Finite State Space

Finite State Machines (FSMs)

Pure FSM form is composed of:

- A set of states
- A set of possible inputs (or events)
- A set of possible outputs (or actions)
- A transition function:
  - Given the current state and an input: defines the output and the next state

Finite State Machines (FSMs)

States:

- Represent all possible “situations” that must be distinguished
- At any given time, the system is in exactly one of the states
- There is a finite number of these states

Finite State Machines (FSMs)

An example: a 3-bit counter that increments when “count” input is received

- States: ?

Finite State Machines (FSMs)

An example: a counter

- States: the different combinations of the digits: 000, 001, 010, … 111

- Inputs: ?

Finite State Machines (FSMs)

An example: a counter

- Inputs (events):
  - Only one: “count”
  - We will call this “C”

- Outputs: ?

Finite State Machines (FSMs)

An example: a counter

- Outputs: same as the set of states
- Transition function: ?

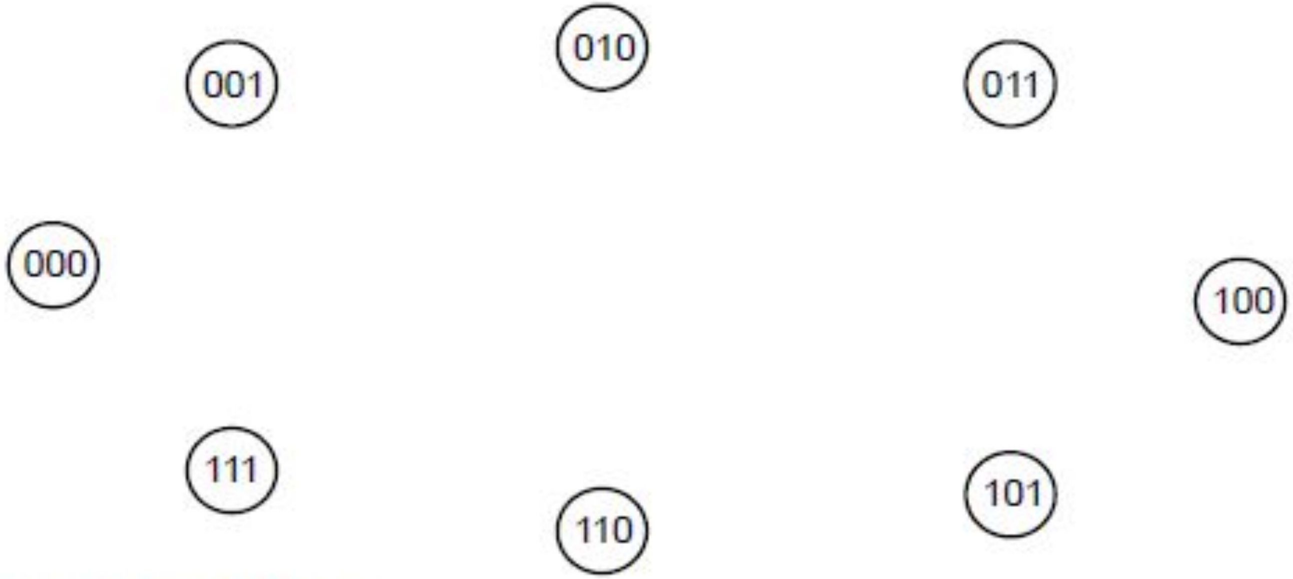
Finite State Machines (FSMs)

An example: a counter

- Transition function:
  - On the count event, transition to the next highest value

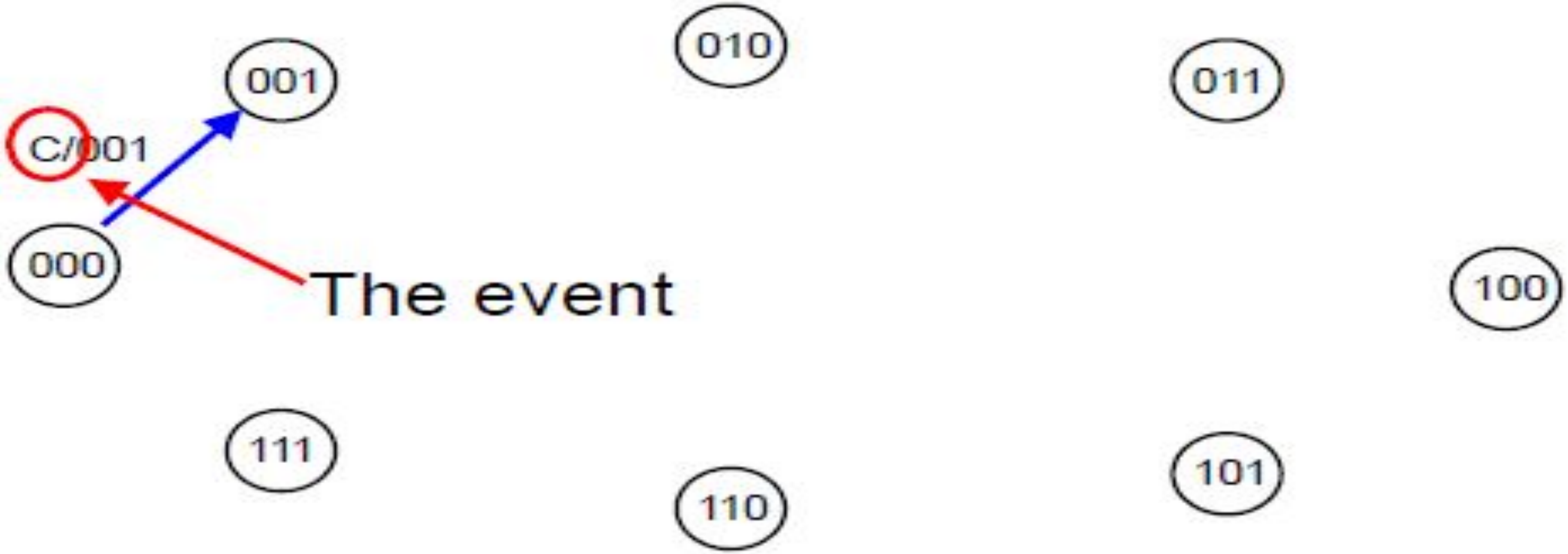
FSM Example: Synchronous Counter

A Graphical Representation:



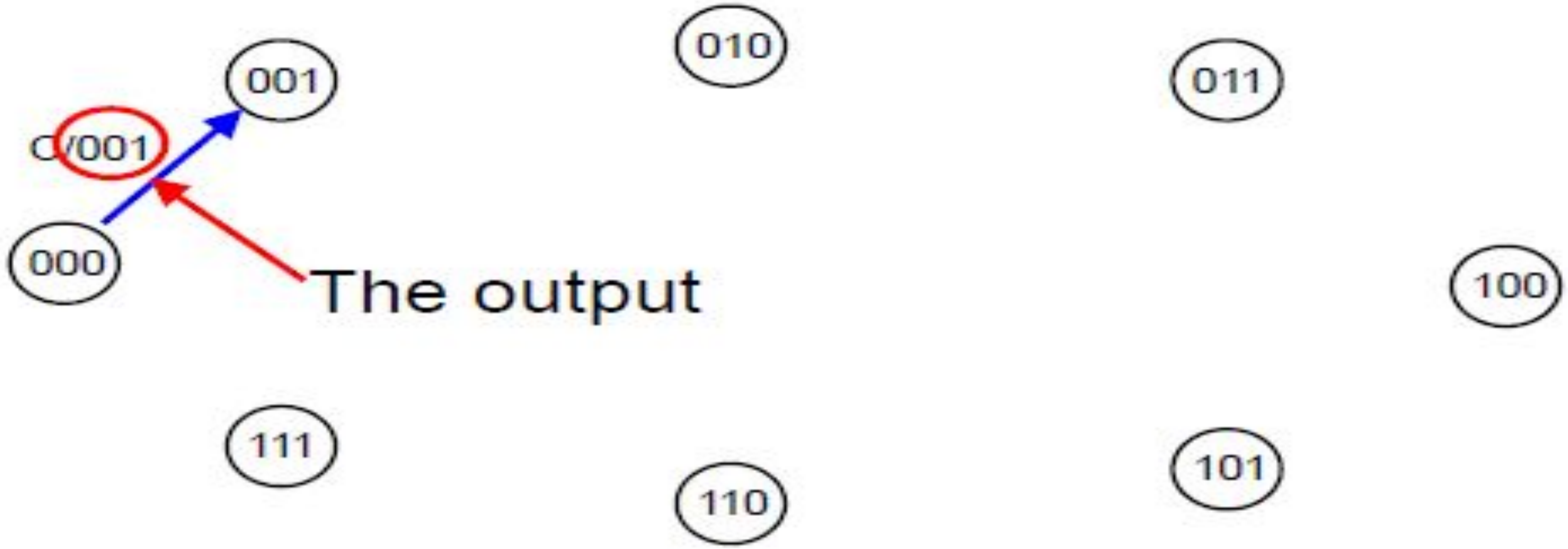
FSM Example: Synchronous Counter

A transition



FSM Example: Synchronous Counter

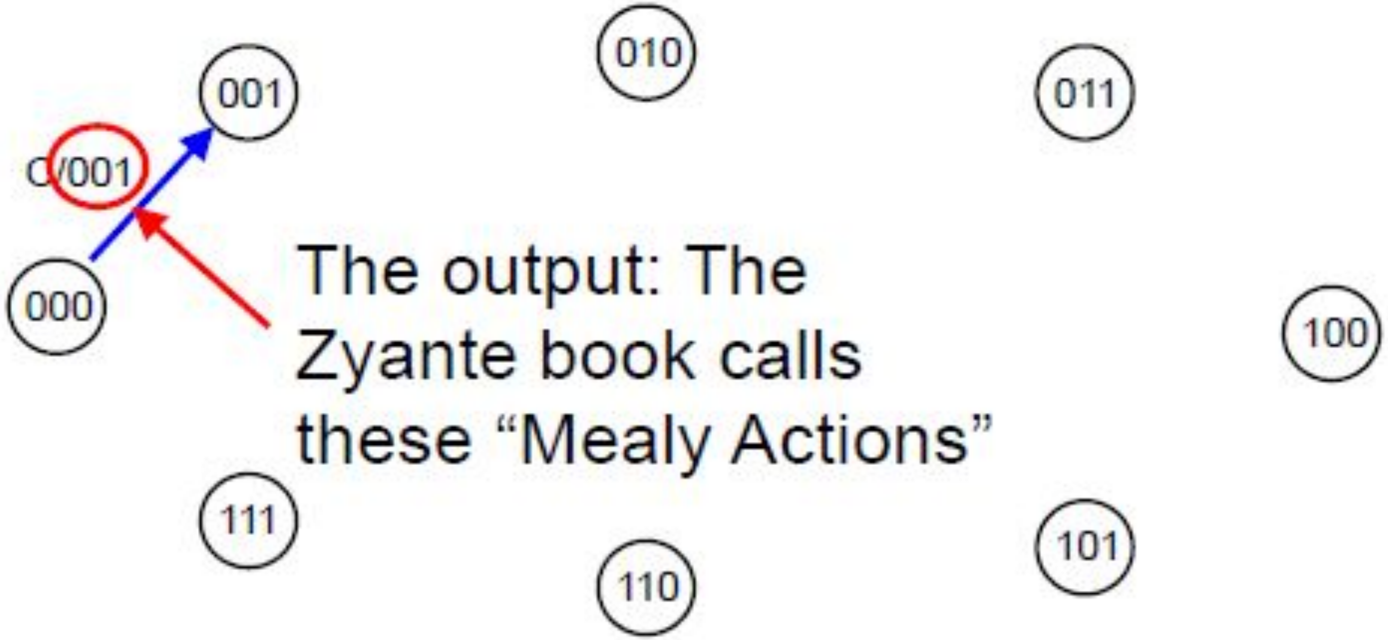
A transition



The output

FSM Example: Synchronous Counter

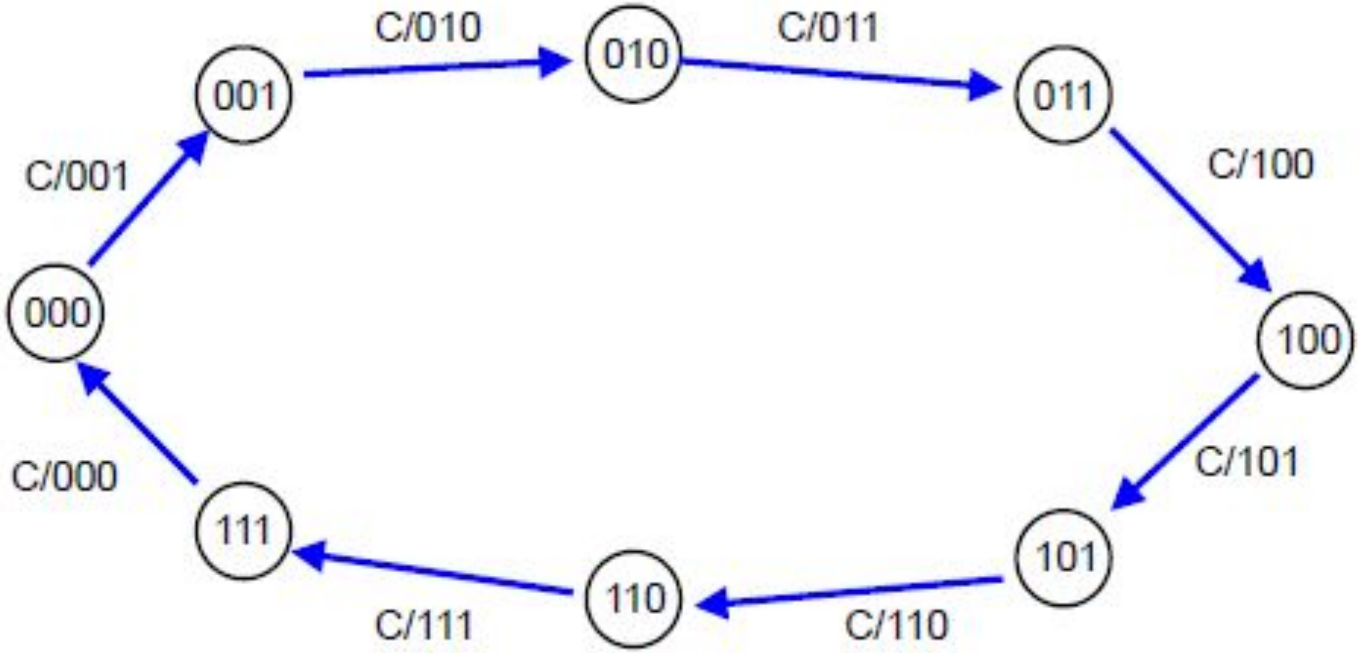
A transition



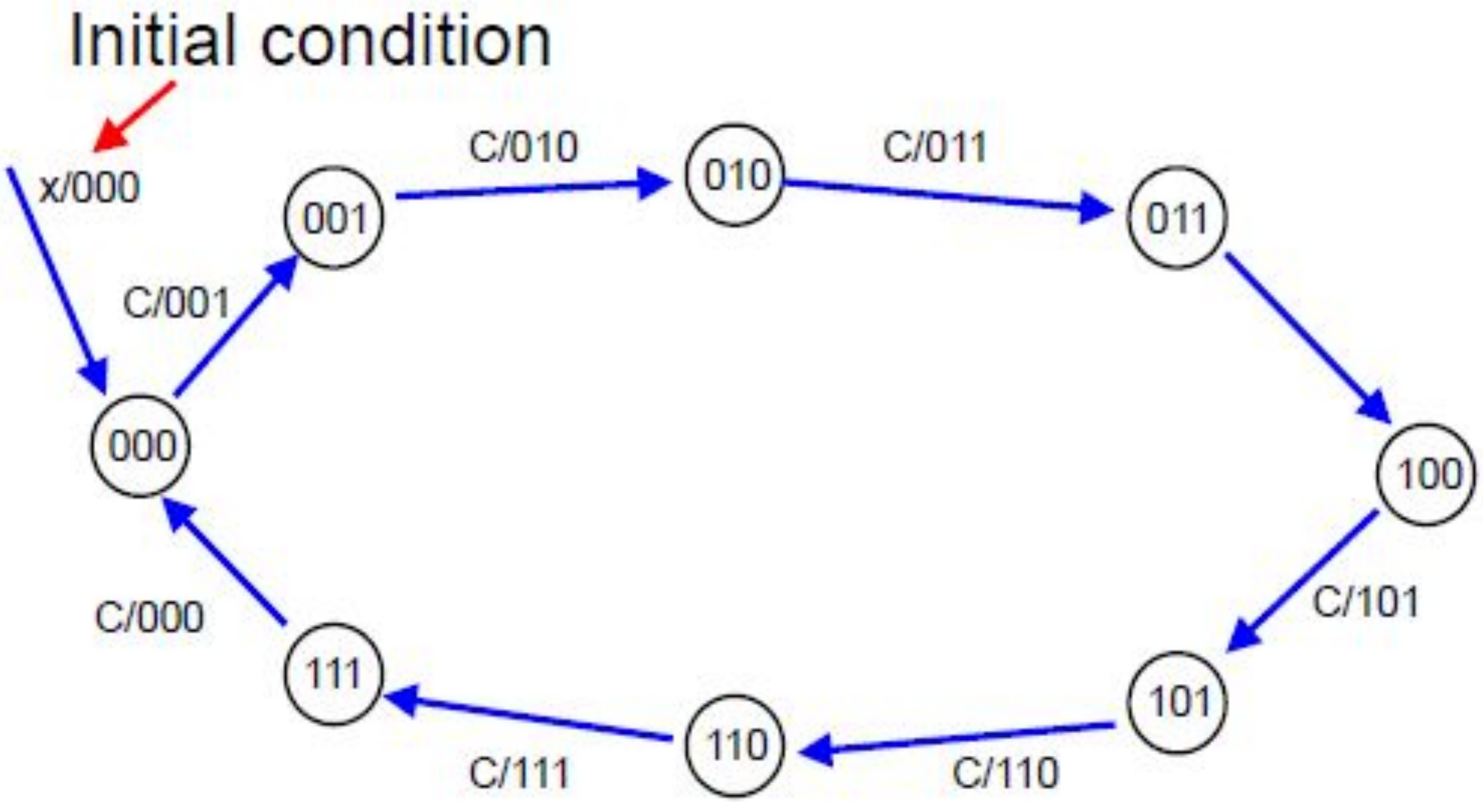
The output: The Zyante book calls these “Mealy Actions”

FSM Example: Synchronous Counter

The full transition set



FSM Example: Synchronous Counter



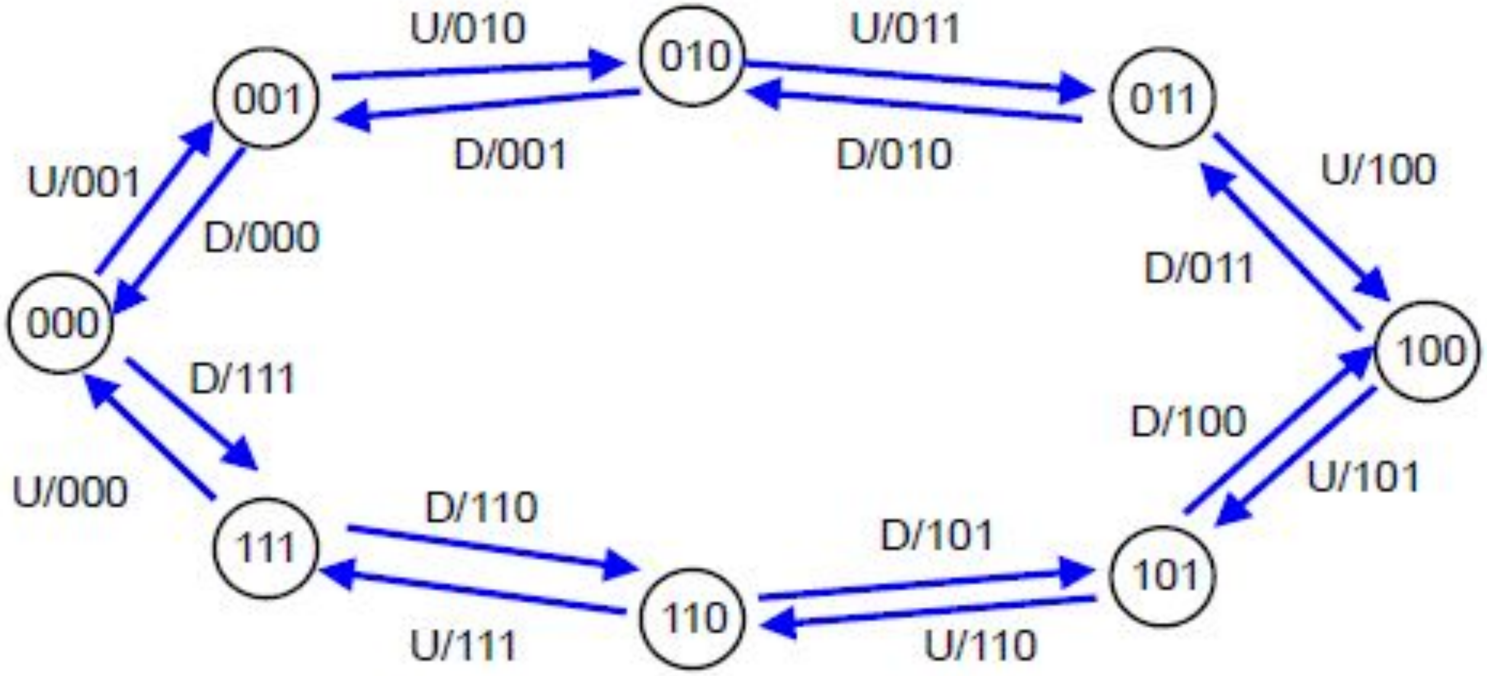
Example II: An Up/Down Counter

Suppose we have two events (instead of one): Count up and count down

- How does this change our state transition diagram?

Example II: An Up/Down Counter

The full transition set


U/010
010
U/011
001
011
D/001
D/010
U/001
U/100
D/000
D/011
000
100
D/111
D/100
U/000
U/101
D/110
D/101
111
101
110
U/111
U/110