ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πρόγραμμα Μεταπτυχιακών Σπουδών «Πληροφορική και Εφαρμογές»

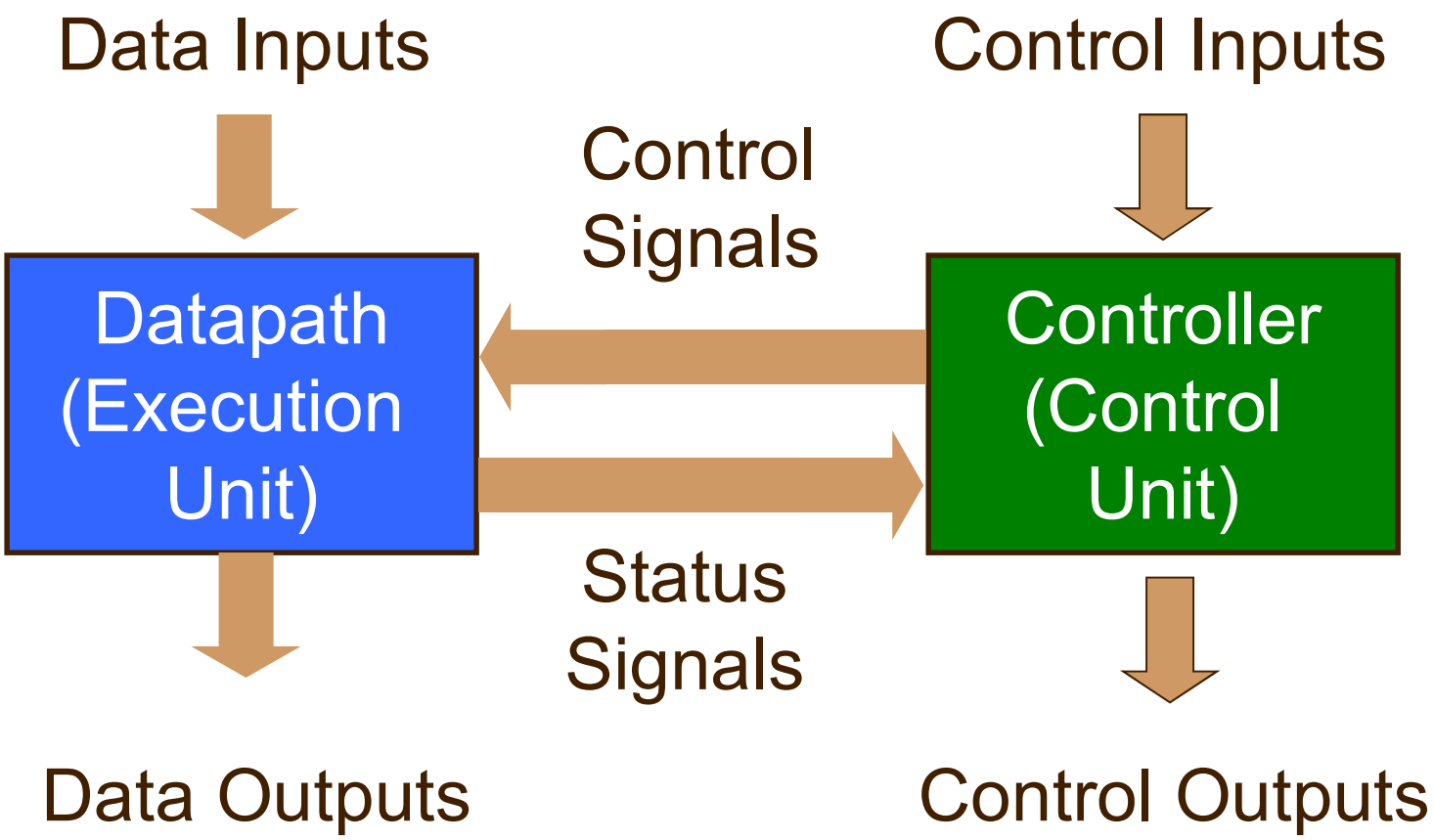
Αρχές Ψηφιακής Τεχνολογίας

Μηχανές πεπερασμένων καταστάσεων

Γιάννης Βογιατζής

2018-2019

Τυπικό Ψηφιακό σύστημα



Datapath (Execution Unit)

- Επεξεργάζεται δεδομένα
- Εκτελεί data-processing tasks
  - αριθμητικές και λογικές πράξεις,
  - shifting κ.λπ.
- Αποτελείται από
  - registers, gates, multiplexers, decoders, adders, comparators, ALUs, etc.
- Παρέχει τα απαραίτητα resources και interconnects μεταξύ τους για να υλοποιήσει συγκεκριμένα task
- Ερμηνεύει τα control signals από τον Controller και παράγει σήματα status για τον Controller

Controller (Control Unit)

- Ελέγχει τη μετακίνηση data στο Datapath με την επιλογή multiplexers και ενεργοποιώντας ή απενεργοποιώντας resources, π.χ.
  - enable signals για registers
  - control signals για muxes
- Παρέχει τα σήματα για να ενεργοποιηθούν διάφορα processing tasks στο Datapath
- Αποφασίζει την ακολουθία των λειτουργιών που εκτελεί το Datapath

Controller

- Μπορεί να είναι προγραμματιζόμενος ή όχι
- Προγραμματιζόμενος
  - Διαθέτει program counter δείχνει στο next instruction
  - Οι εντολές αποθηκεύονται σε μια μνήμη
- Όχι προγραμματιζόμενος
  - Υλοποιεί την ίδια λειτουργία
  - Αναφέρεται και ως
    - “hardwired state machine”

Finite State Machines

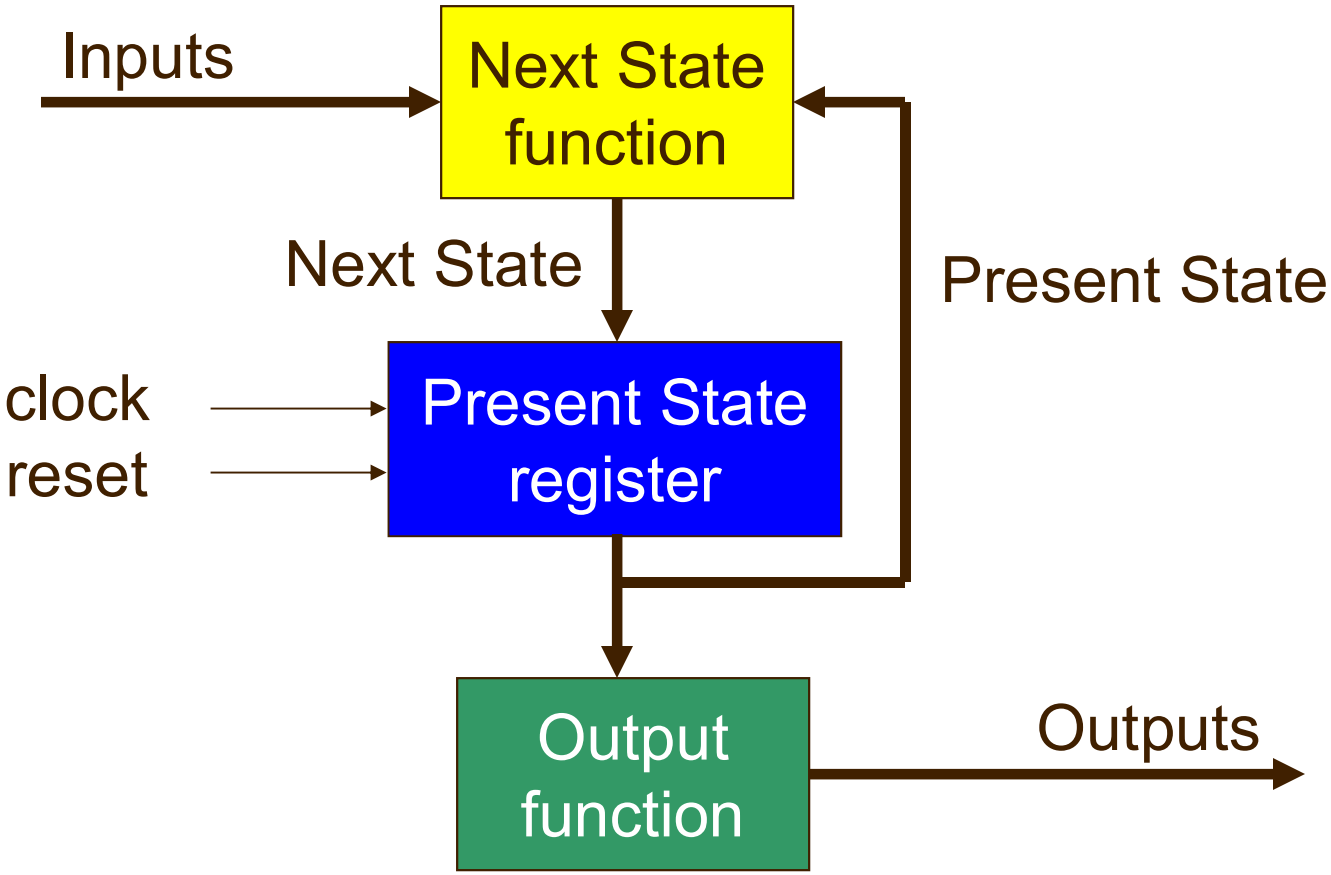
- Τα ψηφιακά συστήματα και οι Controllers μπορούν να περιγραφούν ως Finite State Machines (FSMs)
- Τα FSMs μπορούν να αναπαρασταθούν με **State Diagrams and State Tables**
- Αυτές οι περιγραφές μπορούν να μεταφραστούν σε συνθέσιμο κώδικα VHDL

Finite State Machines (FSMs)

- Οποιοδήποτε κύκλωμα με μνήμη είναι ένα Finite State Machine
- Ο σχεδιασμός ενός FSM περιλαμβάνει:
  - Προσδιορισμό καταστάσεων
  - Προσδιορισμό μεταβάσεων μεταξύ των καταστάσεων
  - Βελτιστοποίηση / ελαχιστοποίηση

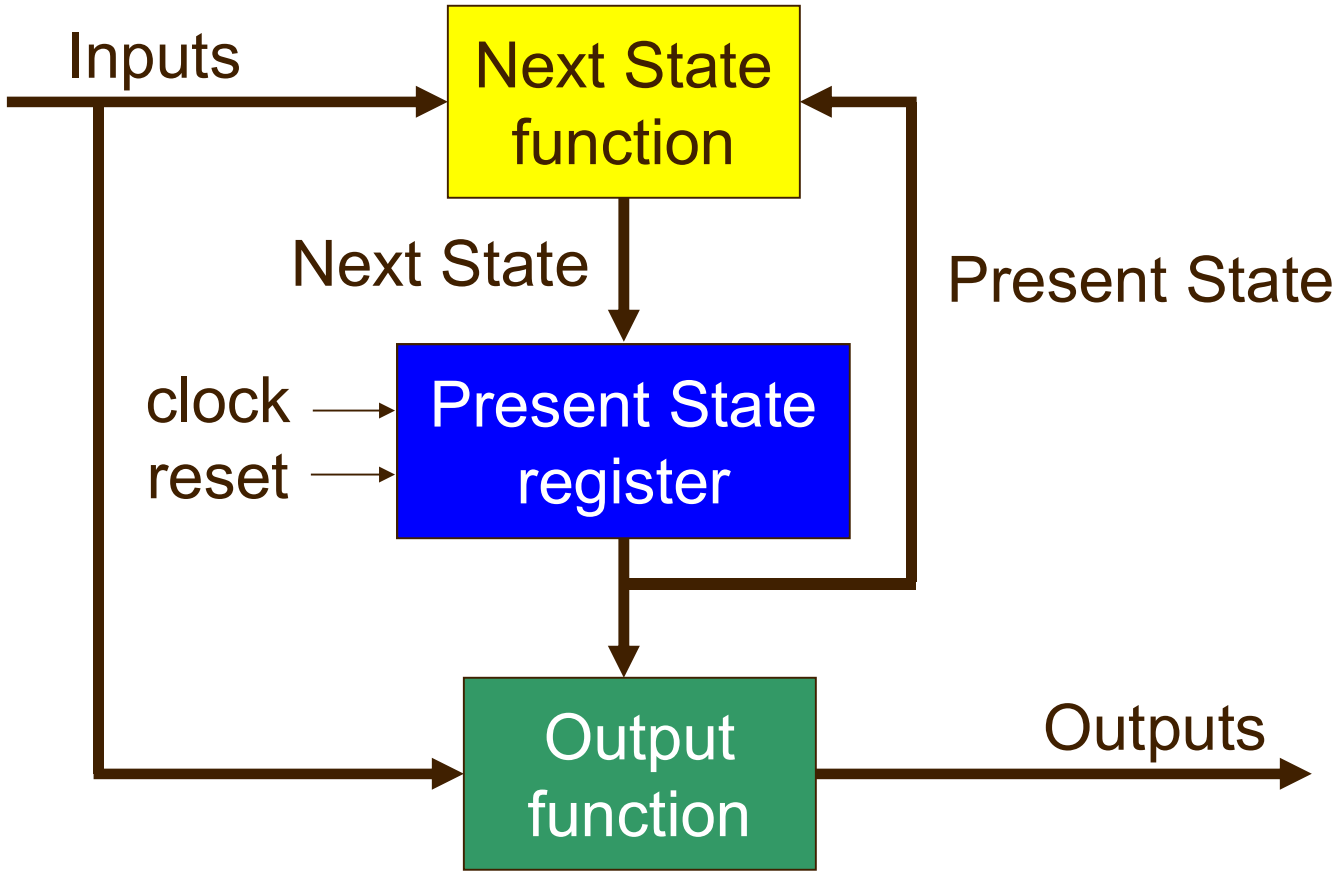
Moore FSM

- Output Is a Function of a Present State Only

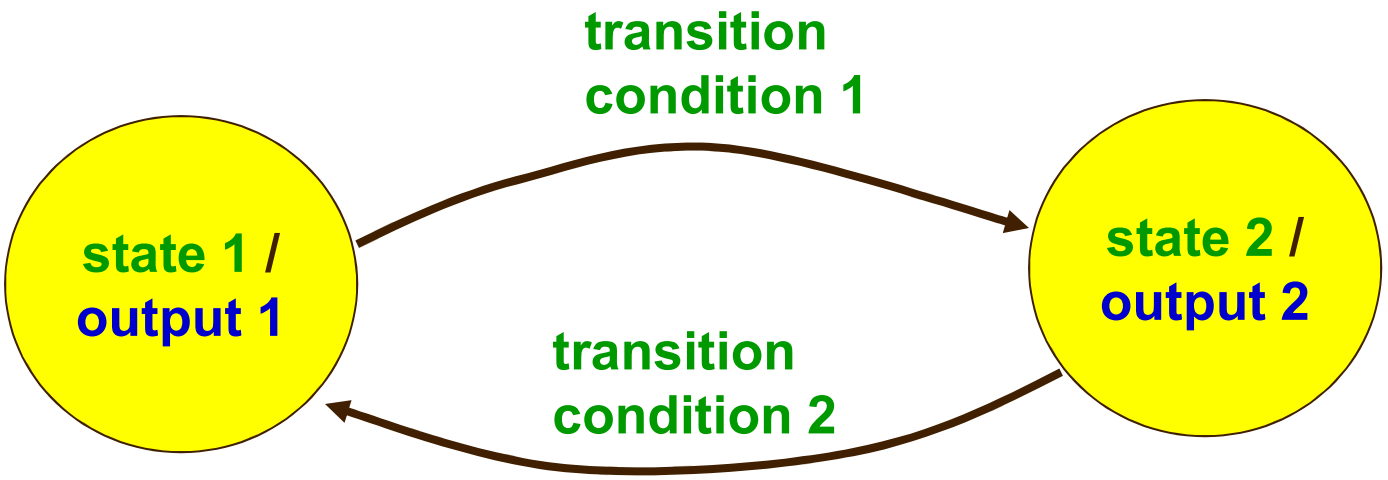


Mealy FSM

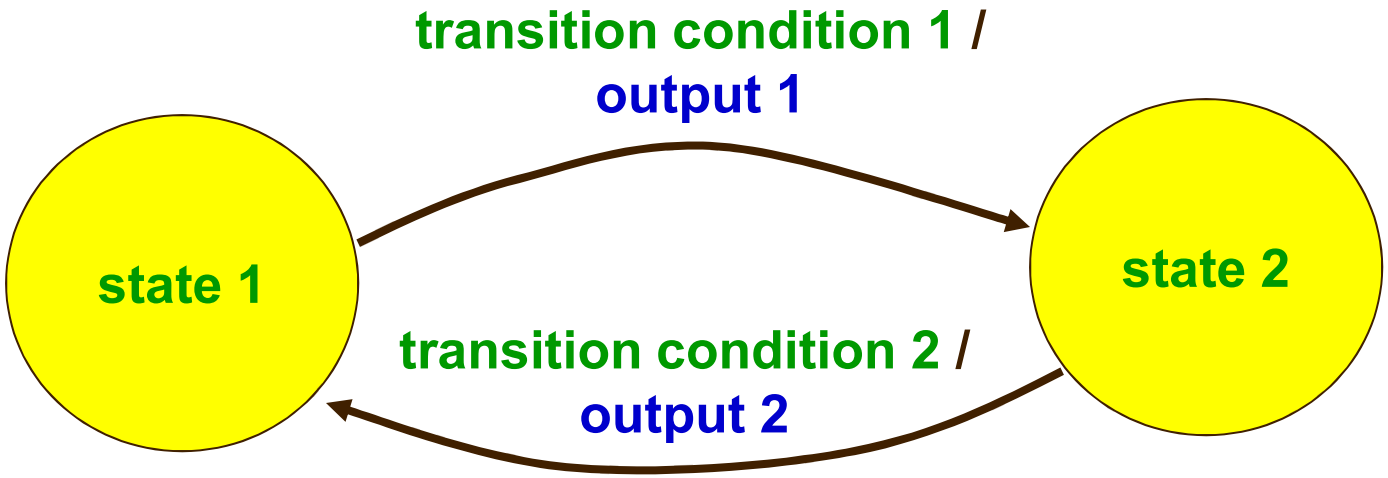
- Output Is a Function of a Present State and Inputs



Moore Machine



Mealy Machine

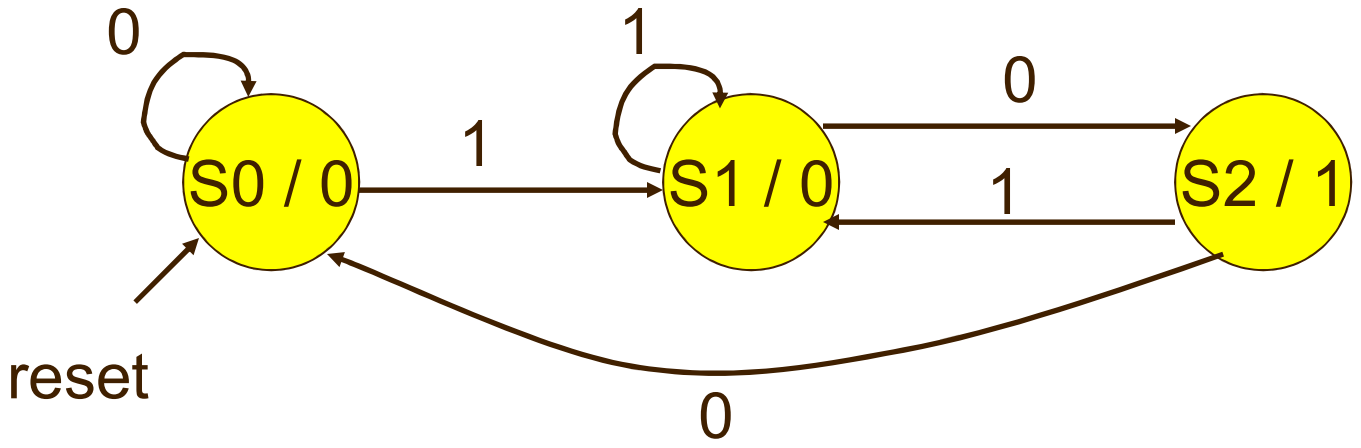


Moore vs. Mealy FSM (2)

- Mealy FSM Computes Outputs as soon as Inputs Change
  - Mealy FSM responds one clock cycle sooner than equivalent Moore FSM
- Moore FSM Has No Combinational Path Between Inputs and Outputs
  - Moore FSM is more likely to have a shorter critical path

Moore FSM - Example 1

- Moore FSM that Recognizes Sequence “10”



Meaning of states:

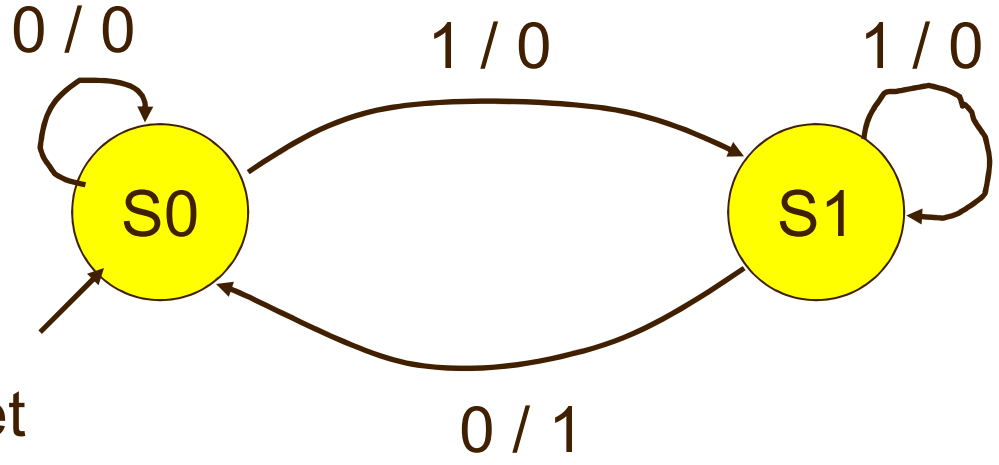
S0: No elements of the sequence observed

S1: “1” observed

S2: “10” observed

Mealy FSM - Example 1

- Mealy FSM that Recognizes Sequence “10”

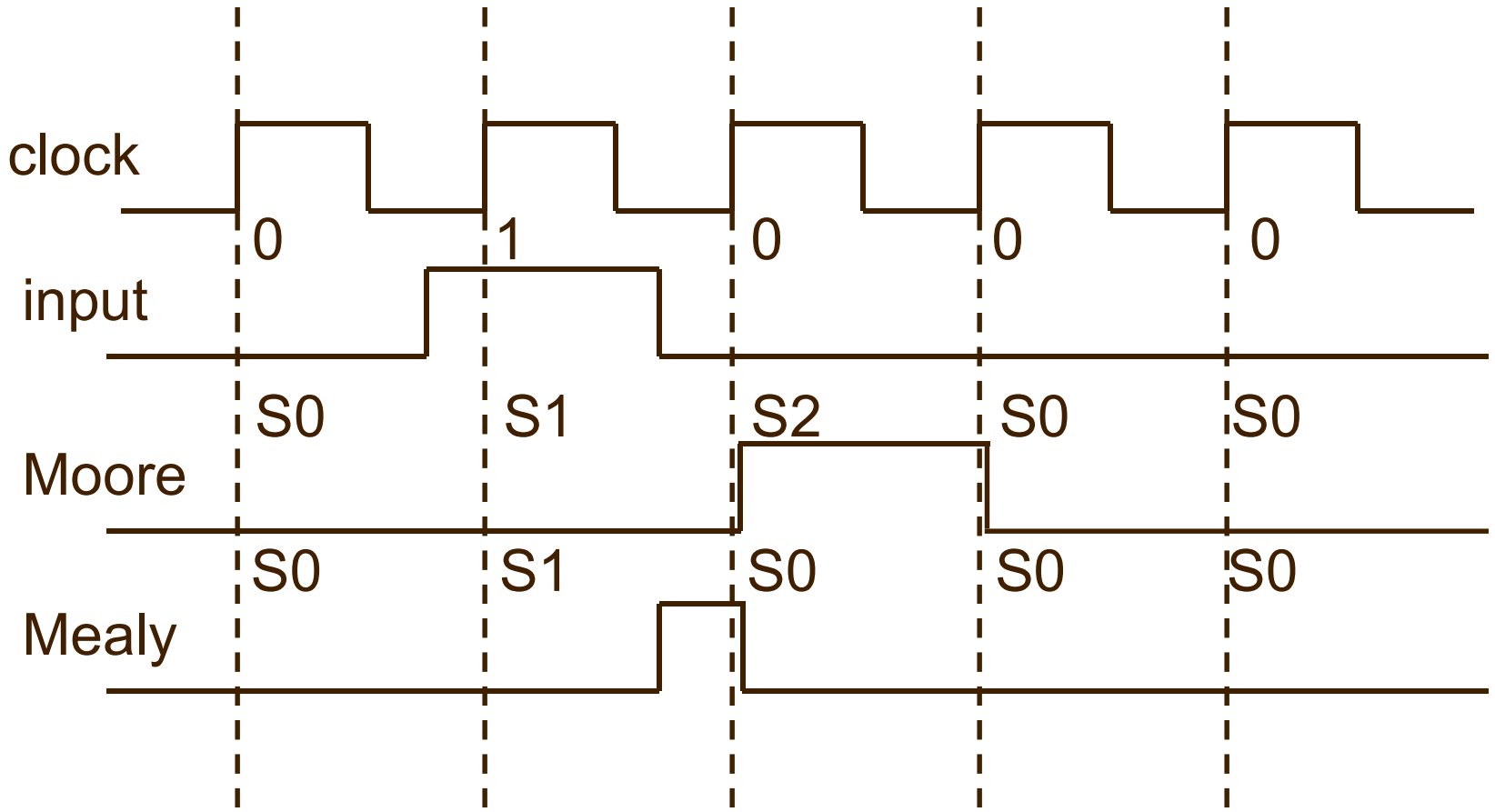


Meaning of states:

S0: No elements of the sequence observed

S1: “1” observed

Moore & Mealy FSMs – Example 1

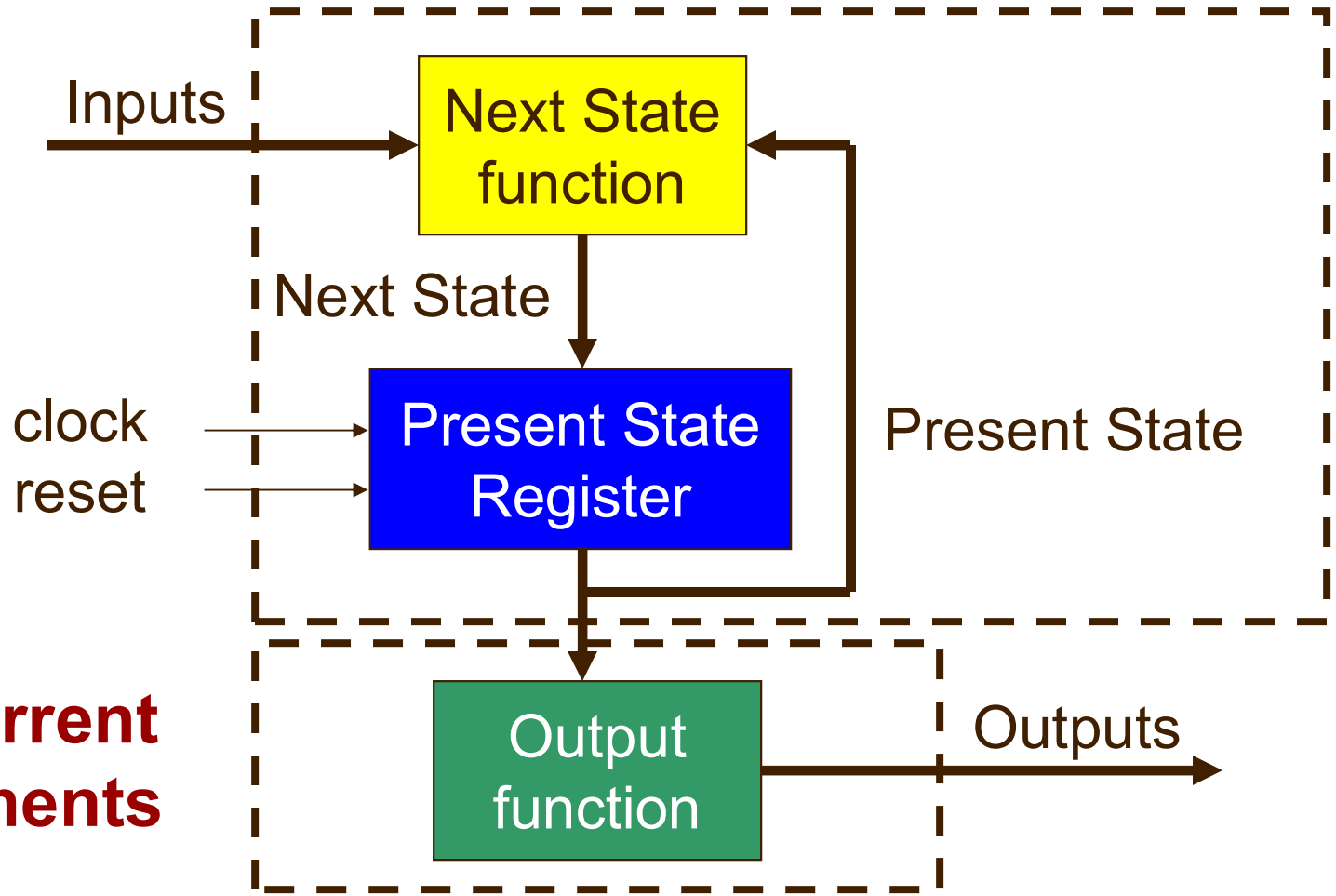


FSMs in VHDL

- Finite State Machines Can Be Easily Described With Processes
- Synthesis Tools Understand FSM Description if Certain Rules Are Followed
  - State transitions should be described in a process sensitive to *clock* and *asynchronous reset* signals only
  - Output function described using rules for combinational logic, i.e.
    - as concurrent statements or
    - as process with all inputs in the sensitivity list

Moore FSM

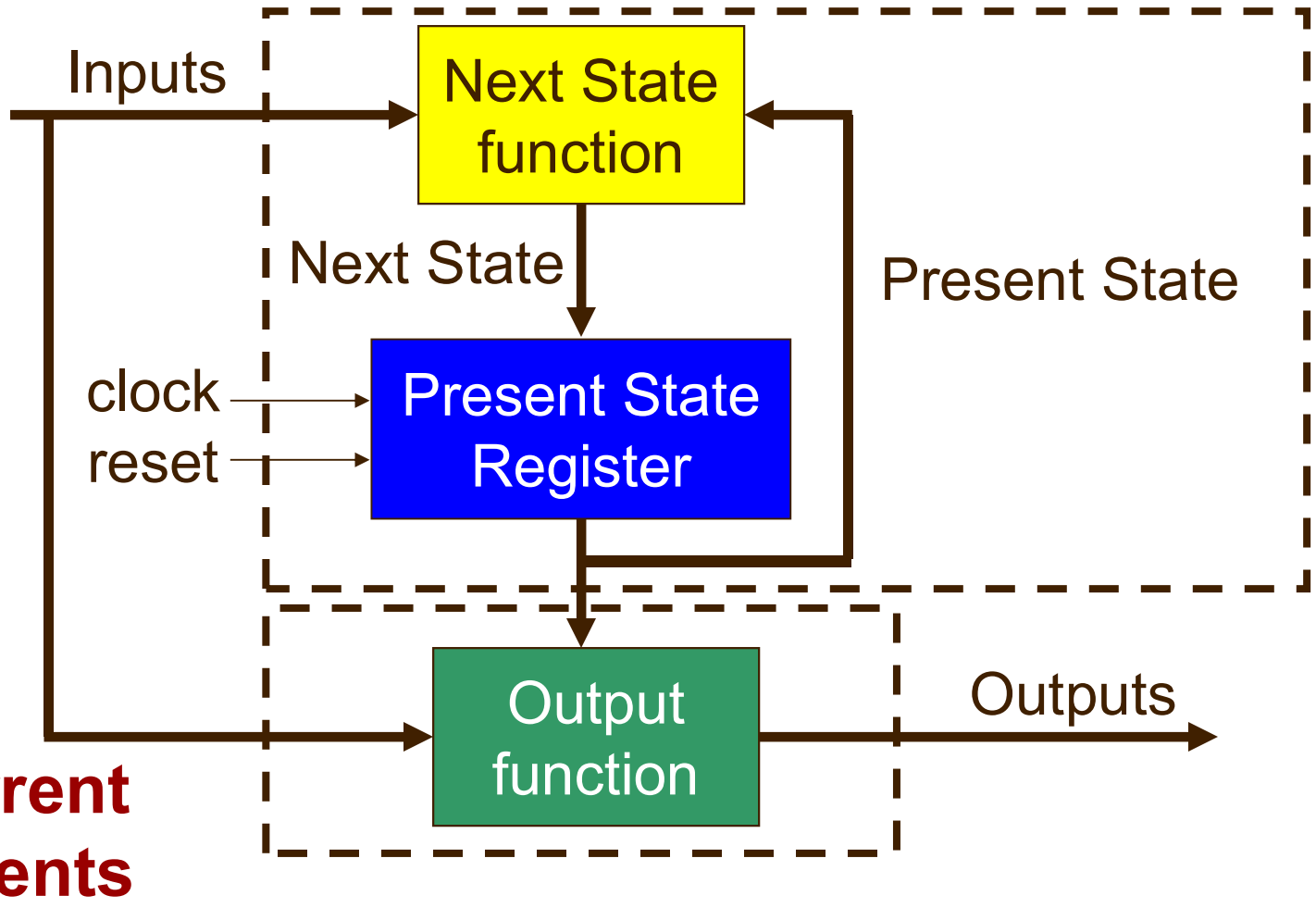
process(clock, reset)



concurrent statements

Mealy FSM

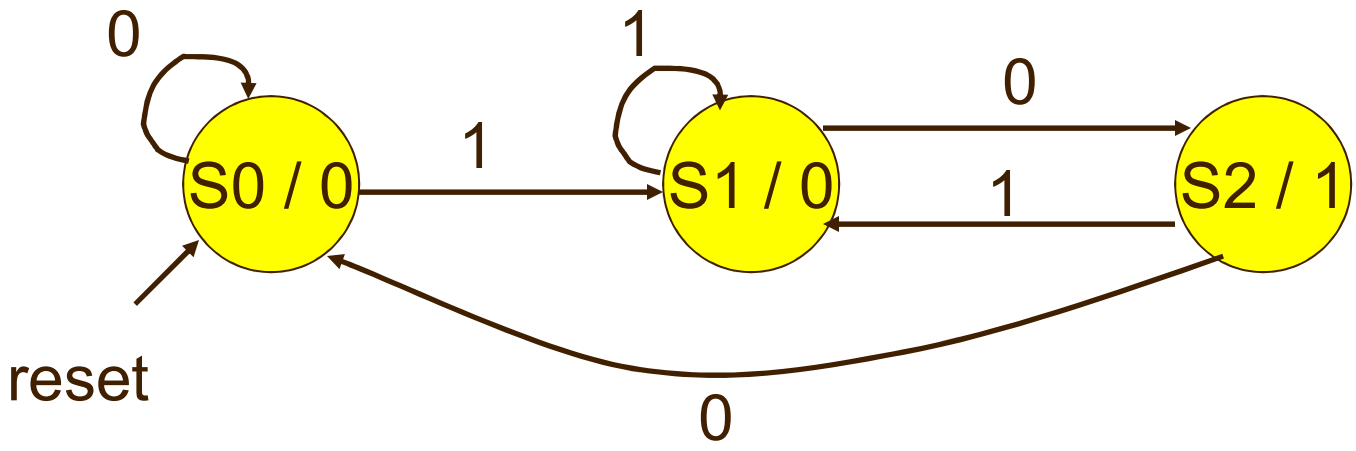
process(clock, reset)



concurrent statements

Moore FSM - Example 1

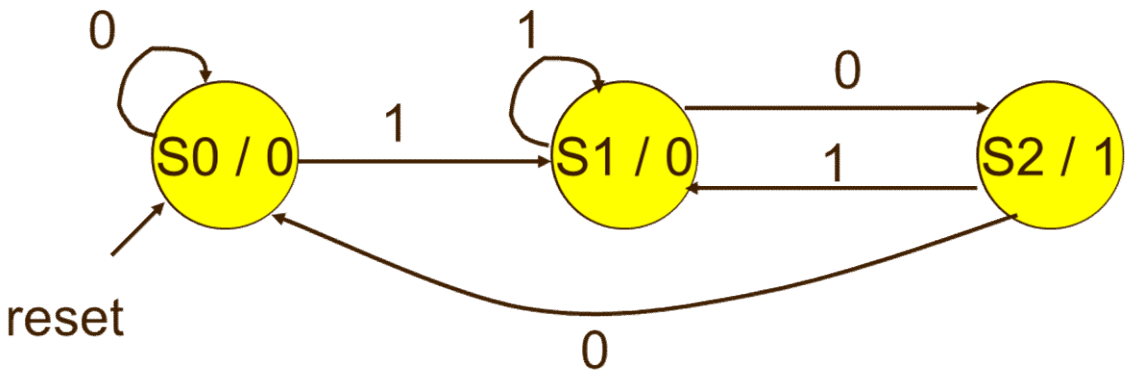
- Moore FSM that Recognizes Sequence “10”



Moore FSM in VHDL

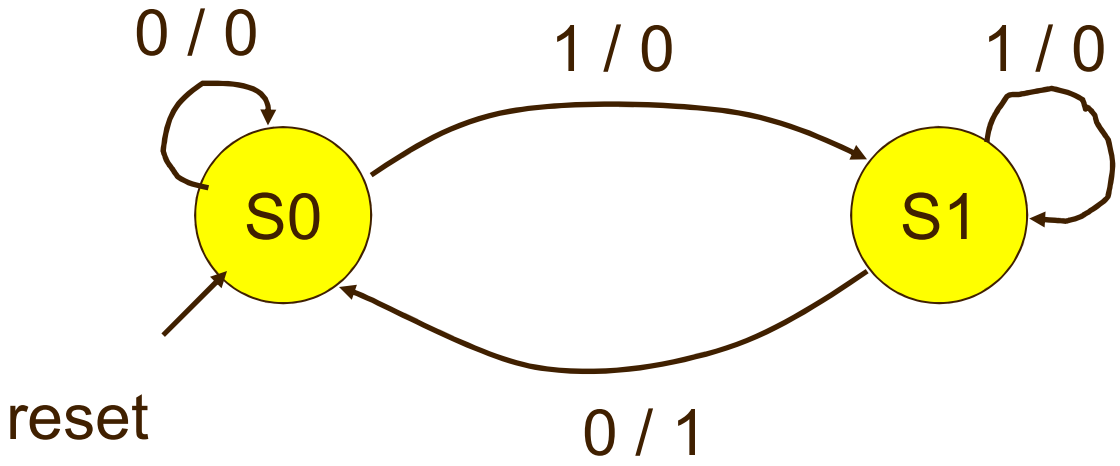
```
PROCESS (clock, reset)
BEGIN
 IF(reset = '1') THEN
  Moore_state <= S0;
 ELSIF (clock = '1' AND clock'event) THEN
  CASE Moore_state IS
   WHEN S0 =>
    IF input = '1' THEN
     Moore_state <= S1;
    ELSE
     Moore_state <= S0;
    END IF;
   WHEN S1 =>
    IF input = '0' THEN
     Moore_state <= S2;
    ELSE
     Moore_state <= S1;
    END IF;
   WHEN S2 =>
    IF input = '0' THEN
     Moore_state <= S0;
    ELSE
     Moore_state <= S1;
    END IF;
   END CASE;
  END IF;
END PROCESS;
Output <= '1' WHEN Moore_state = S2
               ELSE '0';
```

```
TYPE state IS (S0, S1, S2);
SIGNAL Moore_state: state;
```

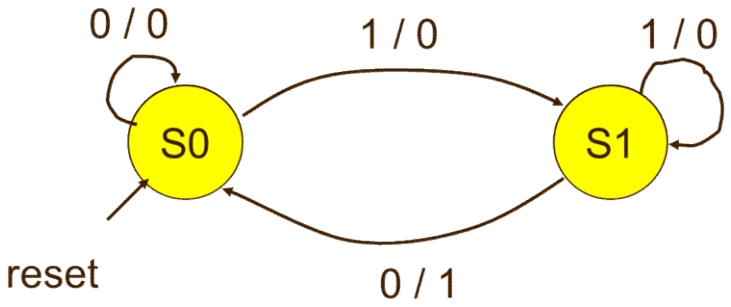


Mealy FSM - Example 1

- Mealy FSM that Recognizes Sequence “10”



Mealy FSM in VHDL




```
TYPE state IS (S0, S1);
SIGNAL Mealy_state: state;
U_Mealy: PROCESS(clock, reset)
BEGIN
 IF(reset = '1') THEN
  Mealy_state <= S0;
 ELSIF (clock = '1' AND clock'event) THEN
  CASE Mealy_state IS
   WHEN S0 =>
    IF input = '1' THEN
     Mealy_state <= S1;
    ELSE
     Mealy_state <= S0;
    END IF;
   WHEN S1 =>
    IF input = '0' THEN
     Mealy_state <= S0;
    ELSE
     Mealy_state <= S1;
    END IF;
   END CASE;
  END IF;
END PROCESS;
Output <= '1' WHEN (Mealy_state = S1 AND input = '0') ELSE '0';
```

Παράδειγμα 1

- Να σχεδιαστεί FSM τύπου Moore που να αναγνωρίζει την ακολουθία 011
- Να σχεδιαστεί FSM τύπου Mealy που να αναγνωρίζει την ακολουθία 011
- Να σχεδιαστεί κύκλωμα που να αναγνωρίζει την ακολουθία 01x1