς Περιεχομένων Καταχωρητών

- Εισαγωγή
- Βασικές έννοιες και ορολογία
- Αλγοριθμικές μηχανές καταστάσεων (Algorithmic State Machines – ASM )
- Παραδείγματα σχεδιασμών
- Λογικό κύκλωμα ελέγχου
- Σχεδιασμός με πολυπλέκτες
- Σχεδιασμός χωρίς κυνηγητά
- Σχεδιασμός χωρίς μανδαλωτές

- Ένα ρεαλιστικό ψηφιακό κύκλωμα περιέχει μεγάλο αριθμό κυκλωμάτων
  - Μεγάλοι πίνακες αληθείας και διαγράμματα καταστάσεων => **αυξημένη πολυπλοκότητα**
- **Αρθρωτή σχεδίαση – Modular approach** => μείωση της πολυπλοκότητας
  - Χωρισμός & σχεδιασμός σε αυτόνομα υπο-κυκλώματα
- Τυπικά υπο-κυκλώματα
  - Λογικών/Αριθμητικών πράξεων, κύκλωμα ελέγχου, καταχωρητές, πολυπλέκτες
- Διασύνδεση με διαδρόμους δεδομένων και σήματα ελέγχου
- **Σκοπός του κεφαλαίου η μεθοδολογική σχεδίαση σύνθετων κυκλωμάτων και συγκεκριμένα του κυκλώματος ελέγχου**

- Εισαγωγή
- Βασικές έννοιες και ορολογία
- Αλγοριθμικές μηχανές καταστάσεων (Algorithmic State Machines – ASM )
- Παραδείγματα σχεδιασμών
- Λογικό κύκλωμα ελέγχου
- Σχεδιασμός με πολυπλέκτες
- Σχεδιασμός χωρίς κυνηγητά
- Σχεδιασμός χωρίς μανδαλωτές



- Στη γενική μορφή κάθε ψηφιακό κύκλωμα αποτελείται από:
  - **Σύνολο καταχωρητών** για αποθήκευση των δεδομένων, συγχρονισμό των λειτουργιών ...
  - **Σύνολο συνδυαστικών κυκλωμάτων** που εκτελούν αριθμητικές / λογικές πράξεις στα δεδομένα
  - **Κύκλωμα ελέγχου** που ελέγχει / εποπτεύει τη λειτουργία των παραπάνω
  
- Ένα ψηφιακό σύστημα περιγράφεται σε **επίπεδο μεταφοράς (περιεχομένων) καταχωρητών (Register Transfer Level – RTL)** όταν
  - προσδιορίζονται πλήρως τα παραπάνω υποσυστήματα και
  - οι χρονικές λειτουργίες τους
  - χωρίς να περιγράφεται η υλοποίηση αυτών σε **επίπεδο πύλης**

- Αριθμητικές/λογικές πράξεις σε δεδομένα – **συνδυαστικά κυκλώματα** που καλούνται **κυκλώματα χειρισμού δεδομένων (Data path units)**
- Λειτουργίες υλοποιούμενες από καταχωρητές
  - Ολίσθηση/περιστροφή (shifting/rotation), μέτρηση (counting), μηδενισμού δεδομένων (clear), φόρτωσης δεδομένων (load) ...
- Λειτουργίες μεταφοράς δεδομένων
  - Διαύλους μεταφοράς δεδομένων (data buses)
  - Συνδεσμολογίες από πολυπλέκτες και άλλα κυκλώματα (steering logic) για επιλογή διαδρόμων μεταφοράς δεδομένων

## ➤ Χωρίς συνθήκη

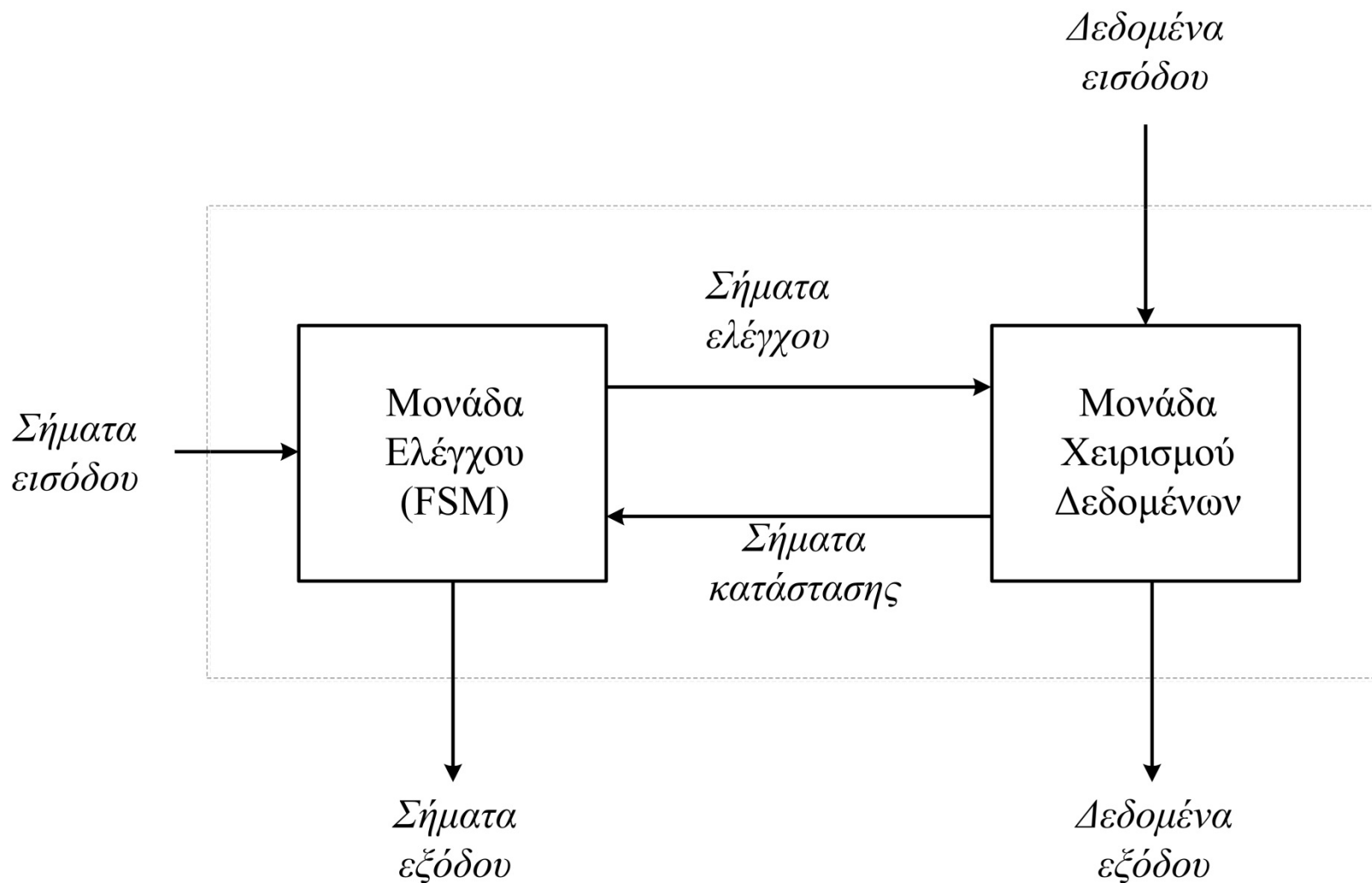
- $R1 <= R1 + R2$
- $R3 <= R2 + 1$
- $R4 <= \text{shr } R4$
- $R2 <= 0$
- .....

## ➤ Με συνθήκη

- If (T1=1) then ( $R2 <= R1$ )
- If (T2 =1) then ( $R4 <= R1 + R4, R3 <= R2$ )
- ..

➤ Σε ένα σύγχρονο σχεδιασμό όλες οι λειτουργίες και οι μεταφορές δεδομένων είναι συγχρονισμένες με ένα σήμα χρονισμού (ρολόι)

- Οι λειτουργίες ενός ψηφιακού συστήματος ελέγχονται από σήματα ελέγχου για τη σωστή χρονική εκτέλεση τους
- Το κύκλωμα ελέγχου είναι ακολουθιακό κύκλωμα –Μηχανή Πεπερασμένων Καταστάσεων (Finite State Machine – FSM)– που:
  - παρακολουθεί την κατάσταση των κυκλωμάτων χειρισμού δεδομένων
  - δέχεται σήματα από το εξωτερικό περιβάλλον
  - λαμβάνει υπόψη την τρέχουσα κατάσταση
- και παράγει σήματα
  - προς τις μονάδες χειρισμού δεδομένων για τον καθορισμό των λειτουργιών
  - εξόδου σχετικά με την κατάσταση του συστήματος
  - καθορίζει την επόμενη κατάσταση του συστήματος



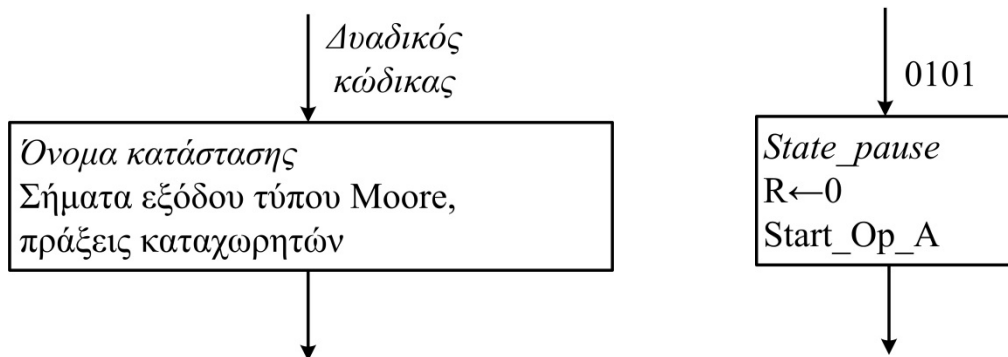
- Εισαγωγή
- Βασικές έννοιες και ορολογία
- Αλγοριθμικές μηχανές καταστάσεων (Algorithmic State Machines – ASM )
- Παραδείγματα σχεδιασμών
- Λογικό κύκλωμα ελέγχου
- Σχεδιασμός με πολυπλέκτες
- Σχεδιασμός χωρίς κυνηγητά
- Σχεδιασμός χωρίς μανδαλωτές

- Τα διαγράμματα **ASM** είναι διαγράμματα ροής για την **περιγραφή της χρονικής διαδοχής των πράξεων/λειτουργιών**
  - Ένα διάγραμμα ASM προσδιορίζει τις λειτουργίες των κυκλωμάτων χειρισμού δεδομένων (data path logic) και Ελέγχου (control logic)
  
- Τα διαγράμματα ASM περιγράφουν
  - Την ακολουθία των γεγονότων / λειτουργιών σε ένα ψηφιακό σύστημα
  - Χρονική σχέση μεταξύ των καταστάσεων του ελεγκτή και των γεγονότων που παρουσιάζονται κατά τη μετάβαση από μια κατάσταση στην επόμενη

- Ένα διάγραμμα ASM αποτελείται από κουτιά:
  - Κατάστασης
  - Απόφασης και
  - Υπό συνθήκη ενεργειών
  
- Περιλαμβάνεις επίσης συνδέσεις μεταξύ των κουτιών που δείχνουν
  - την προτεραιότητα των ενεργειών και
  - τη χρονική εξέλιξη των καταστάσεων του συστήματος

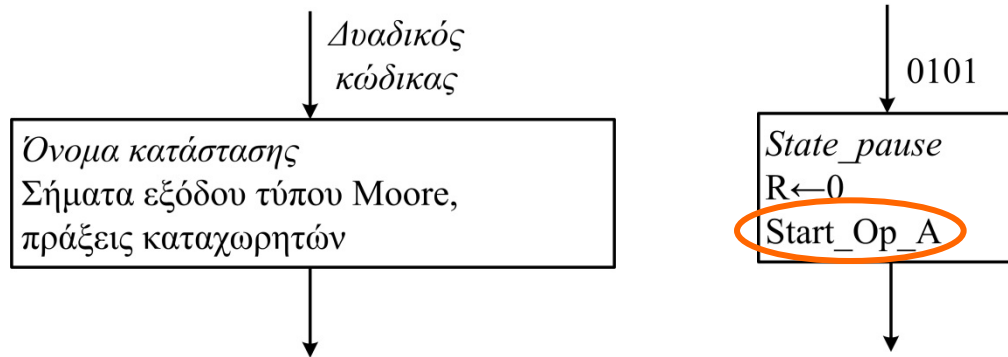


## Κουτί Κατάστασης (1/2)



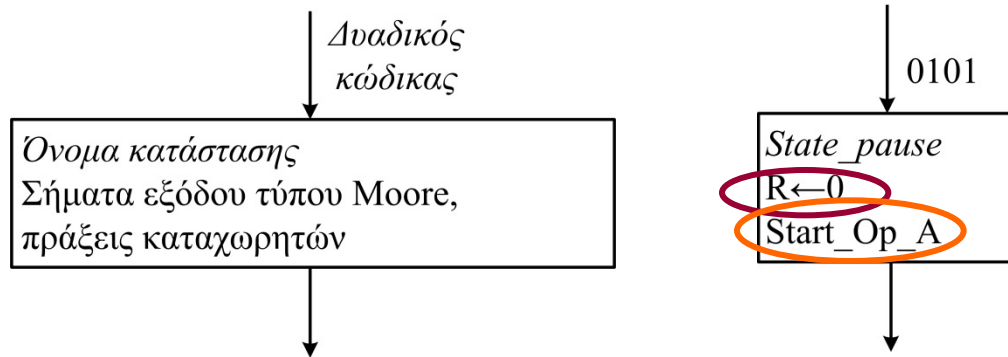
➤ Περιλαμβάνει:

## Κουτί Κατάστασης (1/2)



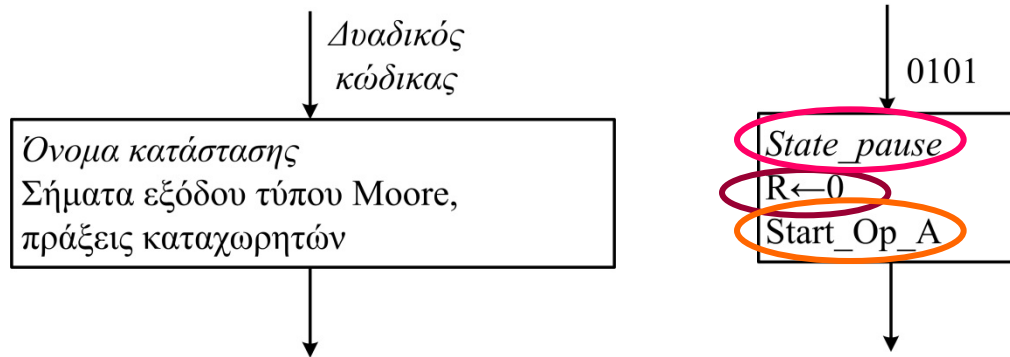
- Περιλαμβάνει:
- Πράξεις καταχωρητών για τη συγκεκριμένη κατάσταση

## Κουτί Κατάστασης (1/2)



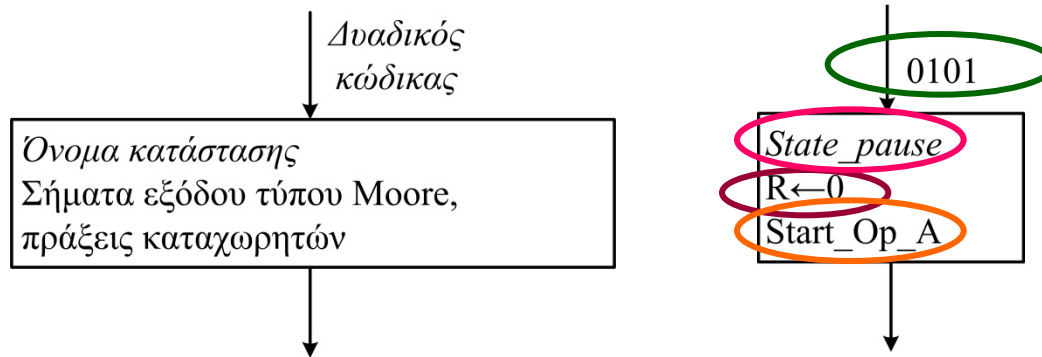
- Περιλαμβάνει:
- Πράξεις καταχωρητών για τη συγκεκριμένη κατάσταση
- Σήματα ελέγχου που αντιστοιχούν στη συγκεκριμένη κατάσταση

## Κουτί Κατάστασης (1/2)



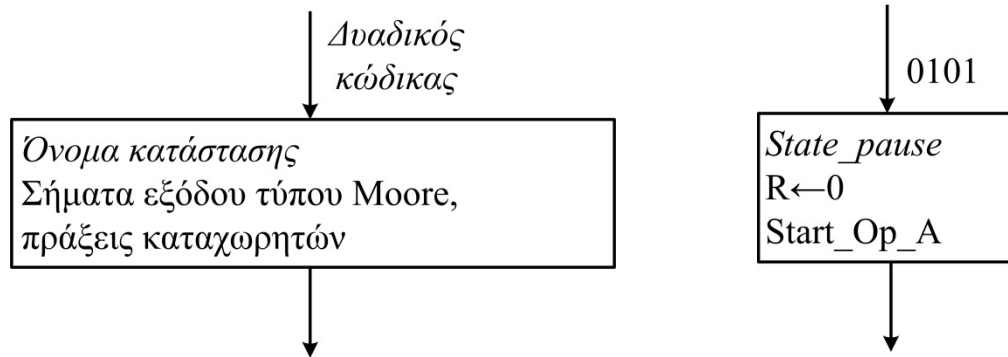
- Περιλαμβάνει:
- Πράξεις καταχωρητών για τη συγκεκριμένη κατάσταση
- Σήματα ελέγχου που αντιστοιχούν στη συγκεκριμένη κατάσταση
- Συμβολικό όνομα της κατάστασης

## Κουτί Κατάστασης (1/2)



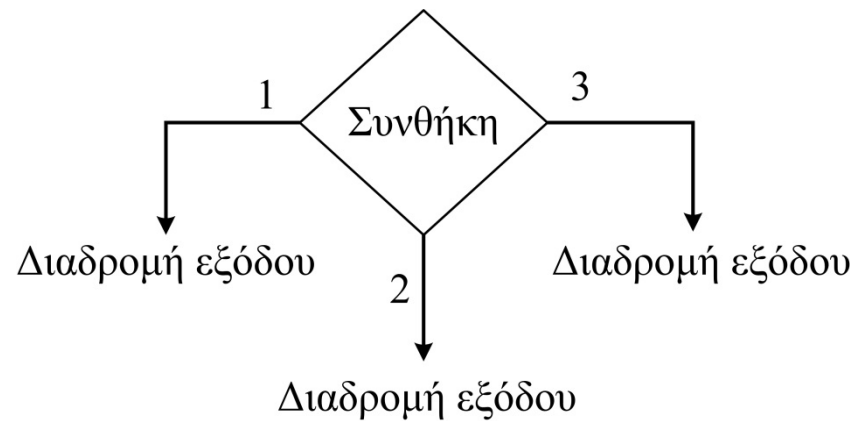
- Περιλαμβάνει:
- Πράξεις καταχωρητών για τη συγκεκριμένη κατάσταση
- Σήματα ελέγχου που αντιστοιχούν στη συγκεκριμένη κατάσταση
- Συμβολικό όνομα της κατάστασης
- Δυαδικό κώδικα της κατάστασης

## Κουτί Κατάστασης (2/2)



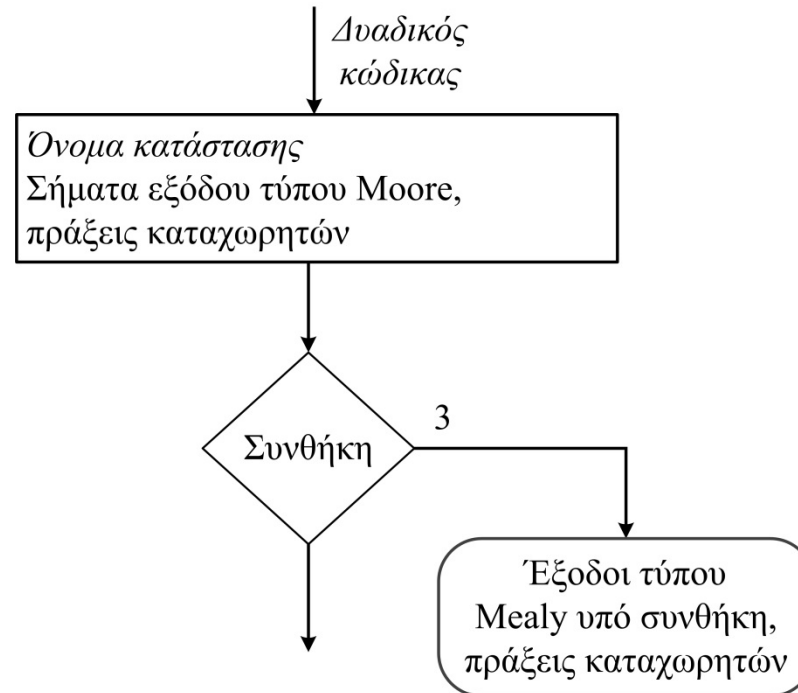
- Οι πράξεις των καταχωρητών είναι σύγχρονες με το ρολόι του συστήματος
  - άρα γίνονται όταν η μηχανή μεταβαίνει από την τρέχουσα κατάσταση στην επόμενη και
  - όχι όταν η μηχανή (σύστημα) βρίσκεται στην τρέχουσα κατάσταση
  
- Η ύπαρξη πράξης σε ένα κουτί κατάστασης σημαίνει ότι το κύκλωμα ελέγχου πρέπει να γεννήσει τα κατάλληλα σήματα ελέγχου για να γίνουν οι πράξεις στην επόμενη μετάβαση του ρολογιού

# Κουτί Απόφασης



- Χρησιμοποιείται για να περιγράψει το αποτέλεσμα που έχει μία είσοδος στο κύκλωμα ελέγχου
- Ελέγχεται μία συνθήκη που σχετίζεται με το συγκεκριμένο σήμα και με βάση το αποτέλεσμα του ελέγχου ακολουθείται το αντίστοιχο μονοπάτι εξόδου

## Κουτί υπό Συνθήκη Ενεργειών (1/2)

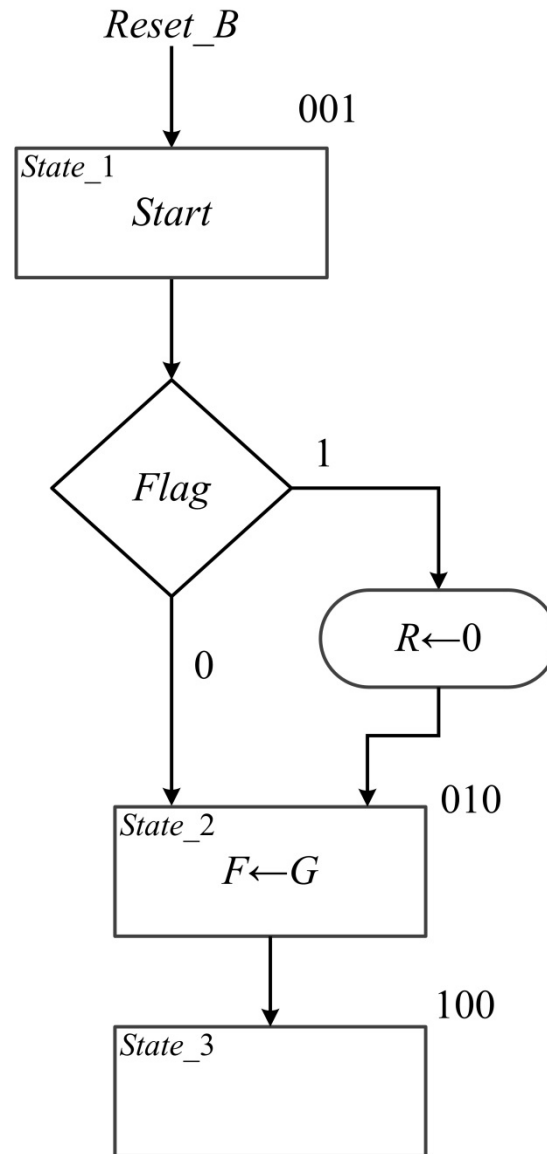


- Είναι παρόμοιο με το κουτί κατάστασης αλλά οι ενέργειες που περιγράφει εκτελούνται όταν ικανοποιείται μία συνθήκη
  - Έχει πάντα είσοδο μία έξοδο ενός κουτιού απόφασης

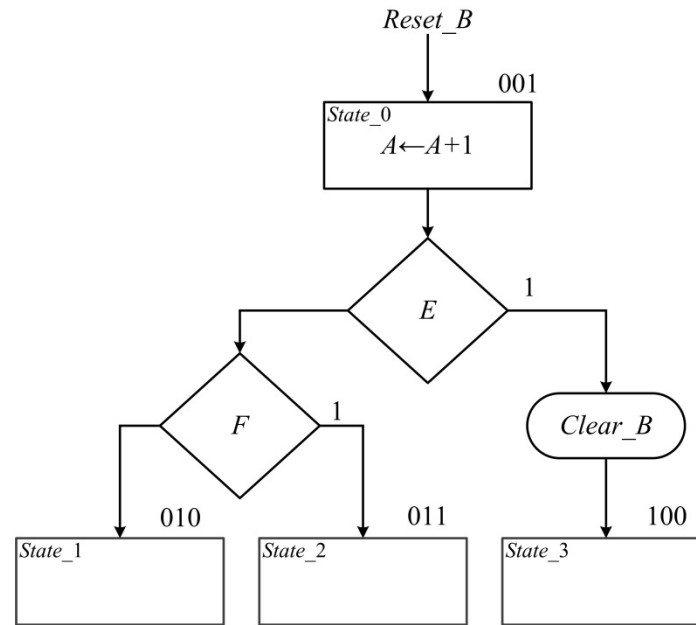
- Οι έξοδοι είναι τύπου Mealy



# Κουτί υπό Συνθήκη Ενεργειών (2/2)

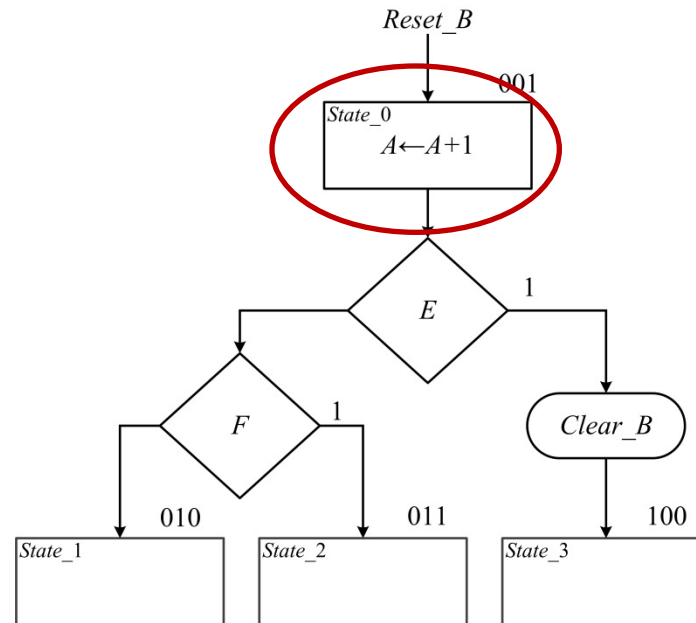


# ASM Block (1/3)



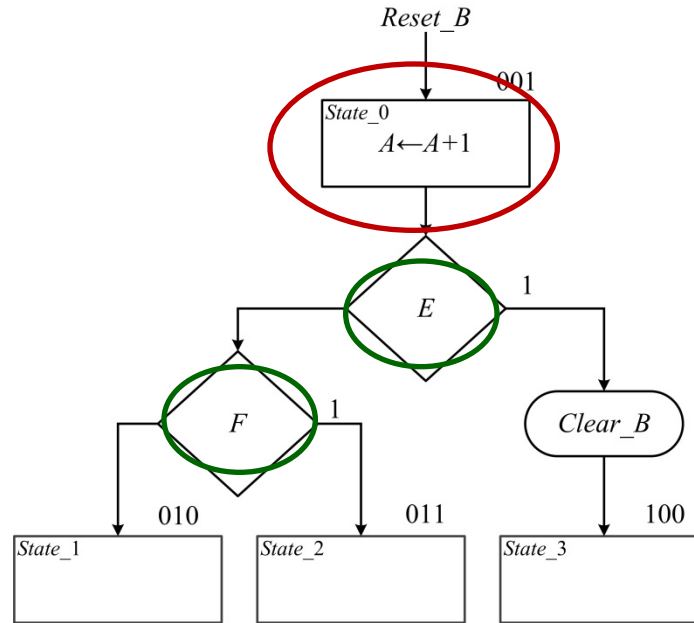
➤ Το μπλοκ ASM είναι μία δομή που αποτελείται από

# ASM Block (1/3)



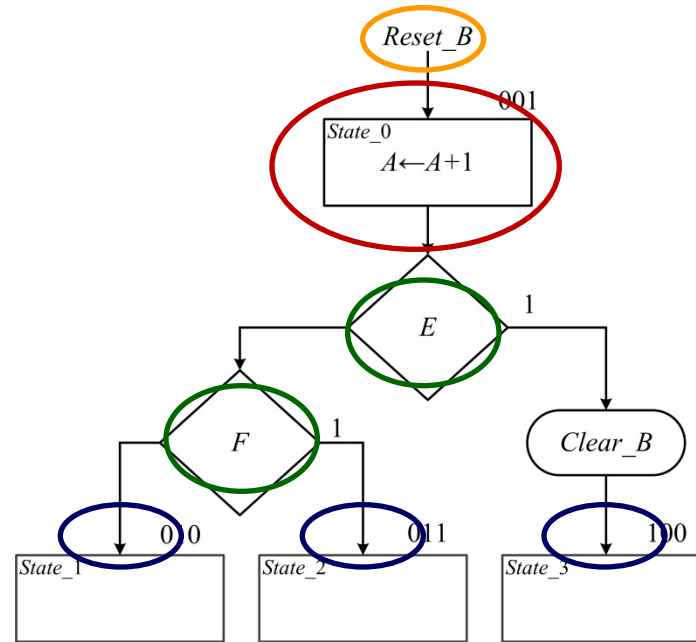
- Το μπλοκ ASM είναι μία δομή που αποτελείται από
  - ένα κουτί κατάστασης

## ASM Block (1/3)



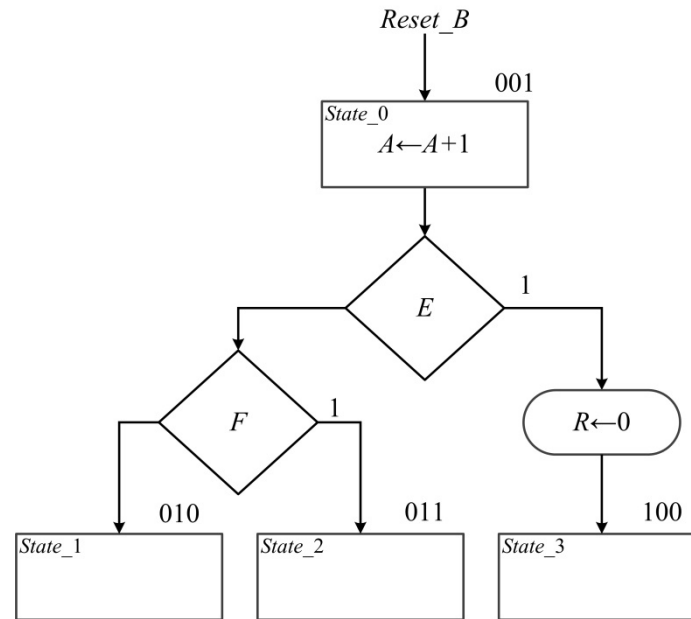
- Το μπλοκ ASM είναι μία δομή που αποτελείται από
  - ένα κουτί κατάστασης
  - όλα τα κουτιά αποφάσεων και ενεργειών που συνδέονται με τις γραμμές εξόδου του κουτιού κατάστασης

## ASM Block (1/3)



- Το μπλοκ ASM είναι μία δομή που αποτελείται από
  - ένα κουτί κατάστασης
  - όλα τα κουτιά αποφάσεων και ενεργειών που συνδέονται με τις γραμμές εξόδου του κουτιού κατάστασης
- Ένα μπλοκ ASM έχει μία είσοδο και πιθανώς πολλές εξόδους
- Ένα διάγραμμα ASM αποτελείται από πολλαπλά μπλοκ ASM

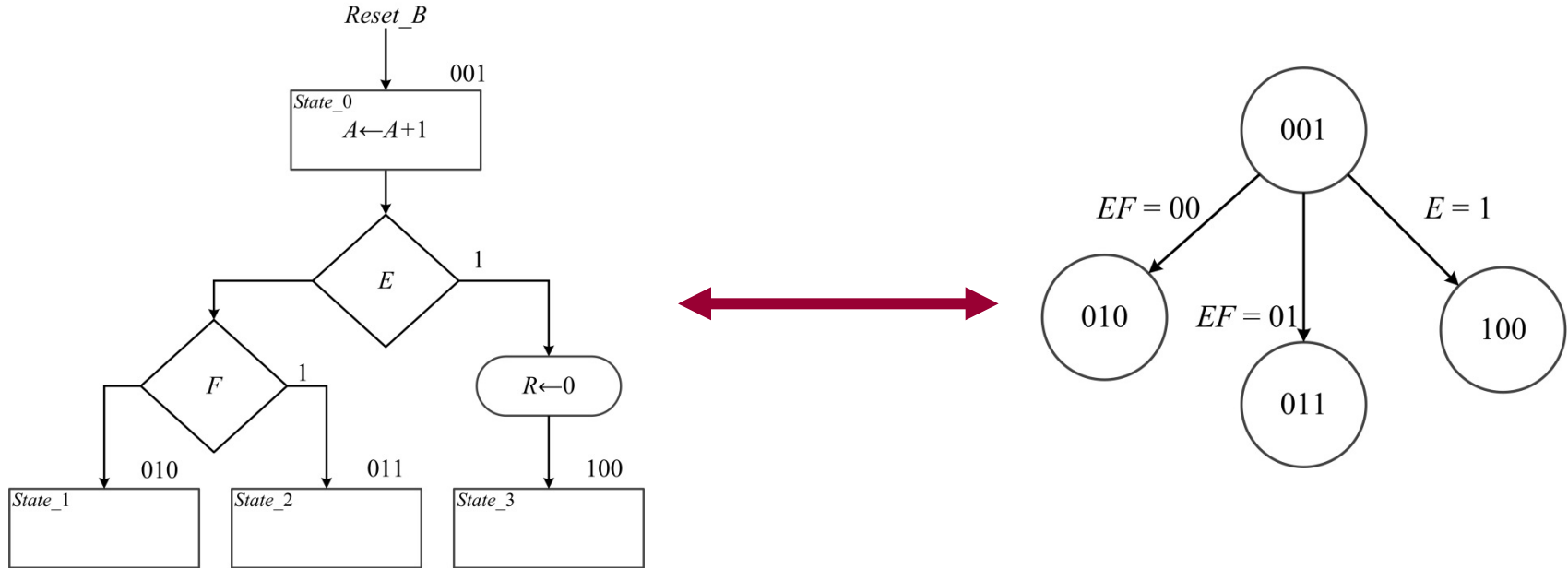
## ASM Block (2/3)



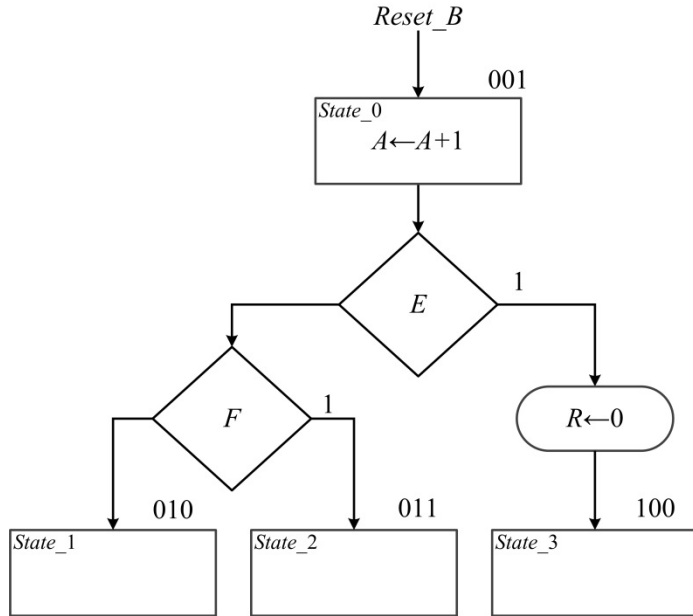
➤ Κάθε μπλοκ ASM περιγράφει την κατάσταση του συστήματος κατά τη διάρκεια ενός παλμού ρολογιού

➤ Οι πράξεις καταχωρητών (κουτιά κατάστασης και υπό συνθήκη ενεργειών) πυροδοτούνται από τον ίδιο παλμό ρολογιού όταν το σύστημα αλλάζει κατάσταση (State\_0 → επόμενη)

–Ανάλογα με τις τιμές των E, F το σύστημα μεταβαίνει σε μία από τις State\_1, State\_2, State\_3



- Αρκετές φορές είναι πιο βολικό να μετατρέπεται το μπλοκ ASM ή το διάγραμμα ASM σε διάγραμμα καταστάσεων
- Χρησιμοποιώντας το διάγραμμα καταστάσεων και γνώστες μεθοδολογίες μπορεί να σχεδιαστεί το ακολουθιακό κύκλωμα του ελεγκτή



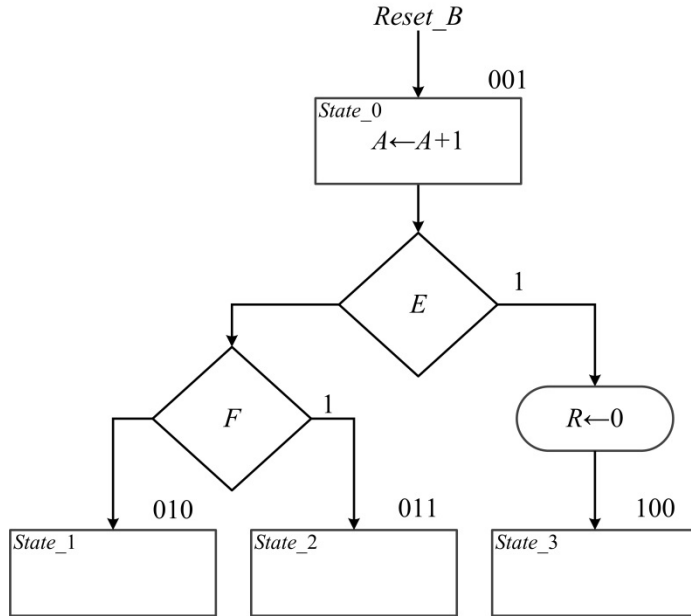
➤ Η κύρια διαφορά ενός συμβατικού διαγράμματος και ενός ASM βρίσκεται στη χρονική σχέση μεταξύ των διαφόρων ενεργειών και πράξεων καταχωρητών που περιγράφει

➤ Σε ένα συμβατικό διάγραμμα οι πράξεις θα εκτελούνταν ακολουθιακά  
–Αύξηση του A ( $A=A+1$ ), έλεγχος του E, έλεγχος του F ....

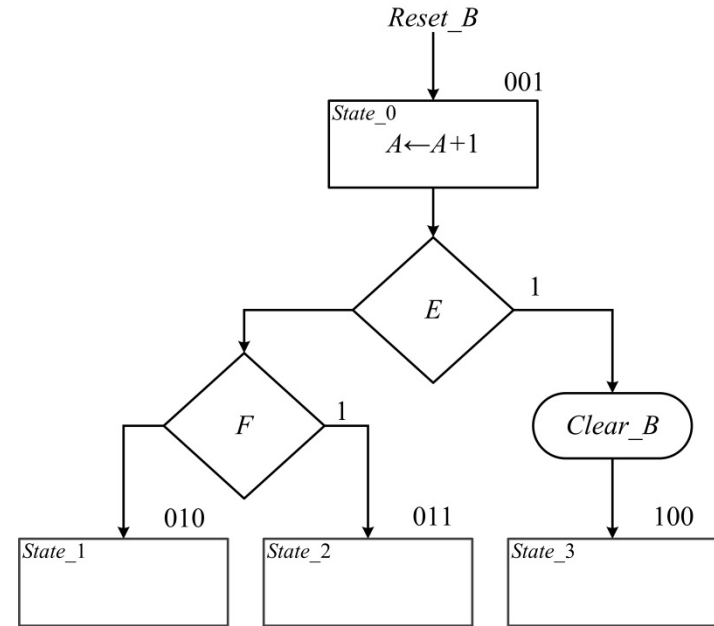
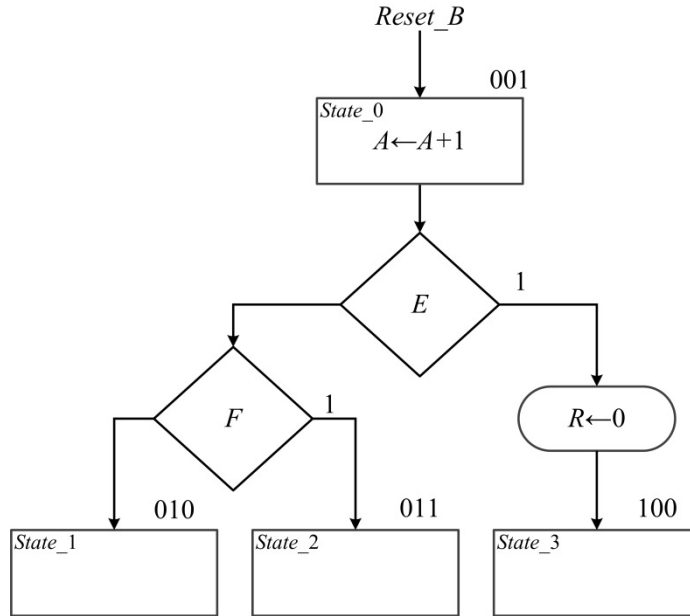
➤ Σε ένα διάγραμμα ASM κάθε μπλοκ είναι μία ενιαία μονάδα

➤ Όλες οι πράξεις καταχωρητών εκτελούνται ταυτόχρονα σε συγχρονισμό με την ακμή του ρολογιού

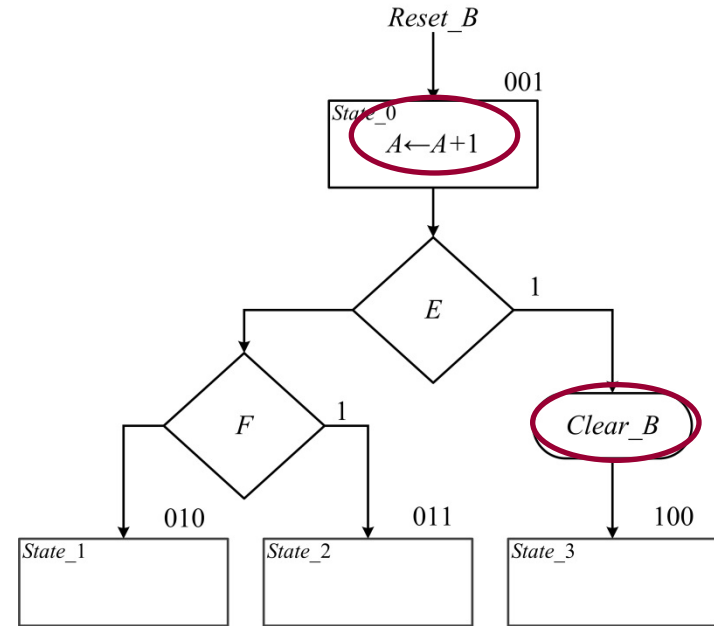
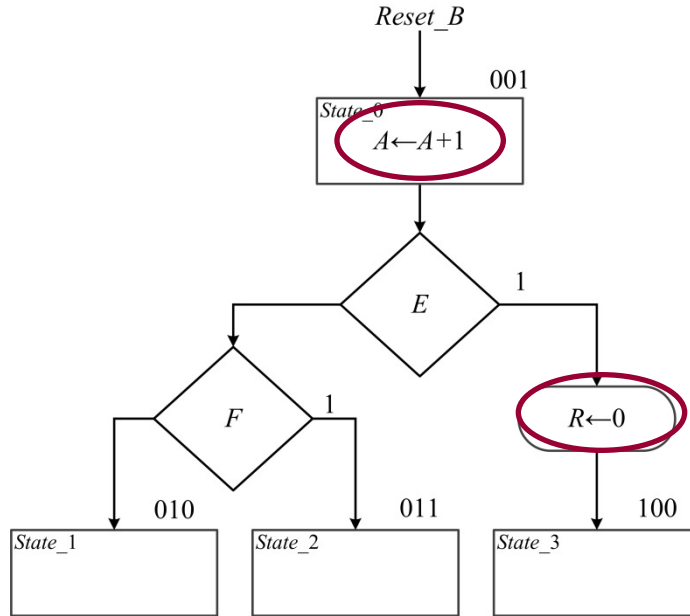




- Αν το  $Reset\_B=1$  τότε το κύκλωμα μεταβαίνει στην  $State\_0$
- Στην επόμενο κύκλο ρολογιού γίνονται τα παρακάτω
  - $A=A+1$
  - Αν  $E=1$  τότε  $R=0$  και επόμενη κατάσταση είναι η  $State\_3$
  - Αν  $F=0$  (1) η επόμενη κατάσταση είναι η  $State\_1$  ( $State\_2$ )



- Στο αρχικό διάγραμμα ASM δείχνονται οι πράξεις καταχωρητών που πρέπει να εκτελεστούν από τα κυκλώματα χειρισμού δεδομένων
- Δε δείχνονται τα σήματα ελέγχου πρέπει να γεννήσει το κύκλωμα ελέγχου



- Στο αρχικό διάγραμμα ASM δείχνονται οι πράξεις καταχωρητών που πρέπει να εκτελεστούν από τα κυκλώματα χειρισμού δεδομένων
- Δε δείχνονται τα σήματα ελέγχου πρέπει να γεννήσει το κύκλωμα ελέγχου
- Η εξαγωγή των σημάτων ελέγχου και η τροποποίηση του διαγράμματος είναι άμεση

➤ Το διάγραμμα **ASMD** (Algorithmic State Machine and Datapath) διαφέρει από το **ASM** στα ακόλουθα:

- Στο **ASMD** δε γράφονται πράξεις καταχωρητών μέσα στα κουτιά καταστάσεων
- **Οι πράξεις καταχωρητών γράφονται στα βέλη μεταβάσεων καταστάσεων στις οποίες συμβαίνουν**
- Περιλαμβάνει κουτιά ενεργειών υπό συνθήκη όπου καθορίζονται τα σήματα με τα οποία ελέγχονται οι πράξεις καταχωρητών που γράφονται δίπλα στα βέλη μεταβάσεων

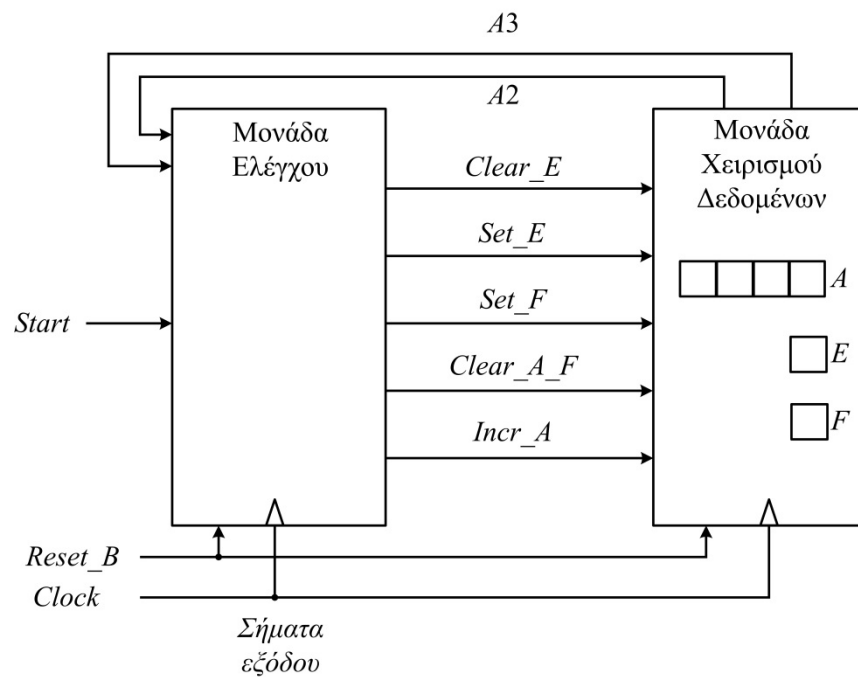
➤ Με τα διαγράμματα **ASMD** αποσαφηνίζονται όλα τα προβλήματα των διαγραμμάτων **ASM** που σχετίζονται με τη χρονική εκτέλεση των λειτουργιών

- 1. Σχηματισμός του ASM όπου εμφανίζονται μόνο οι είσοδοι του ελεγκτή που καθορίζουν τις μεταβάσεις καταστάσεων
  
- 2. Μετατροπή του ASM σε ASMD
  - γράφοντας δίπλα στα βέλη των μεταβάσεων καταστάσεων πληροφορία που αφορά την ταυτόχρονη εκτέλεση πράξεων καταχωρητών
  
- 3. Τροποποίηση του ASMD για να προσδιοριστούν τα απαιτούμενα σήματα ελέγχου για την υλοποίηση των ζητούμενων λειτουργιών
  
- Με τον τρόπο αυτό καθορίζονται πλήρως οι λειτουργίες των κυκλωμάτων ελέγχου και χειρισμού δεδομένων

- Εισαγωγή
- Βασικές έννοιες και ορολογία
- Αλγοριθμικές μηχανές καταστάσεων (Algorithmic State Machines – ASM )
- Παραδείγματα σχεδιασμών
- Λογικό κύκλωμα ελέγχου
- Σχεδιασμός με πολυπλέκτες
- Σχεδιασμός χωρίς κυνηγητά
- Σχεδιασμός χωρίς μανδαλωτές

- Μονάδα χειρισμού δεδομένων που περιλαμβάνει
  - 2 JK F/Fs (**E**, **F**) & 1 counter **A** [A3, A2, A1, A0]
  
- **Σήματα & λειτουργία μονάδας**
  
- Σήμα Start
  - Αν start=1 ξεκινά η λειτουργία και γίνονται  $A=0$  και  $F=0$
  
- Σε κάθε επόμενο παλμό  $A=A+1$
  
- Συνθήκες ελέγχου
  - Αν  $A_2=0$  τότε  $E=0$
  - Αν  $A_2=1$  τότε  $E=1$ , αν  $A_3=0$  η μέτρηση συνεχίζεται
  - Αν  $A_3=1$  τότε  $F=1$  (στον επόμενο παλμό) και σταματά η μέτρηση
  - Αν start = 0 το σύστημα κάνει hold, αλλιώς αν start = 1 επαναλαμβάνει τη λειτουργία

# Σχηματικό Διάγραμμα του Κυκλώματος



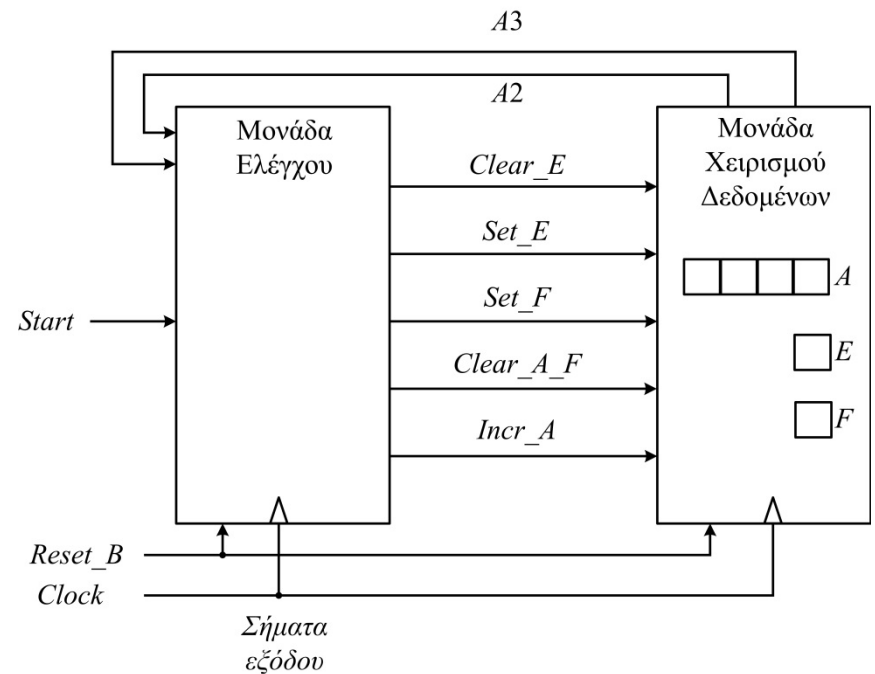


## ► Start

- Αν  $start = 1$  ξεκινά η λειτουργία και γίνονται  $A = 0$  και  $F = 0$
- Σε κάθε επόμενο παλμό  $A = A+1$  (counter)

## ► Συνθήκες ελέγχου

- Αν  $A2 = 0$  τότε  $E = 0$
- Αν  $A2 = 1$  τότε  $E = 1$ , αν  $A3 = 0$  η μέτρηση συνεχίζεται
- Αν  $A3 = 1$  τότε  $F = 1$  (στον επόμενο παλμό) και σταματά η μέτρηση
- Αν  $start = 0$  το σύστημα κάνει hold, αλλιώς αν  $start = 1$  επαναλαμβάνει τη λειτουργία

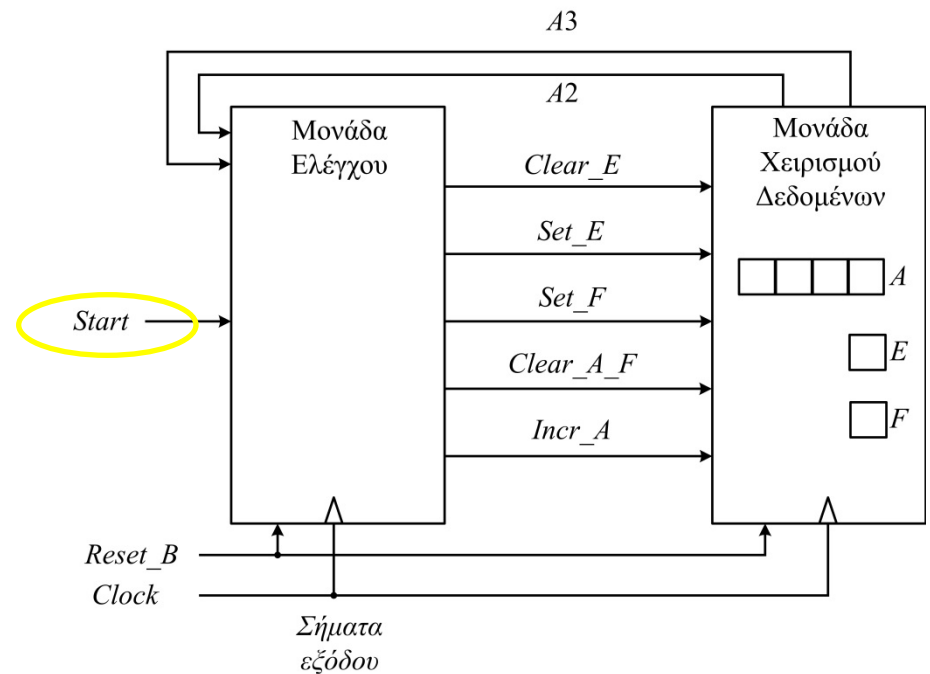


## ► Start

- Αν  $start = 1$  ξεκινά η λειτουργία και γίνονται  $A = 0$  και  $F = 0$
- Σε κάθε επόμενο παλμό  $A = A + 1$  (counter)

## ► Συνθήκες ελέγχου

- Αν  $A2 = 0$  τότε  $E = 0$
- Αν  $A2 = 1$  τότε  $E = 1$ , αν  $A3 = 0$  η μέτρηση συνεχίζεται
- Αν  $A3 = 1$  τότε  $F = 1$  (στον επόμενο παλμό) και σταματά η μέτρηση
- Αν  $start = 0$  το σύστημα κάνει hold, αλλιώς αν  $start = 1$  επαναλαμβάνει τη λειτουργία

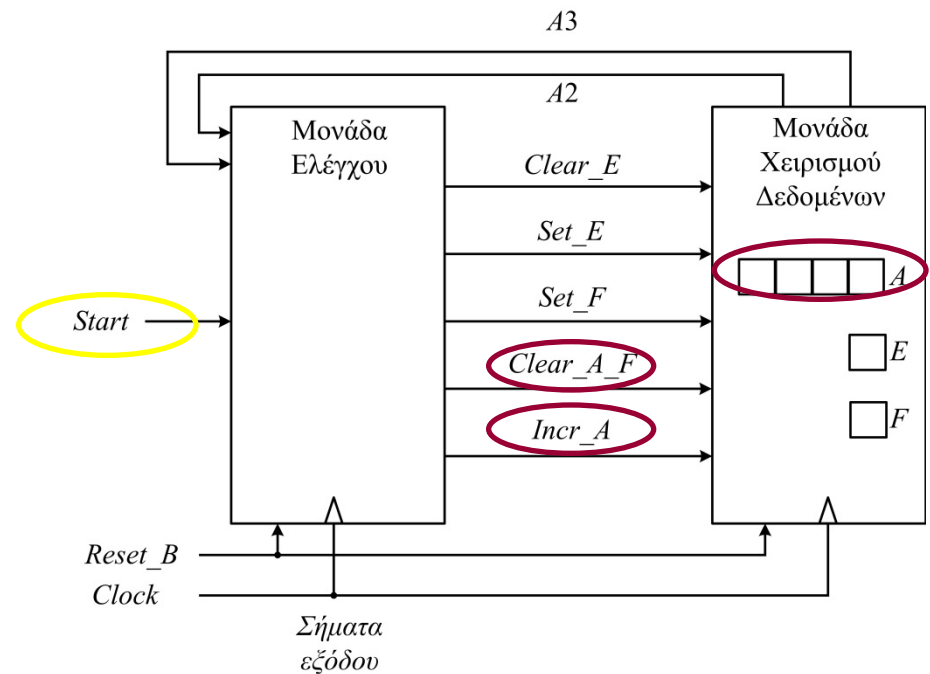


## ► Start

- Αν  $start = 1$  ξεκινά η λειτουργία και γίνονται  $A = 0$  και  $F = 0$
- Σε κάθε επόμενο παλμό  $A = A + 1$  (counter)

## ► Συνθήκες ελέγχου

- Αν  $A2 = 0$  τότε  $E = 0$
- Αν  $A2 = 1$  τότε  $E = 1$ , αν  $A3 = 0$  η μέτρηση συνεχίζεται
- Αν  $A3 = 1$  τότε  $F = 1$  (στον επόμενο παλμό) και σταματά η μέτρηση
- Αν  $start = 0$  το σύστημα κάνει hold, αλλιώς αν  $start = 1$  επαναλαμβάνει τη λειτουργία

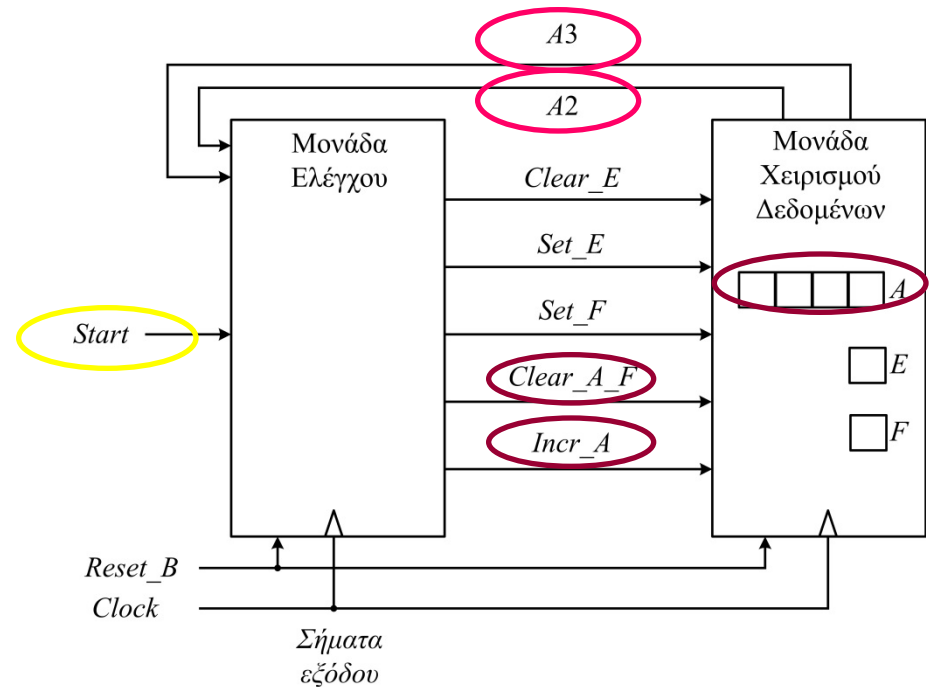


## ► Start

- Αν  $start = 1$  ξεκινά η λειτουργία και γίνονται  $A = 0$  και  $F = 0$
- Σε κάθε επόμενο παλμό  $A = A + 1$  (counter)

## ► Συνθήκες ελέγχου

- Αν  $A2 = 0$  τότε  $E = 0$
- Αν  $A2 = 1$  τότε  $E = 1$ , αν  $A3 = 0$  η μέτρηση συνεχίζεται
- Αν  $A3 = 1$  τότε  $F = 1$  (στον επόμενο παλμό) και σταματά η μέτρηση
- Αν  $start = 0$  το σύστημα κάνει hold, αλλιώς αν  $start = 1$  επαναλαμβάνει τη λειτουργία

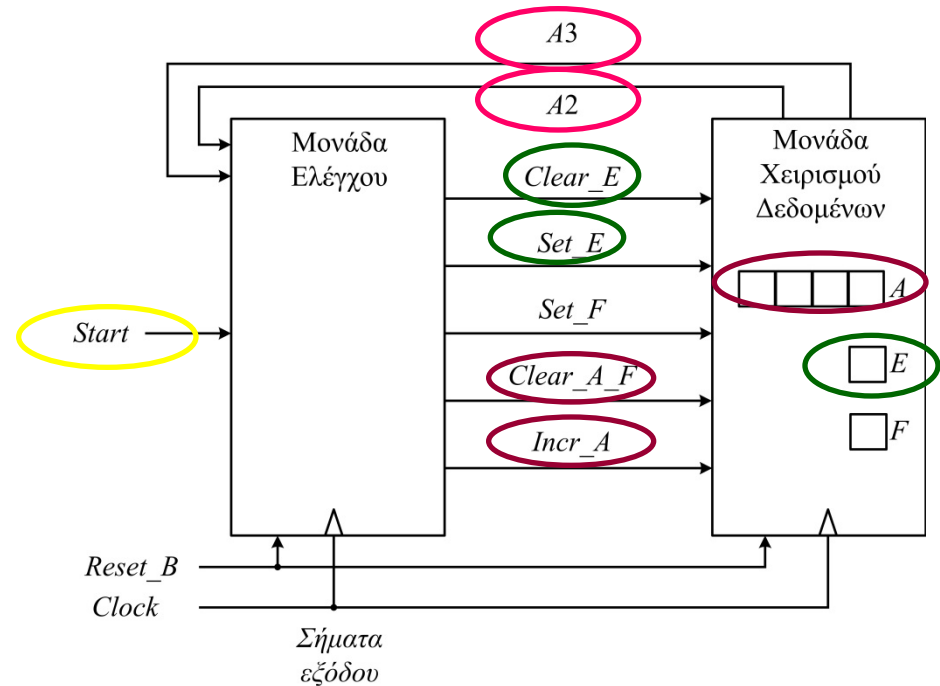


## ► Start

- Αν  $start = 1$  ξεκινά η λειτουργία και γίνονται  $A = 0$  και  $F = 0$
- Σε κάθε επόμενο παλμό  $A = A + 1$  (counter)

## ► Συνθήκες ελέγχου

- Αν  $A2 = 0$  τότε  $E = 0$
- Αν  $A2 = 1$  τότε  $E = 1$ , αν  $A3 = 0$  η μέτρηση συνεχίζεται
- Αν  $A3 = 1$  τότε  $F = 1$  (στον επόμενο παλμό) και σταματά η μέτρηση
- Αν  $start = 0$  το σύστημα κάνει hold, αλλιώς αν  $start = 1$  επαναλαμβάνει τη λειτουργία

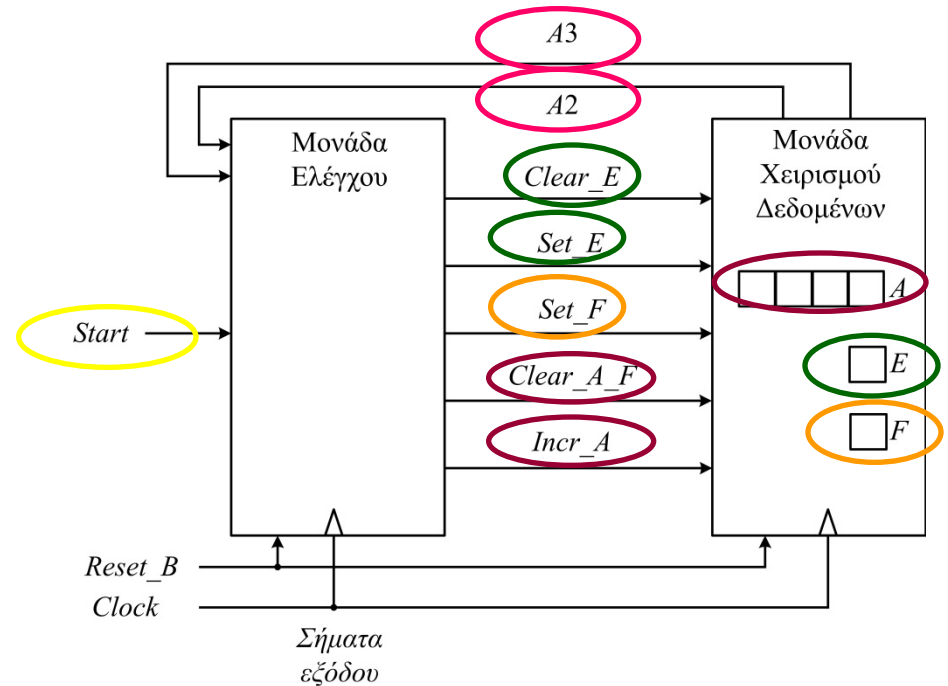


## ► Start

- Αν  $start = 1$  ξεκινά η λειτουργία και γίνονται  $A = 0$  και  $F = 0$
- Σε κάθε επόμενο παλμό  $A = A+1$  (counter)

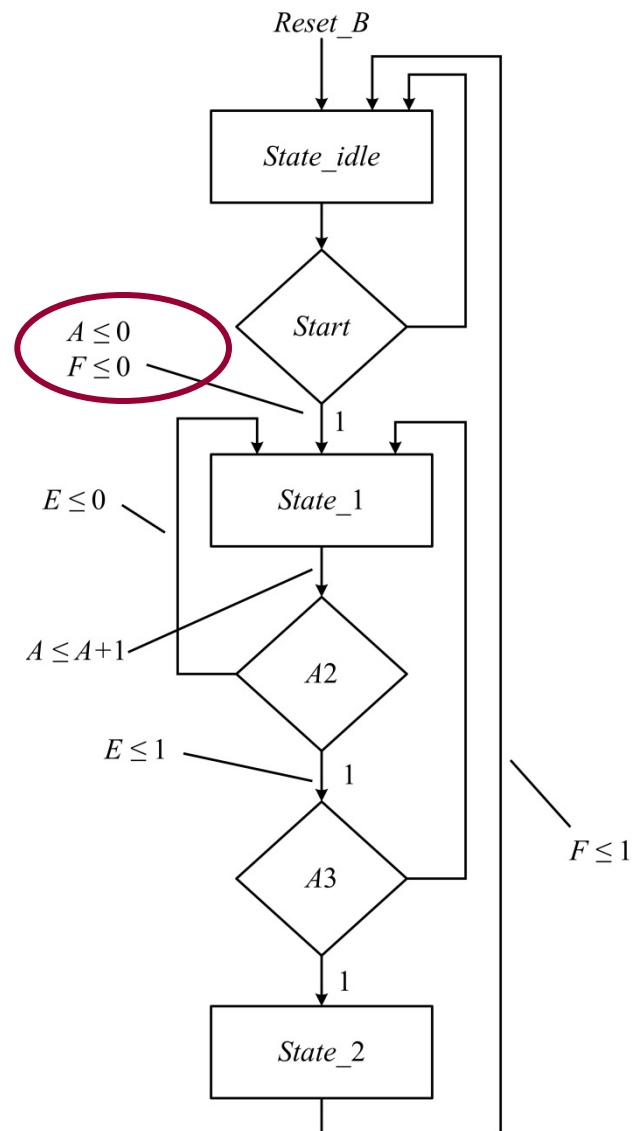
## ► Συνθήκες ελέγχου

- Αν  $A2 = 0$  τότε  $E = 0$
- Αν  $A2 = 1$  τότε  $E = 1$ , αν  $A3 = 0$  η μέτρηση συνεχίζεται
- Αν  $A3 = 1$  τότε  $F = 1$  (στον επόμενο παλμό) και σταματά η μέτρηση
- Αν  $start = 0$  το σύστημα κάνει hold, αλλιώς αν  $start = 1$  επαναλαμβάνει τη λειτουργία



## ► Start

- Αν start=1 τότε  $A=0$  και  $F=0$ , επόμενη κατάσταση η State\_1
- Αν start=0 επομ. κατάσταση η State\_idle

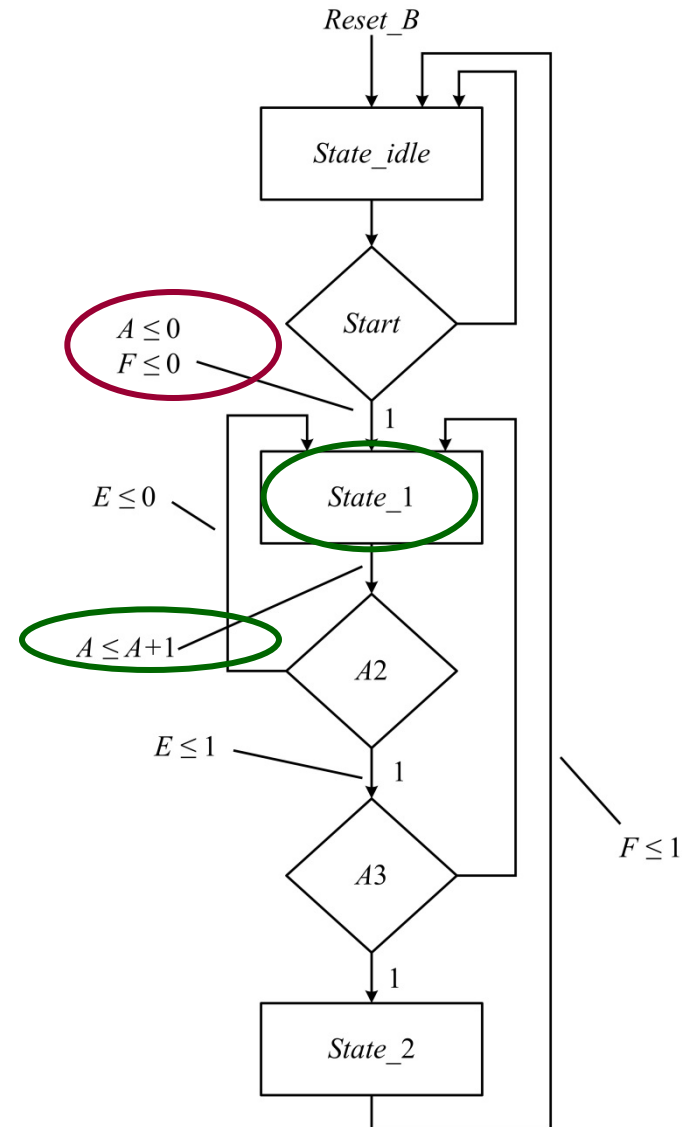


## ➤ Start

- Αν start=1 τότε  $A=0$  και  $F=0$ , επόμενη κατάσταση η State\_1
- Αν start=0 επομ. κατάσταση η State\_idle

## ➤ State\_1

- Σε κάθε επόμενο παλμό  $A=A+1$





## ► Start

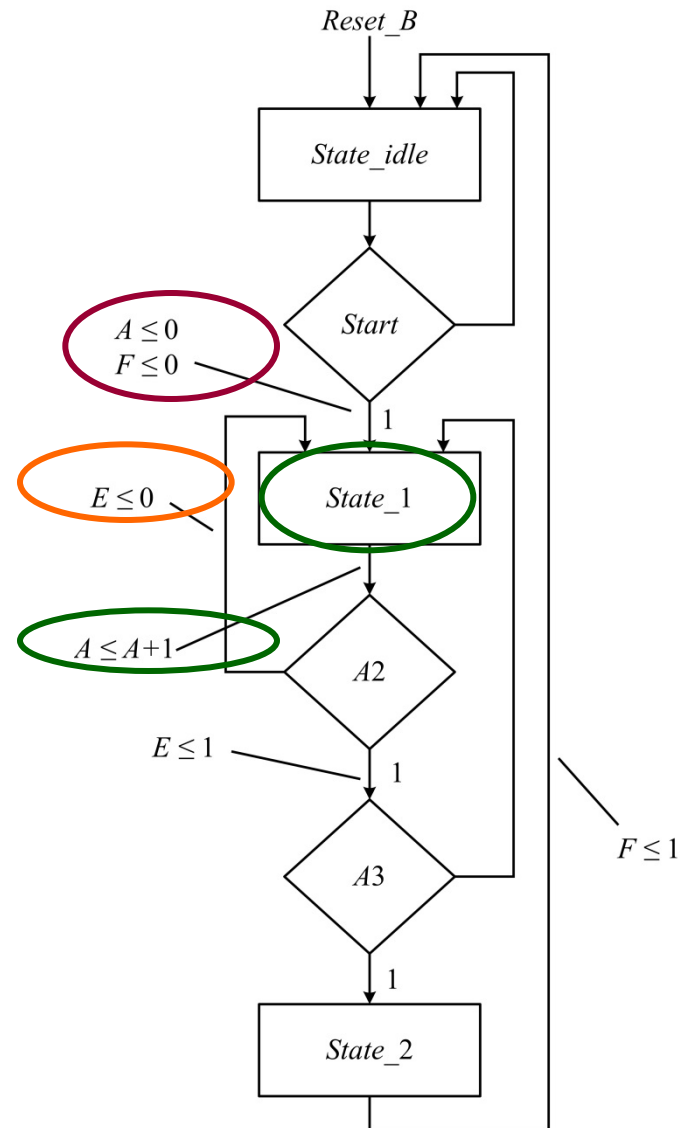
- Αν start=1 τότε  $A=0$  και  $F=0$ , επόμενη κατάσταση η State\_1
- Αν start=0 επομ. κατάσταση η State\_idle

## ► State\_1

- Σε κάθε επόμενο παλμό  $A=A+1$

## ► Συνθήκες ελέγχου

- Αν  $A2=0$  τότε  $E=0$



## ► Start

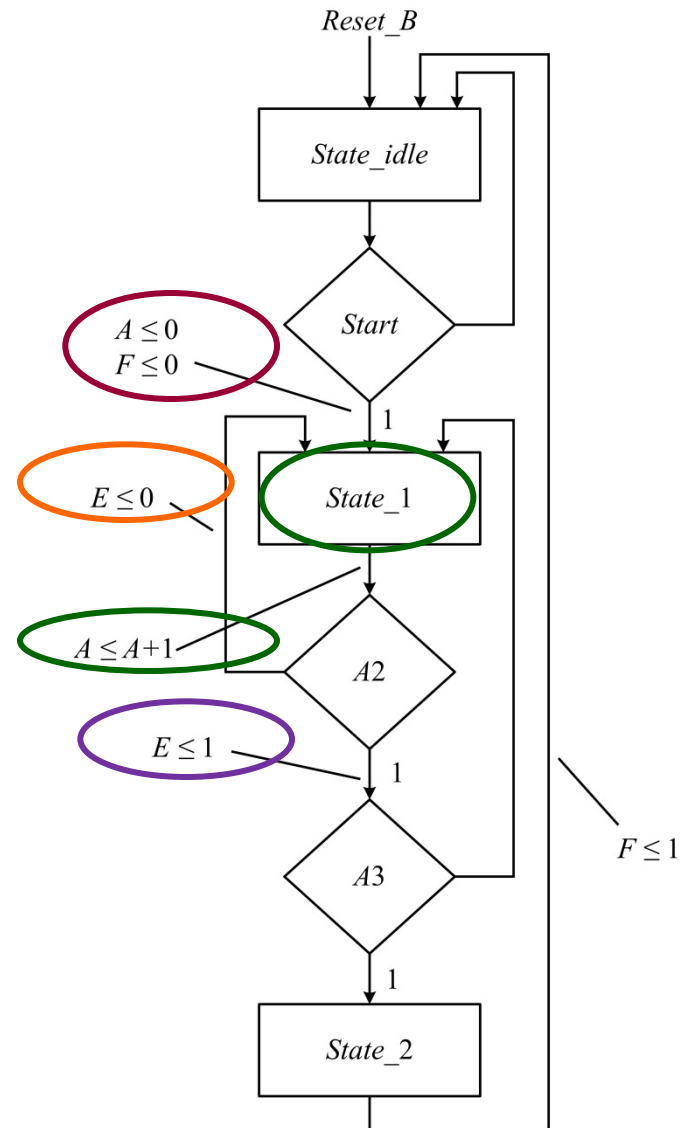
- Αν start=1 τότε  $A=0$  και  $F=0$ , επόμενη κατάσταση η State\_1
- Αν start=0 επομ. κατάσταση η State\_idle

## ► State\_1

- Σε κάθε επόμενο παλμό  $A=A+1$

## ► Συνθήκες ελέγχου

- Αν  $A2=0$  τότε  $E=0$
- Αν  $A2=1$  τότε  $E=1$  και  $A3=0$  η μέτρηση συνεχίζεται



## ► Start

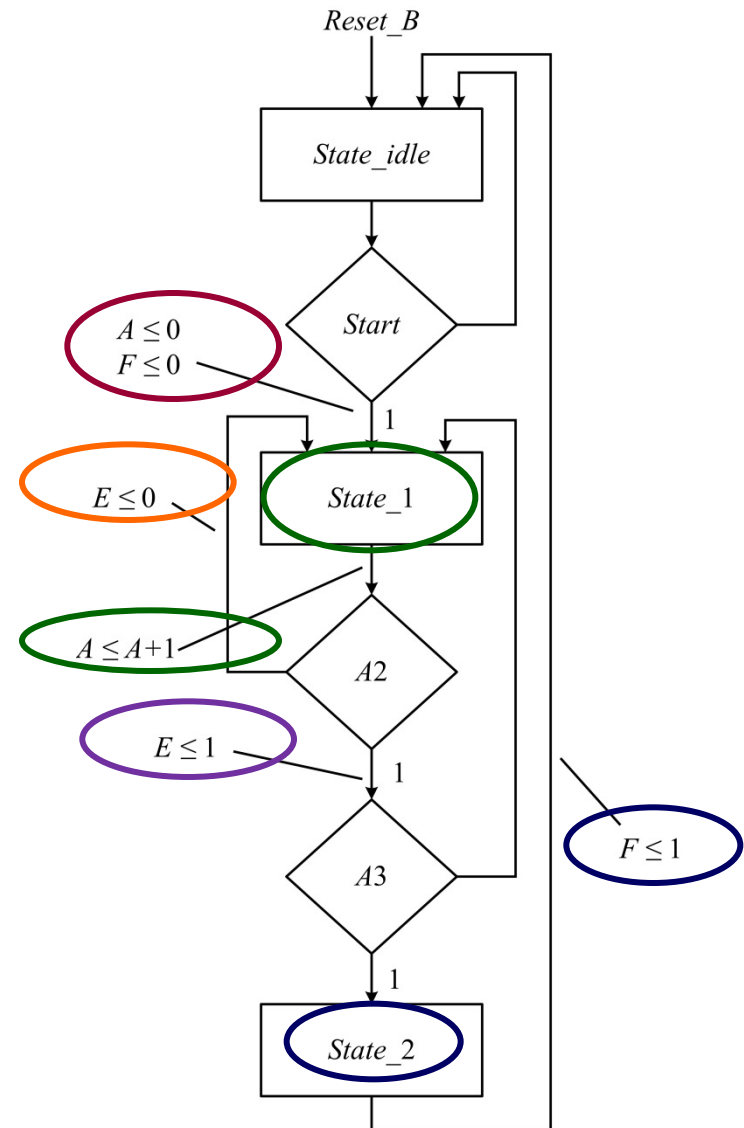
- Αν start=1 τότε  $A=0$  και  $F=0$ , επόμενη κατάσταση η State\_1
- Αν start=0 επομ. κατάσταση η State\_idle

## ► State\_1

- Σε κάθε επόμενο παλμό  $A=A+1$

## ► Συνθήκες ελέγχου

- Αν  $A2=0$  τότε  $E=0$
- Αν  $A2=1$  τότε  $E=1$  και  $A3=0$  η μέτρηση συνεχίζεται
- Αν  $A3=1$  τότε  $F=1$  στον επόμενο παλμό (άρα μια επιπλέον κατάσταση – State\_2) και σταματά η μέτρηση



## ➤ Start

- Αν start=1 τότε  $A=0$  και  $F=0$ , επόμενη κατάσταση η State\_1
- Αν start=0 επομ. κατάσταση η State\_idle

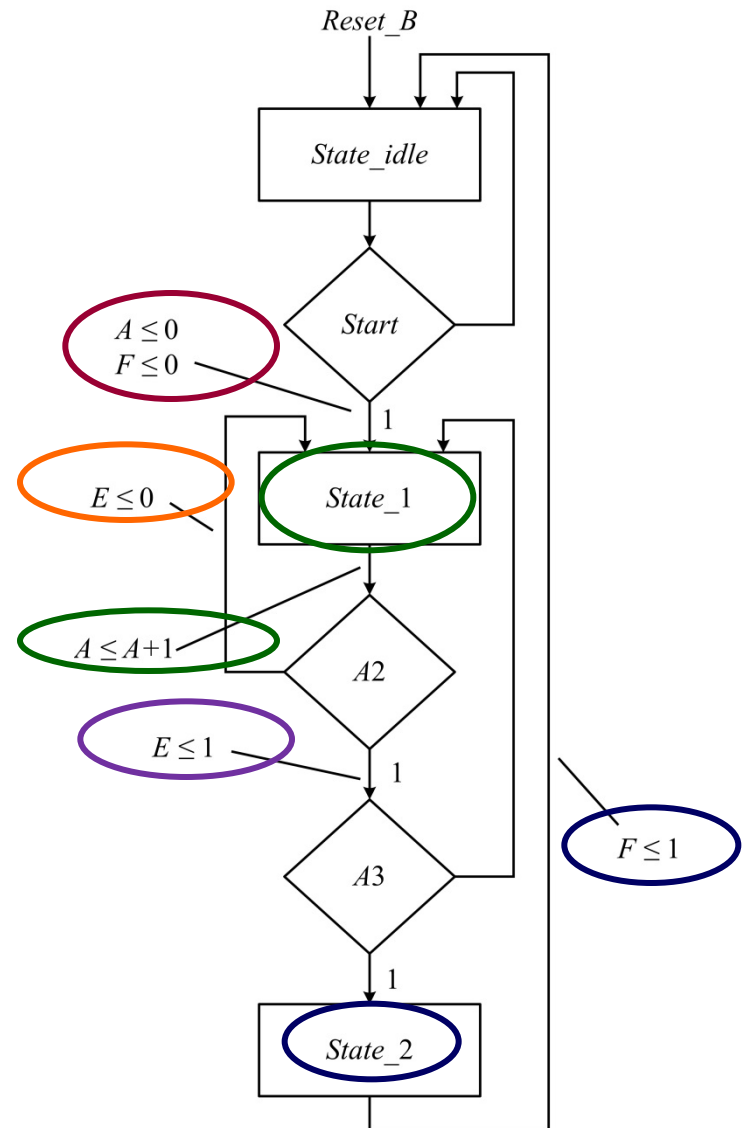
## ➤ State\_1

- Σε κάθε επόμενο παλμό  $A=A+1$

## ➤ Συνθήκες ελέγχου

- Αν  $A2=0$  τότε  $E=0$
- Αν  $A2=1$  τότε  $E=1$  και  $A3=0$  η μέτρηση συνεχίζεται
- Αν  $A3=1$  τότε  $F=1$  στον επόμενο παλμό (άρα μια επιπλέον κατάσταση – State\_2) και σταματά η μέτρηση

- Αν start=0 το σύστημα κάνει hold, αν start=1 επαναλαμβάνει τη λειτουργία

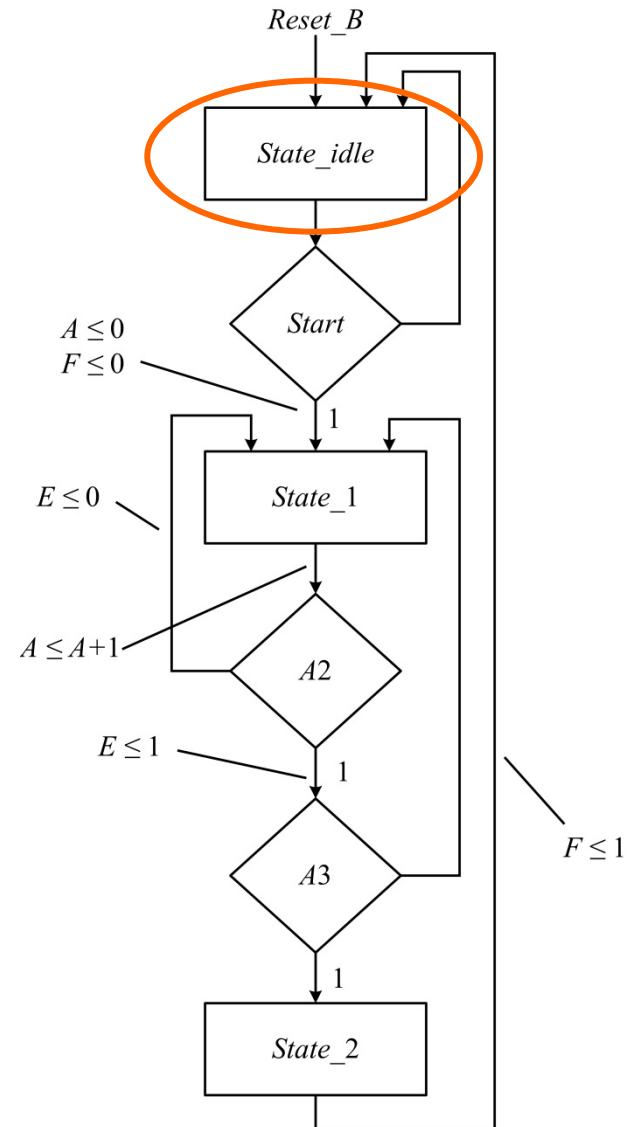


## ► Start

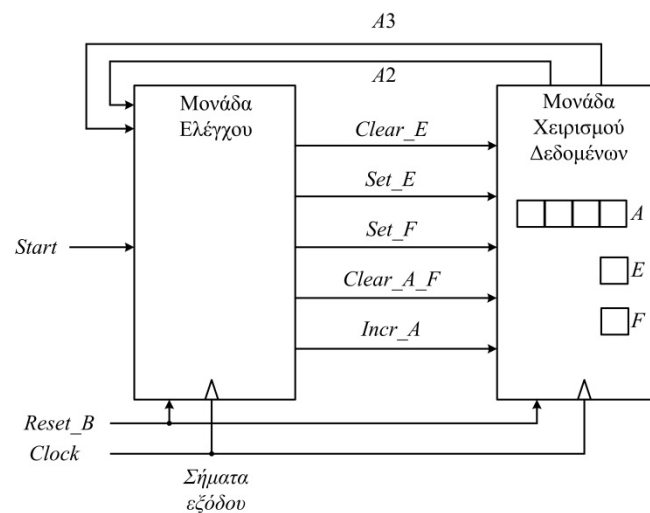
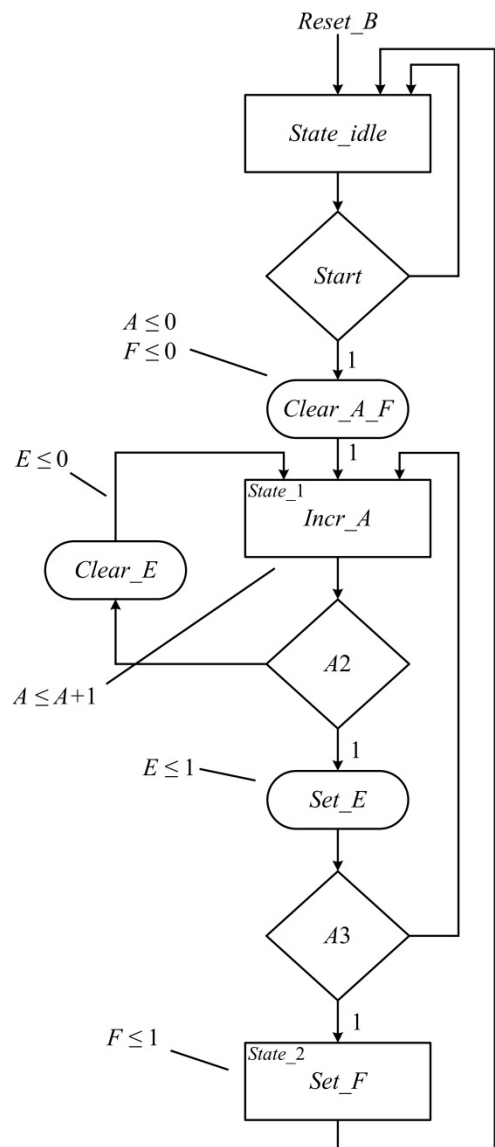
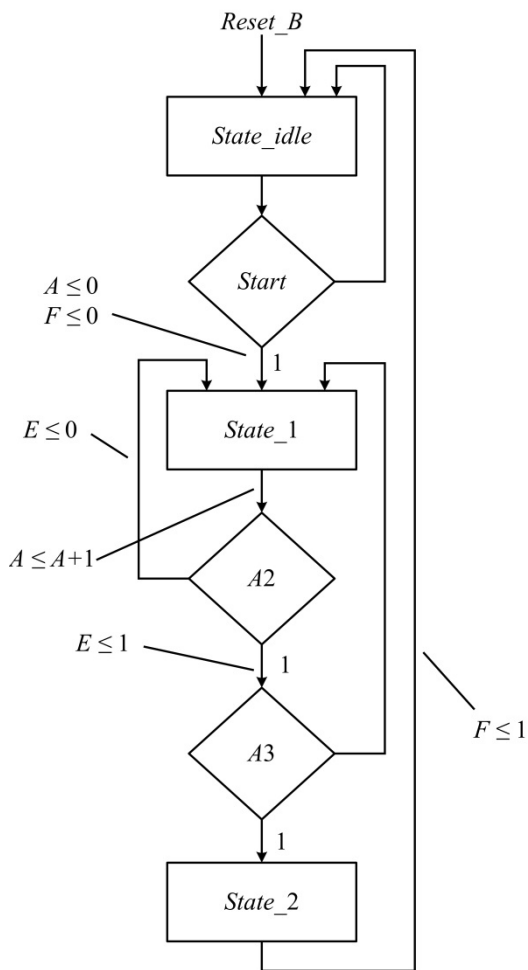
- Αν  $start=1$  ξεκινά η λειτουργία και  $A=0$  και  $F=0$
- Σε κάθε επόμενο παλμό  $A=A+1$

## ► Συνθήκες ελέγχου

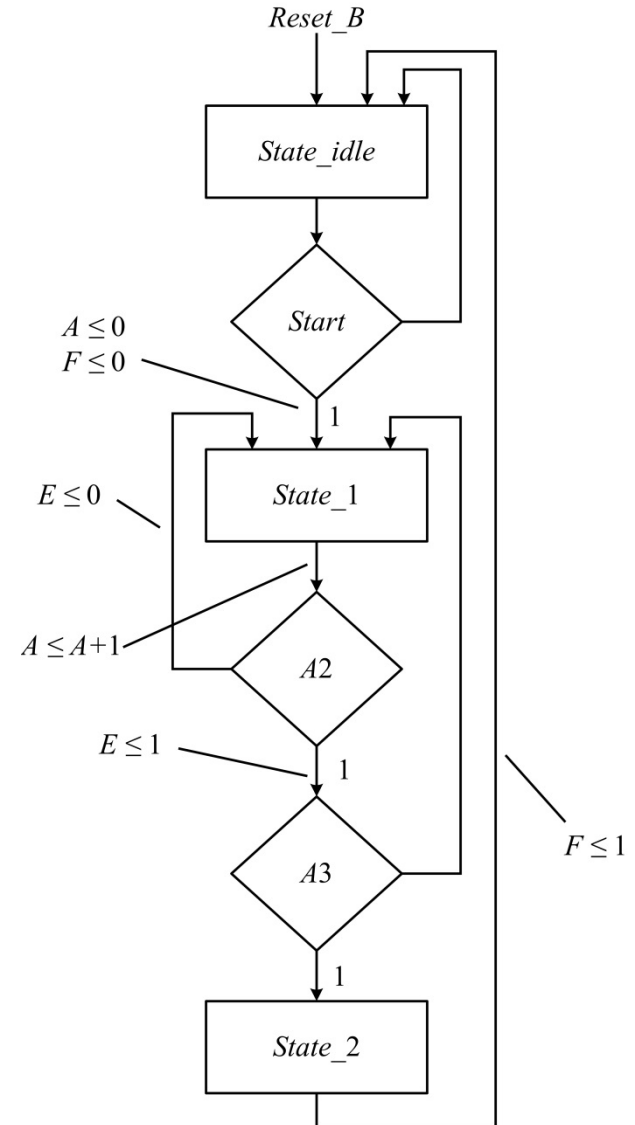
- Αν  $A2=0$  τότε  $E=0$
- Αν  $A2=1$  τότε  $E=1$ , αν  $A3=0$  η μέτρηση συνεχίζεται
- Αν  $A3=1$  τότε  $F=1$  (στον επόμενο παλμό) και σταματά η μέτρηση
- Αν  $start=0$  το σύστημα κάνει hold, αλλιώς αν  $start=1$  επαναλαμβάνει τη λειτουργία



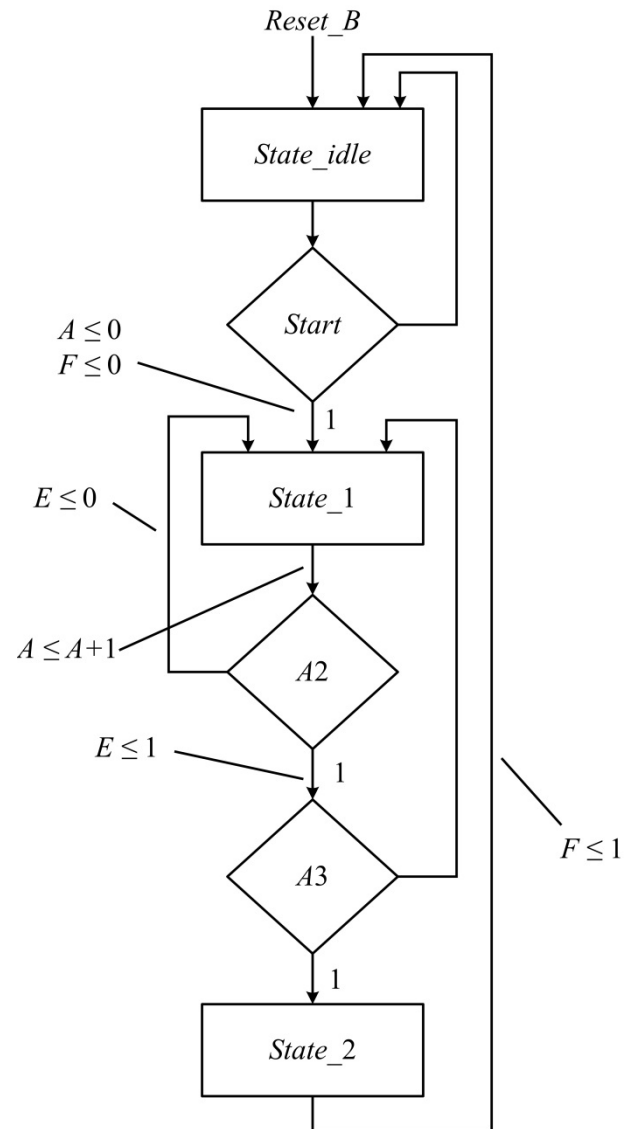
# Διάγραμμα ASMD με τα Σήματα Ελέγχου



- Κάθε μπλοκ ASMD καθορίζει τα σήματα που ελέγχουν τις λειτουργίες που πρέπει να γίνουν στην ενεργή ακμή του ρολογιού
- Τα σήματα ελέγχου στα κουτιά κατάστασης και ενεργειών υπό συνθήκη παράγονται όταν το κύκλωμα ελέγχου βρίσκεται στην αντίστοιχη κατάσταση
- Όταν έρχεται η ακμή του ρολογιού εκτελούνται οι λειτουργίες που αντιστοιχούν στο αντίστοιχο βέλος του ASMD



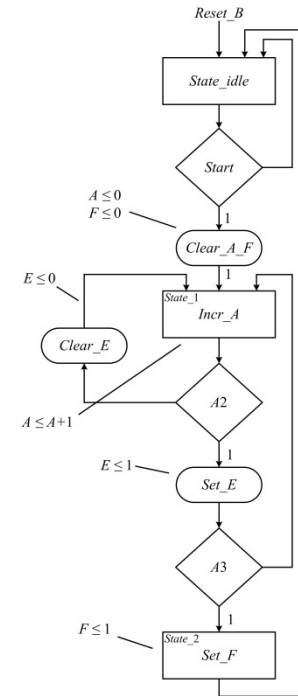
State	Conditions	Counter				Flip-Flops	
		$A_3$	$A_2$	$A_1$	$A_0$	$E$	$F$
State_1	$A_2=0, A_3=0$	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	$A_2=1, A_3=0$	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	$A_2=0, A_3=1$	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	$A_2=1, A_3=1$	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



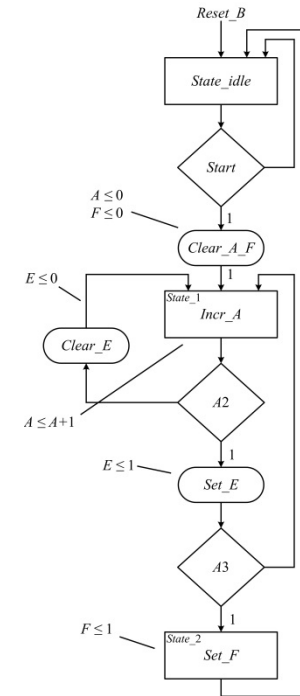


# Χρονισμός (2/6)

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1

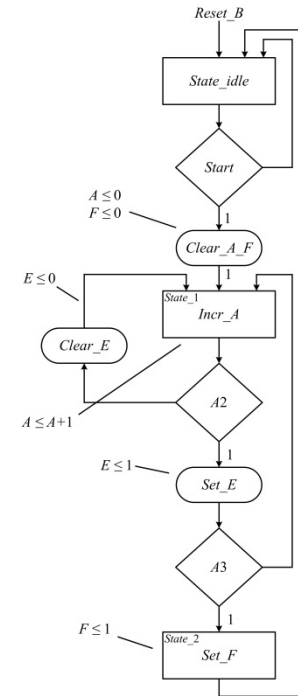


State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



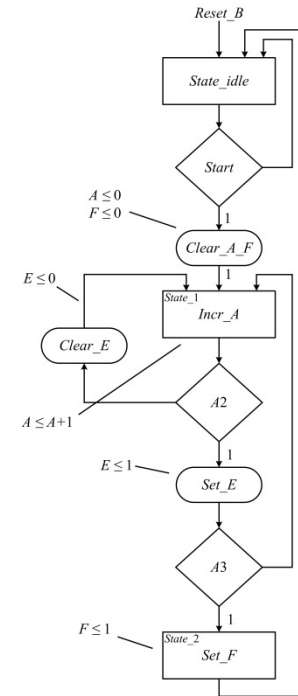
➤ Έστω ότι είμαστε στη State\_1 μετά το μηδενισμό του μετρητή A και του F από το start

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



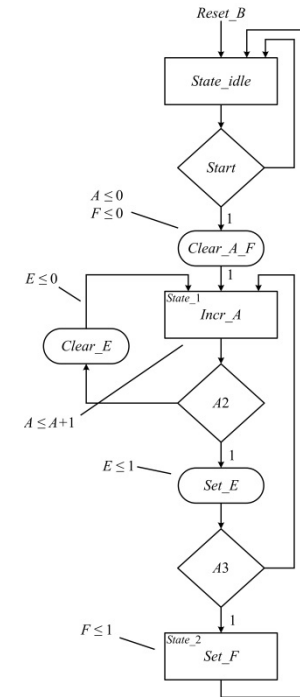
- Έστω ότι είμαστε στη State\_1 μετά το μηδενισμό του μετρητή A και του F από το start
- Υποθέτουμε ότι η μηχανή εκτελούσε κανονικό κύκλο (χωρίς ενεργ. του Rest\_B)
  - Άρα E=1. Το E=1 όταν η μηχανή μεταβαίνει στην State\_2 όταν A2=1

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



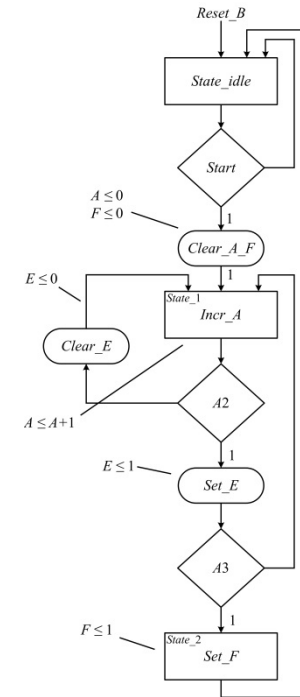
- Έστω ότι είμαστε στη State\_1 μετά το μηδενισμό του μετρητή A και του F από το start
- Υποθέτουμε ότι η μηχανή εκτελούσε κανονικό κύκλο (χωρίς ενεργ. του Rest\_B)
  - Άρα E=1. Το E=1 όταν η μηχανή μεταβαίνει στην State\_2 όταν A2=1
- Η μηχανή (σύστημα) παραμένει στην κατάσταση State\_1 για 13 κύκλους

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



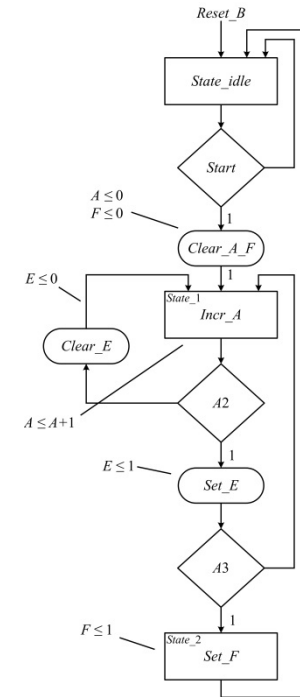
➤ Προσέξτε τη χρονική διαφορά όταν το A<sub>2</sub>=1 και τότε το E=1

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



➤ Προσέξτε τη χρονική διαφορά όταν το A<sub>2</sub>=1 και τότε το E=1

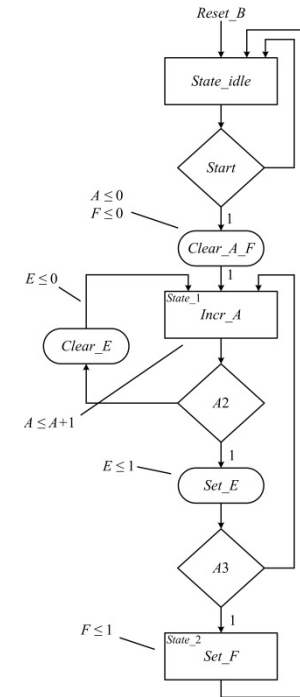
State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



➤ Προσέξτε τη χρονική διαφορά όταν το A<sub>2</sub>=1 και τότε το E=1

-Όταν A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0011 και έρθει η ακμή του ρολογιού τότε η νέα τιμή είναι A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0100

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1

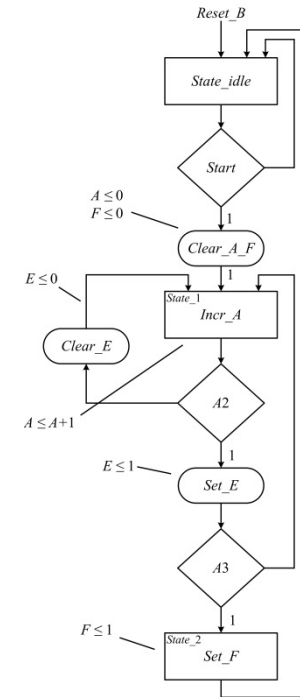


➤ Προσέξτε τη χρονική διαφορά όταν το A<sub>2</sub>=1 και τότε το E=1

- Όταν A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0011 και έρθει η ακμή του ρολογιού τότε η νέα τιμή είναι A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0100
- Όμως η τιμή του A<sub>2</sub> όταν ήρθε η ακμή του ρολογιού ήταν A<sub>2</sub>=0
- Τα F/Fs διαβάζουν την τιμή A<sub>2</sub>=0 και επομένως E=0



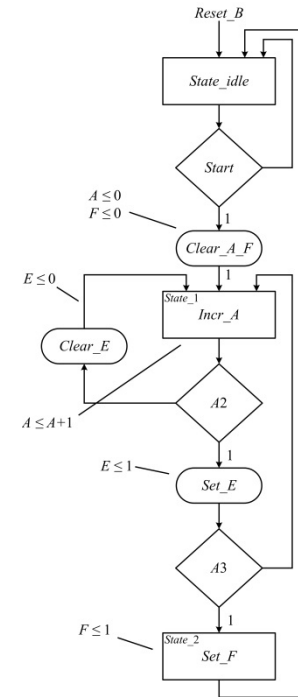
State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
State_1	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
State_1	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
State_2	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



➤ Προσέξτε τη χρονική διαφορά όταν το A<sub>2</sub>=1 και τότε το E=1

- Όταν A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0011 και έρθει η ακμή του ρολογιού τότε η νέα τιμή είναι A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0100
- Όμως η τιμή του A<sub>2</sub> όταν ήρθε η ακμή του ρολογιού ήταν A<sub>2</sub>=0
- Τα F/Fs διαβάζουν την τιμή A<sub>2</sub>=0 και επομένως E=0
- Έτσι, το E=1 γίνεται στην επόμενη ακμή (κύκλο) του ρολογιού

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
State_1	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
State_1	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
State_2	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1

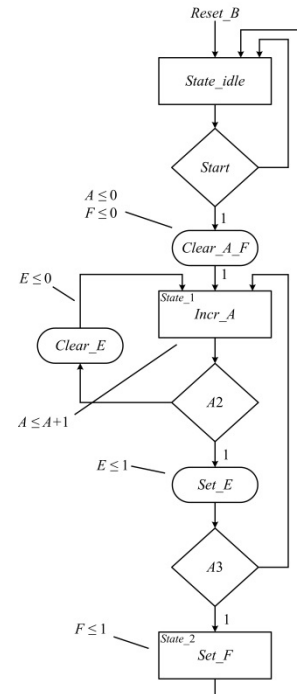


➤ Προσέξτε τη χρονική διαφορά όταν το A<sub>2</sub>=1 και τότε το E=1

- Όταν A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0011 και έρθει η ακμή του ρολογιού τότε η νέα τιμή είναι A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> = 0100
- Όμως η τιμή του A<sub>2</sub> όταν ήρθε η ακμή του ρολογιού ήταν A<sub>2</sub>=0
- Τα F/Fs διαβάζουν την τιμή A<sub>2</sub>=0 και επομένως E=0
- Έτσι, το E=1 γίνεται στην επόμενη ακμή (κύκλο) του ρολογιού

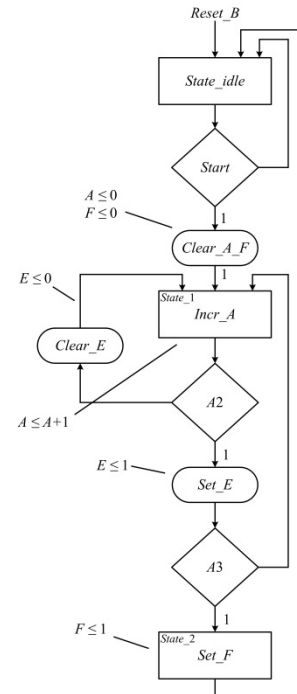
➤ Ομοίως το E=0 όταν η (1000) → (1001) και όχι όταν (0111) → (1000)

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



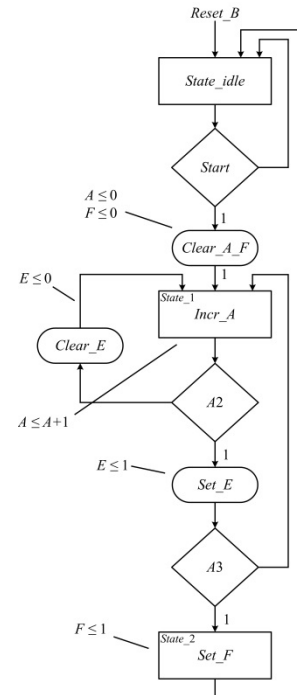
➤ Όταν η μέτρηση φτάσει στο (1100) τότε A<sub>3</sub>=A<sub>2</sub>=1 στον επόμενο παλμό γίνεται E=1 και το σύστημα μεταβαίνει στην κατάσταση State\_2

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



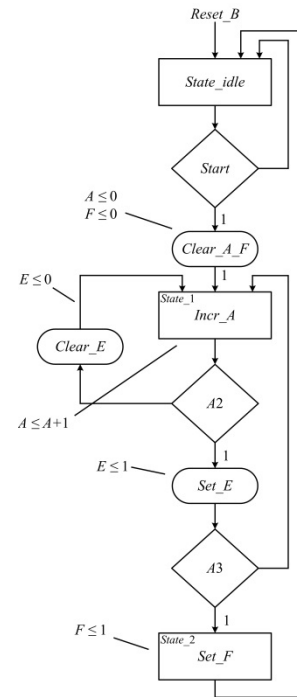
- Όταν η μέτρηση φτάσει στο (1100) τότε A<sub>3</sub>=A<sub>2</sub>=1 στον επόμενο παλμό γίνεται E=1 και το σύστημα μεταβαίνει στην κατάσταση State\_2
- Το σύστημα μένει στην κατάσταση State\_2 για έναν παλμό (περίοδο) ρολογιού

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



- Όταν η μέτρηση φτάσει στο (1100) τότε A<sub>3</sub>=A<sub>2</sub>=1 στον επόμενο παλμό γίνεται E=1 και το σύστημα μεταβαίνει στην κατάσταση State\_2
- Το σύστημα μένει στην κατάσταση State\_2 για έναν παλμό (περίοδο) ρολογιού
- Στον επόμενο παλμό η μηχανή μεταβαίνει από την State\_2 → State\_idle και F=1
- Το σύστημα παραμένει στην State\_idle όσο start = 0

State	Conditions	Counter				Flip-Flops	
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	E	F
State_1	A <sub>2</sub> =0, A <sub>3</sub> =0	0	0	0	0	1	0
		0	0	0	1	0	0
		0	0	1	0	0	0
		0	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =0	0	1	0	0	0	0
		0	1	0	1	1	0
		0	1	1	0	1	0
		0	1	1	1	1	0
	A <sub>2</sub> =0, A <sub>3</sub> =1	1	0	0	0	1	0
		1	0	0	1	0	0
		1	0	1	0	0	0
		1	0	1	1	0	0
	A <sub>2</sub> =1, A <sub>3</sub> =1	1	1	0	0	0	0
State_2		1	1	0	1	1	0
State_idle		1	1	0	1	1	1



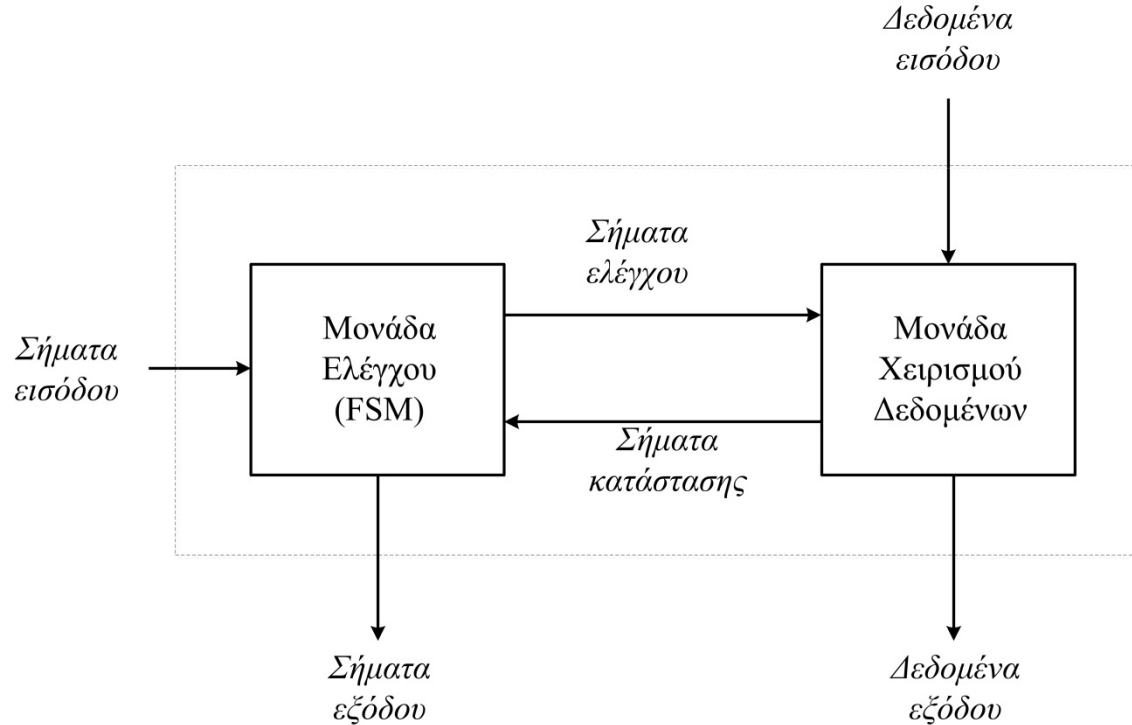
- Συμπερασματικά, βλέπουμε ότι οι λειτουργίες που σχετίζονται με το σήμα E καθυστερούν κατά ένα παλμό ρολογιού
- Σε ένα συμβατικό διάγραμμα δε θα συνέβαινε αυτό

## Χρονισμός (6/6)

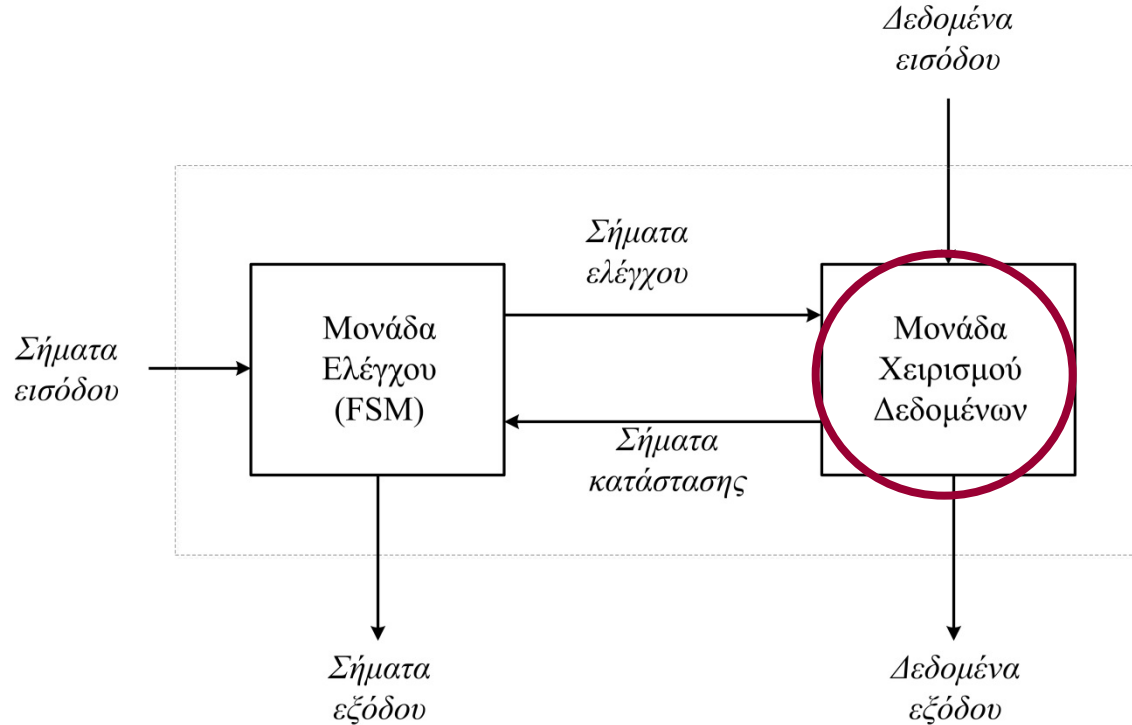
- Η τιμή του A2 που εξετάζει το κουτί απόφασης αντιστοιχεί στην τιμή του μετρητή όταν η παρούσα κατάσταση είναι η State\_1 και πριν ο μετρητής αυξηθεί
  - Το κουτί απόφασης που αφορά το E ανήκει στο μπλοκ της State\_1
- Το κύκλωμα παράγει τα σήματα ελέγχου για όλες τις πράξεις που αφορούν το υπό εκτέλεση μπλοκ **πριν την άφιξη του παλμού**
  - Η επόμενη ακμή ρολογιού θα πυροδοτήσει τις πράξεις των συνδυαστικών κυκλ. και F/Fs συμπεριλαμβανομένου αυτών που αφορούν την επόμενη κατάσταση
- Τα σήματα ελέγχου που ελέγχουν τις πράξεις της μονάδας χειρισμού δεδομένων παράγονται από το κύκλωμα ελέγχου έναν κύκλο πριν την εκτέλεση της πράξης

- Εισαγωγή
- Βασικές έννοιες και ορολογία
- Αλγοριθμικές μηχανές καταστάσεων (Algorithmic State Machines – ASM )
- Παραδείγματα σχεδιασμών
- **Λογικό κύκλωμα ελέγχου**
- Σχεδιασμός με πολυπλέκτες
- Σχεδιασμός χωρίς κυνηγητά
- Σχεδιασμός χωρίς μανδαλωτές

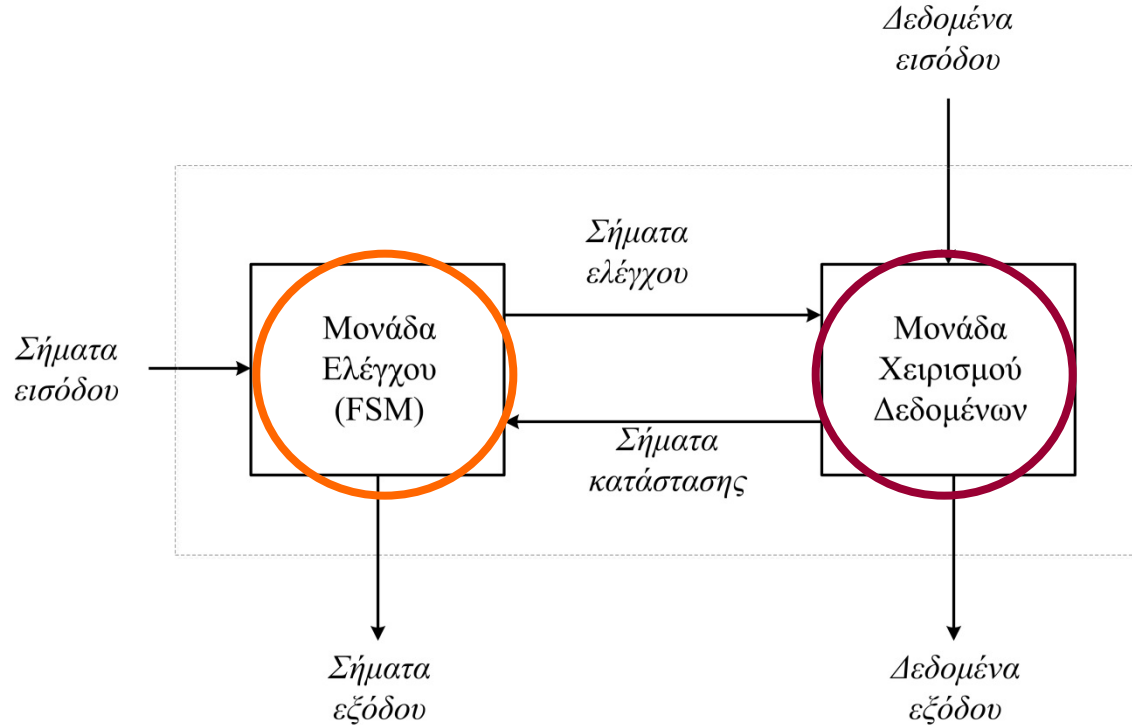




- Στη γενική μορφή κάθε ψηφιακό κύκλωμα /σύστημα αποτελείται από

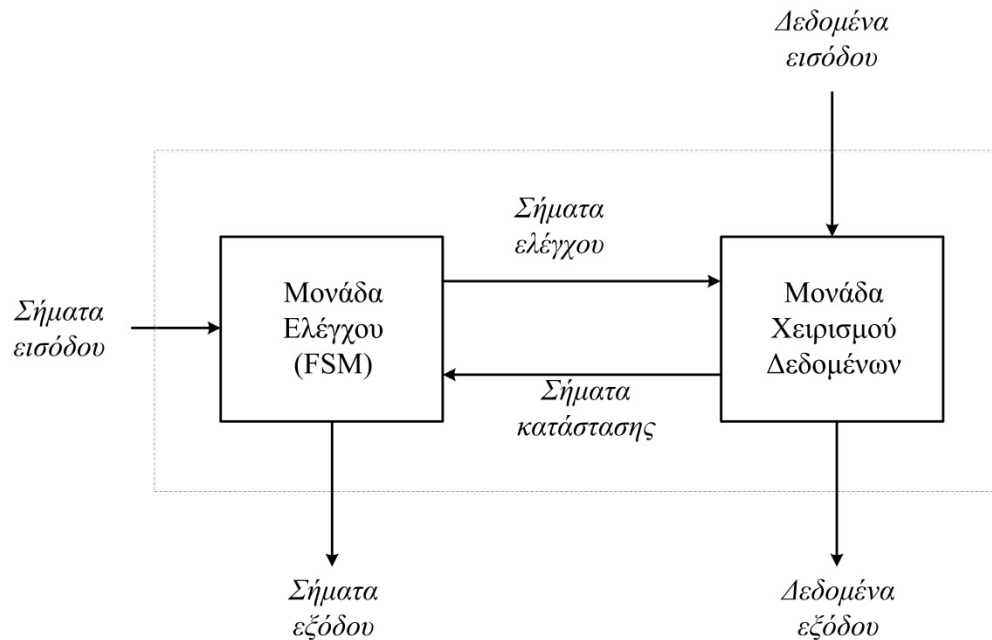


- Στη γενική μορφή κάθε ψηφιακό κύκλωμα /σύστημα αποτελείται από
  - Κύκλωμα χειρισμού δεδομένων



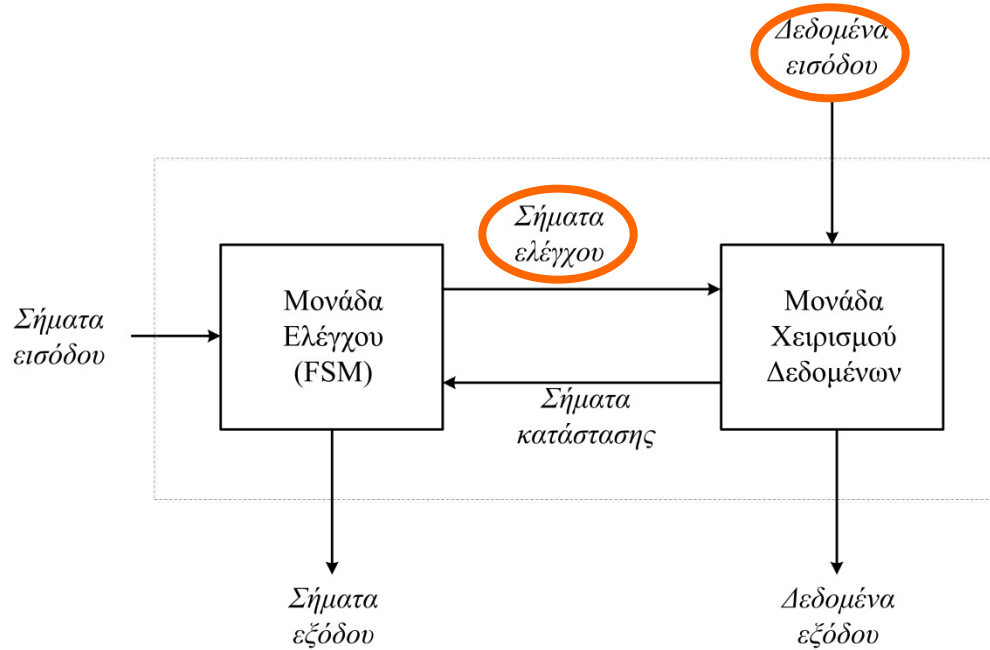
- Στη γενική μορφή κάθε ψηφιακό κύκλωμα /σύστημα αποτελείται από
  - Κύκλωμα χειρισμού δεδομένων
  - Κύκλωμα ελέγχου

# Κύκλωμα Χειρισμού Δεδομένων (1/2)



➤ Είσοδοι /έξοδοι

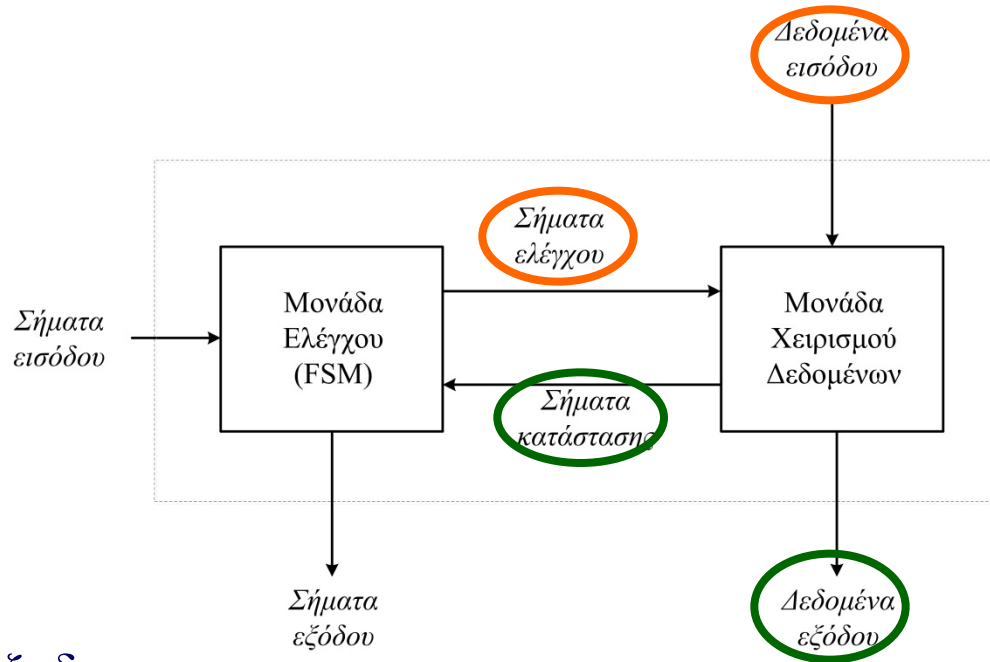
# Κύκλωμα Χειρισμού Δεδομένων (1/2)



## ➤ Είσοδοι /έξοδοι

– Δεδομένα εισόδου (εξωτερικό περιβάλλον), σήματα ελέγχου (κύκλωμα ελέγχου)

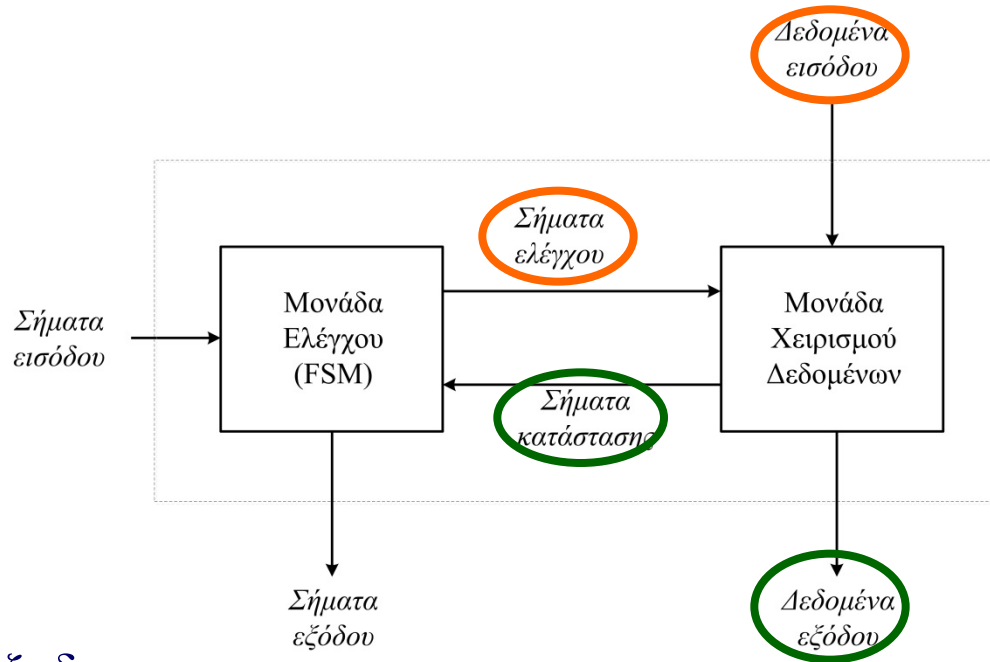
# Κύκλωμα Χειρισμού Δεδομένων (1/2)



## ➤ Είσοδοι /έξοδοι

- Δεδομένα εισόδου (εξωτερικό περιβάλλον), σήματα ελέγχου (κύκλωμα ελέγχου)
- Δεδομένα εξόδου, σήματα κατάστασης

# Κύκλωμα Χειρισμού Δεδομένων (1/2)



## ➤ Είσοδοι /έξοδοι

- Δεδομένα εισόδου (εξωτερικό περιβάλλον), σήματα ελέγχου (κύκλωμα ελέγχου)
- Δεδομένα εξόδου, σήματα κατάστασης

## ➤ Κύκλωμα χειρισμού δεδομένων περιλαμβάνει **συνδυαστικά κυκλώματα** για

- Την υλοποίηση των πράξεων (αριθμητικές, λογικές, ...) σε δεδομένα καταχωρητών
- Την παραγωγή των σημάτων κατάστασης (χρησιμ. από το κύκλωμα ελέγχου)

➤ Οι προδιαγραφές για τα κυκλώματα χειρισμού δεδομένων προσδιορίζονται από:

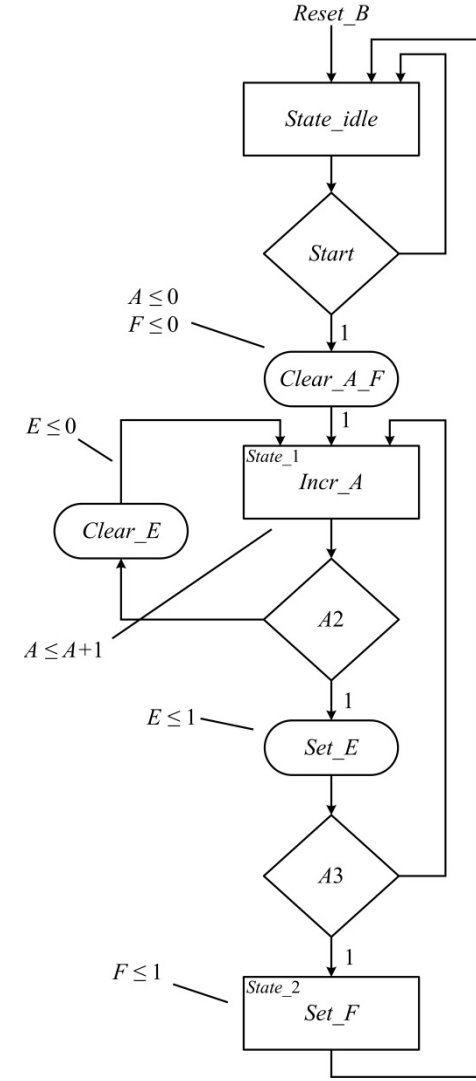
➤ Σήματα ελέγχου για το datapath που έχουν γραφεί εντός

- των κουτιών κατάστασης (π.χ. `incr_A`)
- των κουτιών υπό συνθήκη ενεργειών (π.χ. `clr_E`)

➤ Κυκλώματα datapath

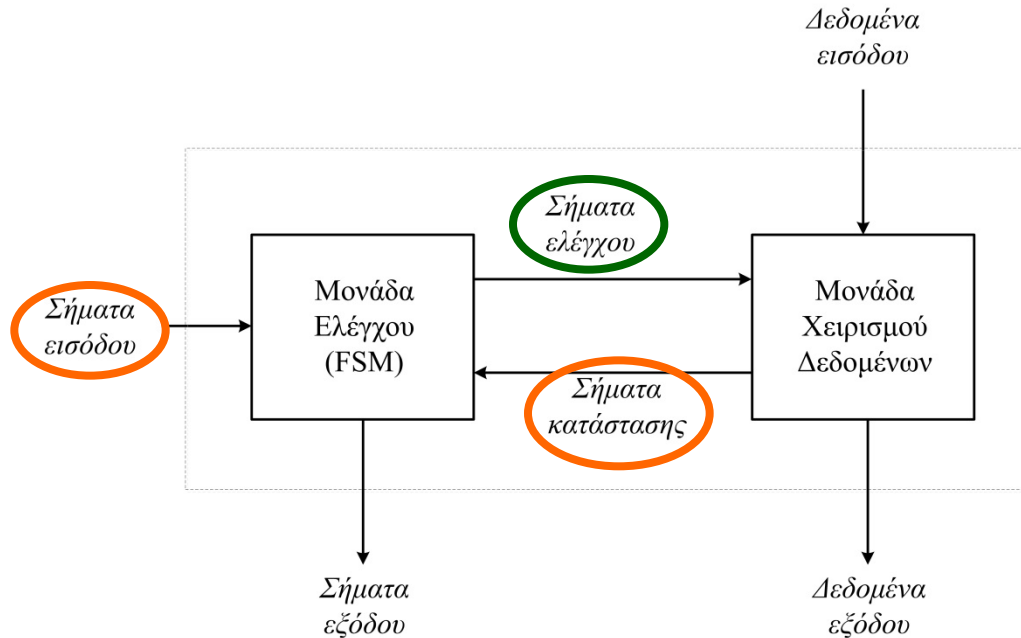
➤ Από την πληροφορία που υπάρχει στα βέλη του διαγράμματος ASMD που σχετίζονται με τις λειτουργίες της μονάδας χειρισμού δεδομένων

- `incr_a => counter`
- `set/clr E, F => F/Fs`





# Κύκλωμα Ελέγχου (1/2)



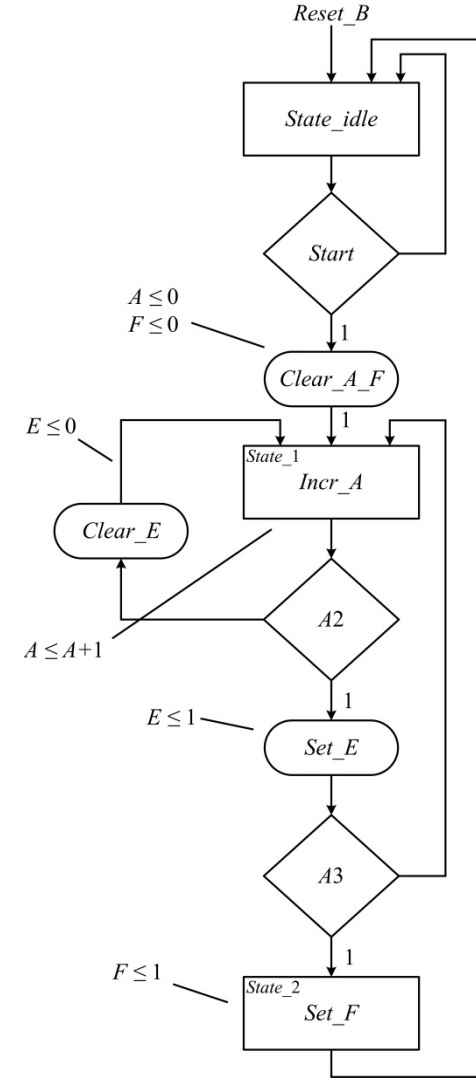
## ➤ Είσοδοι /έξοδοι

- Σήματα ελέγχου (από εξωτερικό περιβάλλον), σήματα κατάστασης (από τη μονάδα χειρισμού δεδομένων)
- Σήματα ελέγχου (προς εξωτερικό περιβάλλον), σήματα ελέγχου (προς μονάδα χειρισμού δεδομένων)

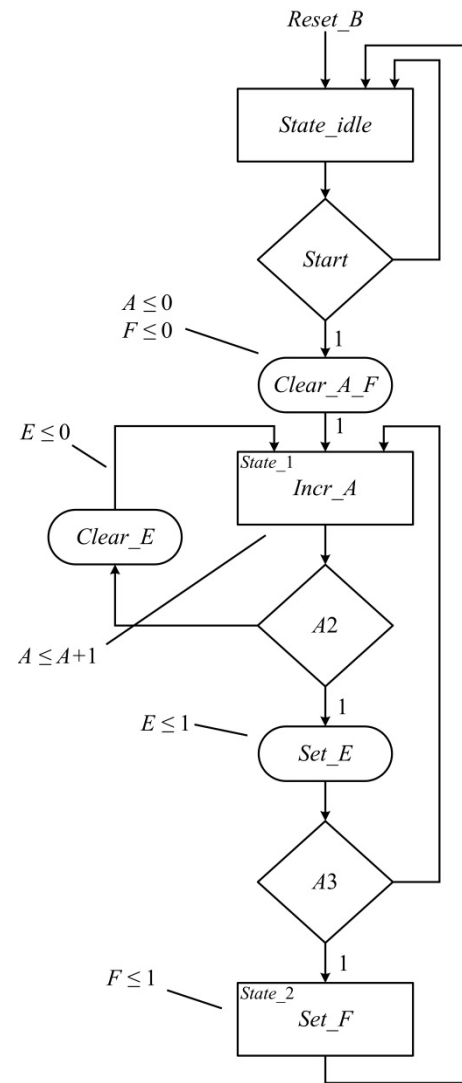
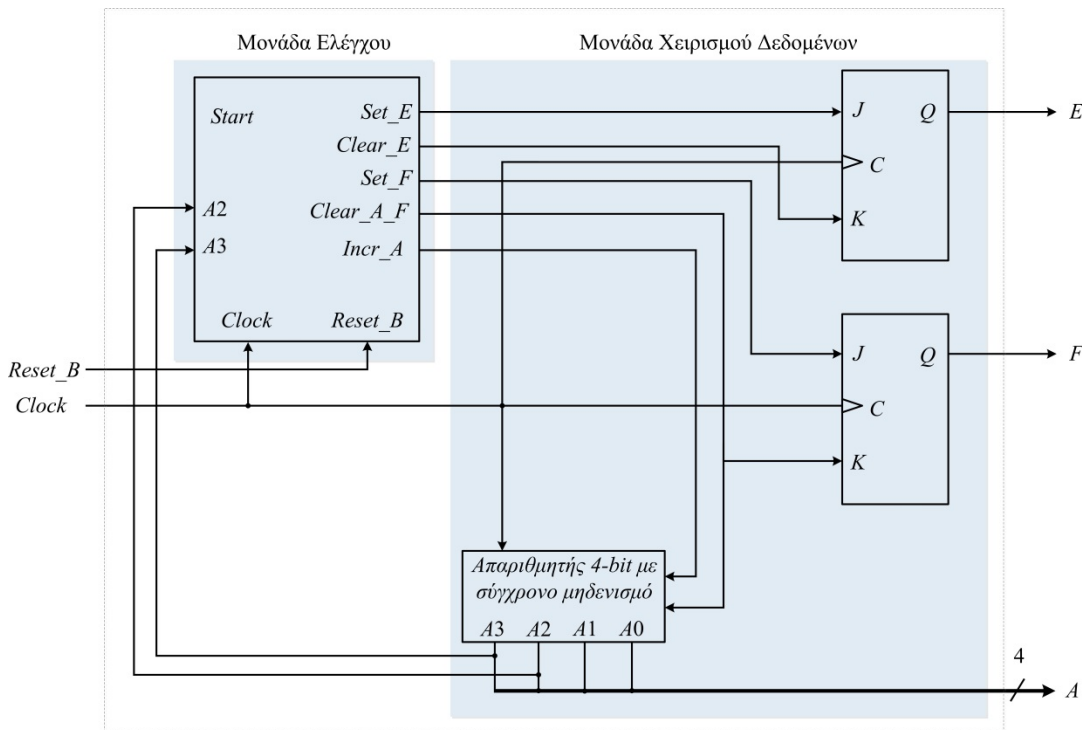
## ➤ Κύκλωμα χειρισμού δεδομένων περιλαμβάνει ακολουθιακά κυκλώματα για

- Την υλοποίηση της μηχανής καταστάσεων (FSM) και την παραγωγή των σημάτων ελέγχου (έξοδοι)

- Οι προδιαγραφές για το κύκλωμα ελέγχου προσδιορίζονται από:
- Τη λειτουργία των κουτιών απόφασης
- Μεταβάσεις των καταστάσεων

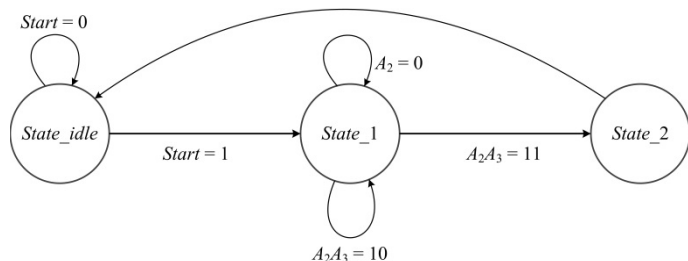


# Κυκλώματα Χειρισμού Δεδομένων & Ελέγχου

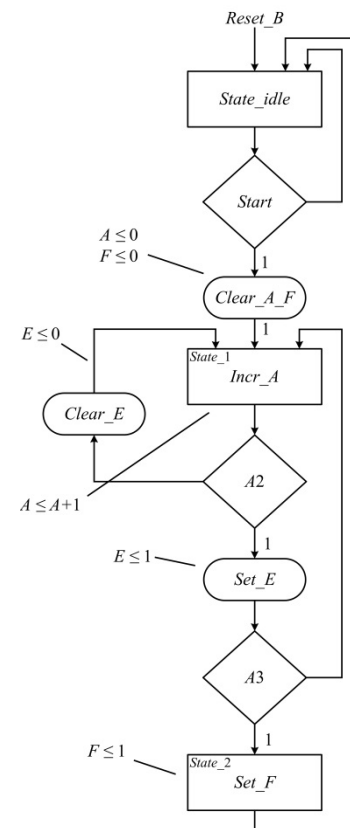


- Ένα RTL σύστημα παριστάνεται αν ορίσουμε:
  - Τους καταχωρητές
  - Τις πράξεις που δίνονται στα δεδομένα των καταχωρητών
  - Τα σήματα ελέγχου και τη χρονική ακολουθία τους
  
- Οι πράξεις & σήματα ελέγχου προκύπτουν από το διάγραμμα ASMD
  
- Είναι βολικό να χωριστούν τα κυκλώματα χειρισμού δεδομένων & ελέγχου
  - Εύκολη και συστηματική σχεδίαση του κυκλώματος ελέγχου
  
- Χρήση διαγραμμάτων καταστάσεων (προκύπτουν από το ASMD)

# Διάγραμμα Καταστάσεων Επίπεδο Καταχωρητή

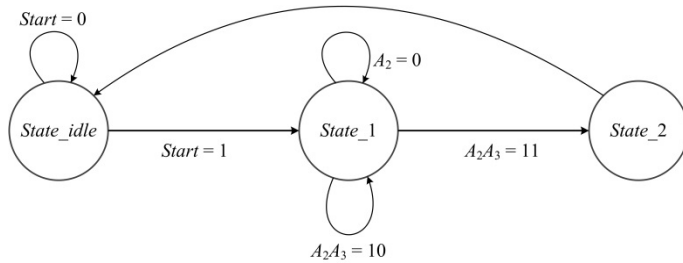


$State\_idle \rightarrow State\_1, Clear\_A\_F$	$A \leftarrow 0, F \leftarrow 0$
$State\_1 \rightarrow State\_1, Incr\_A$	$A \leftarrow A+1$
if ( $A_2=1$ ) then $Set\_E$	$E \leftarrow 1$
if ( $A_2=0$ ) then $Clear\_E$	$E \leftarrow 0$
$State\_2 \rightarrow State\_idle, Set\_F$	$F \leftarrow 1$



- Περιλαμβάνει: καταστάσεις και μεταβάσεις καταστάσεων
- Για κάθε μετάβαση σημειώνεται το σήμα (ή συνδυασμός σημάτων) που την προκαλεί
- Σε κάθε κατάσταση περιγράφονται οι λειτουργίες που απαιτούνται

# Πίνακας Καταστάσεων



$State\_idle \rightarrow State\_1, Clear\_A\_F$	$A \leftarrow 0, F \leftarrow 0$
$State\_1 \rightarrow State\_1, Incr\_A$	$A \leftarrow A+1$
if ( $A_2=1$ ) then $Set\_E$	$E \leftarrow 1$
if ( $A_2=0$ ) then $Clear\_E$	$E \leftarrow 0$
$State\_2 \rightarrow State\_idle, Set\_F$	$F \leftarrow 1$

Present State			Inputs			Next State		Outputs				
Symbol	$G_1$	$G_0$	Start	$A_2$	$A_3$	$G_1$	$G_0$	Set_E	Clear_E	Set_F	Clear_A_F	Incr_A
State_idle	0	0	0	X	X	0	0	0	0	0	0	0
State_idle	0	0	1	X	X	0	1	0	0	0	1	0
State_1	0	1	X	0	X	0	1	0	1	0	0	1
State_1	0	1	X	1	0	0	1	1	0	0	0	1
State_1	0	1	X	1	1	1	1	1	0	0	0	1
State_2	1	1	X	X	X	0	0	0	0	1	0	0

- Το διάγραμμα καταστάσεων μπορεί να μετατραπεί σε πίνακα καταστάσεων
  - Ο πίνακας καταστάσεων χρησιμοποιείται για τη σχεδίαση του ακολουθιακού κυκλώματος ελέγχου
- Ο πίνακας καταστάσεων περιλαμβάνει:
  - Τρέχουσα και επόμενη κατάσταση
  - Τρέχουσες Εισόδους
  - Εξόδους

- Έχοντας το πίνακα μετάβασης καταστάσεων ο σχεδιασμός του ακολουθιακού κυκλώματος γίνεται με γνωστή μεθοδολογία
- Όμως τι περισσότερες φορές υπάρχουν πολλές αδιάφορες καταστάσεις που απλοποιούν την παραπάνω διαδικασία
- Στο συγκεκριμένο παράδειγμα έχουμε
  - 2 σήματα για την παρούσα κατάσταση και
  - 3 σήματα εισόδων
- Άρα ο πίνακας έχει 32 γραμμές
- Χρησιμοποιώντας την προηγούμενη απλοποιημένη μορφή η σχεδίαση μπορεί να γίνει πιο εύκολη

Present State			Inputs			Next State		Outputs				
<i>Symbol</i>	$G_1$	$G_0$	<i>Start</i>	$A_2$	$A_3$	$G_1$	$G_0$	<i>Set_E</i>	<i>Clear_E</i>	<i>Set_F</i>	<i>Clear_A_F</i>	<i>Incr_A</i>
<i>State_idle</i>	0	0	0	X	X	0	0	0	0	0	0	0
<i>State_idle</i>	0	0	1	X	X	0	1	0	0	0	1	0
<i>State_1</i>	0	1	X	0	X	0	1	0	1	0	0	1
<i>State_1</i>	0	1	X	1	0	0	1	1	0	0	0	1
<i>State_1</i>	0	1	X	1	1	1	1	1	0	0	0	1
<i>State_2</i>	1	1	X	X	X	0	0	0	0	1	0	0

Κωδικοποίηση καταστάσεων

$S\_idle = (0, 0)$

$S\_1 = (0, 1)$

$S\_2 = (1, 1)$



Present State			Inputs			Next State		Outputs				
Symbol	$G_1$	$G_0$	Start	$A_2$	$A_3$	$G_1$	$G_0$	Set_E	Clear_E	Set_F	Clear_A_F	Incr_A
State_idle	0	0	0	X	X	0	0	0	0	0	0	0
State_idle	0	0	1	X	X	0	1	0	0	0	1	0
State_1	0	1	X	0	X	0	1	0	1	0	0	1
State_1	0	1	X	1	0	0	1	1	0	0	0	1
State_1	0	1	X	1	1	1	1	1	0	0	0	1
State_2	1	1	X	X	X	0	0	0	0	1	0	0

$$\triangleright D_{G1} = S_{-1} \bullet A_2 \bullet A_3$$

Κωδικοποίηση καταστάσεων

$$S\_idle = (0, 0)$$

$$S\_1 = (0, 1)$$

$$S\_2 = (1, 1)$$

Present State			Inputs			Next State		Outputs				
Symbol	$G_1$	$G_0$	Start	$A_2$	$A_3$	$G_1$	$G_0$	Set_E	Clear_E	Set_F	Clear_A_F	Incr_A
State_idle	0	0	0	X	X	0	0	0	0	0	0	0
State_idle	0	0	1	X	X	0	1	0	0	0	1	0
State_1	0	1	X	0	X	0	1	0	1	0	0	1
State_1	0	1	X	1	0	0	1	1	0	0	0	1
State_1	0	1	X	1	1	1	1	1	0	0	0	1
State_2	1	1	X	X	X	0	0	0	0	1	0	0

➤  $D_{G1} = S\_1 \bullet A_2 \bullet A_3$

➤  $D_{G0} = Start \bullet S\_idle + S\_1$

➤ -----

Κωδικοποίηση καταστάσεων

$S\_idle = (0, 0)$

$S\_1 = (0, 1)$

$S\_2 = (1, 1)$

Present State			Inputs			Next State		Outputs				
Symbol	$G_1$	$G_0$	Start	$A_2$	$A_3$	$G_1$	$G_0$	Set_E	Clear_E	Set_F	Clear_A_F	Incr_A
State_idle	0	0	0	X	X	0	0	0	0	0	0	0
State_idle	0	0	1	X	X	0	1	0	0	0	1	0
State_1	0	1	X	0	X	0	1	0	1	0	0	1
State_1	0	1	X	1	0	0	1	1	0	0	0	1
State_1	0	1	X	1	1	1	1	1	0	0	0	1
State_2	1	1	X	X	X	0	0	0	0	1	0	0

➤  $D_{G1} = S\_1 \bullet A_2 \bullet A_3$

➤  $D_{G0} = Start \bullet S\_idle + S\_1$

➤ -----

➤  $set\_E = S\_1 \bullet A_2$

➤  $clr\_E = S\_1 \bullet A_2'$

➤  $set\_F = S\_2$

➤  $clr\_A\_F = start \bullet S\_idle$

➤  $incr\_A = S\_1$

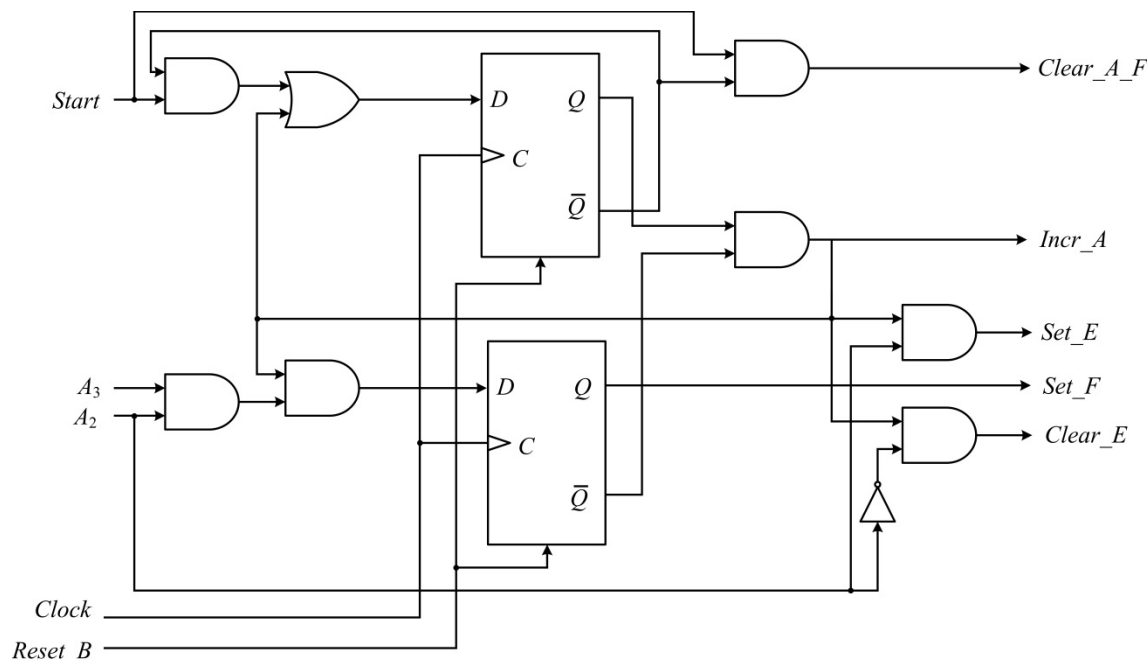
Κωδικοποίηση καταστάσεων

$S\_idle = (0, 0)$

$S\_1 = (0, 1)$

$S\_2 = (1, 1)$

## Σχεδίαση Λογικού Κυκλώματος Ελέγχου (3/3)



- $D_{G1} = S\_1 \bullet A_2 \bullet A_3$
- $D_{G0} = \text{Start} \bullet S\_idle + S\_1$
- $\text{set\_E} = S\_1 \bullet A_2$
- $\text{clr\_E} = S\_1 \bullet A_2'$
- $\text{set\_F} = S\_2$
- $\text{clr\_A\_F} = \text{start} \bullet S\_idle$
- $\text{incr\_A} = S\_1$

Κωδικοποίηση καταστάσεων

$S\_idle = (0, 0)$

$S\_1 = (0, 1)$

$S\_2 = (1, 1)$

- Copyright Πανεπιστήμιο Πατρών,  
Νίκος Φακωτάκης, Γεώργιος Θεοδωρίδης,  
«Ψηφιακή Λογική Σχεδίαση».  
Έκδοση: 1.0 Πάτρα 2015
- Διαθέσιμο στη διαδικτυακή διεύθυνση  
<https://eclass.upatras.gr/courses/EE890/>

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου των διδασκόντων καθηγητών.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ