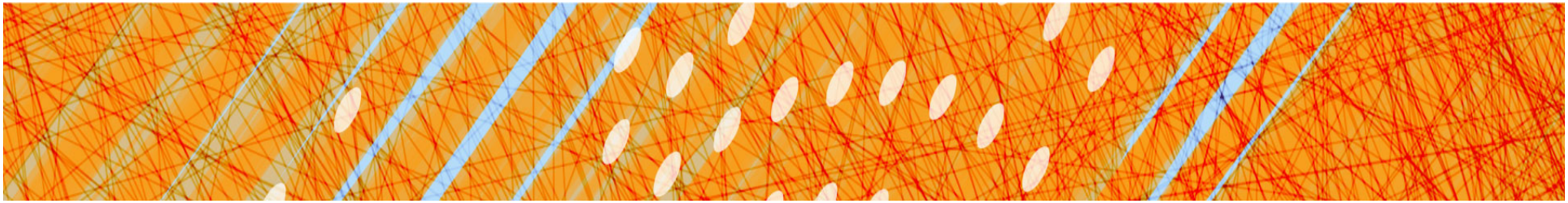


Finite State Machines

CS 3410

Computer System Organization & Programming

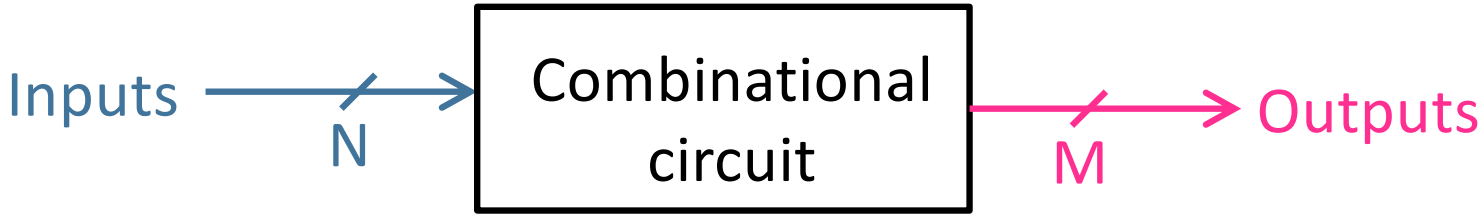
Cornell CIS
COMPUTING AND INFORMATION SCIENCE

[K. Bala, A. Bracy, E. Sirer, and H. Weatherspoon]

Stateful Components

Combinational logic

- Output computed directly from inputs
- System has no internal state
- Nothing depends on the past!



Need:

- to record data
- to build stateful circuits
- a state-holding device

Sequential Logic & Finite State Machines

Finite State Machines

An electronic machine which has

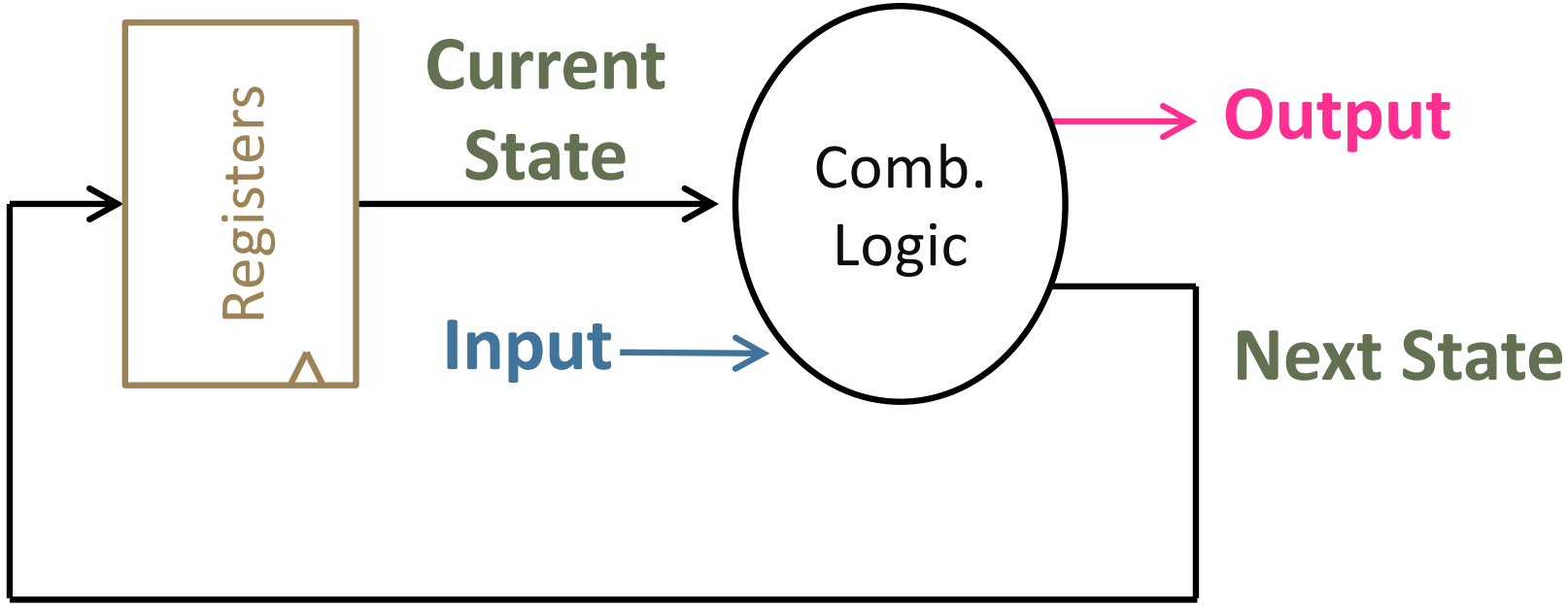
- external inputs
- externally visible outputs
- internal state

Output and next state depend on

- inputs
- current state

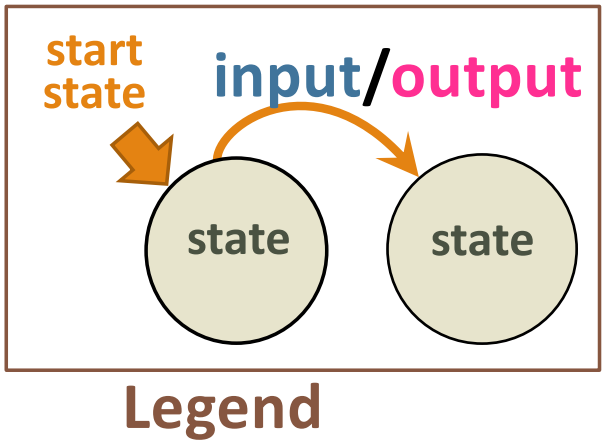
Automata Model

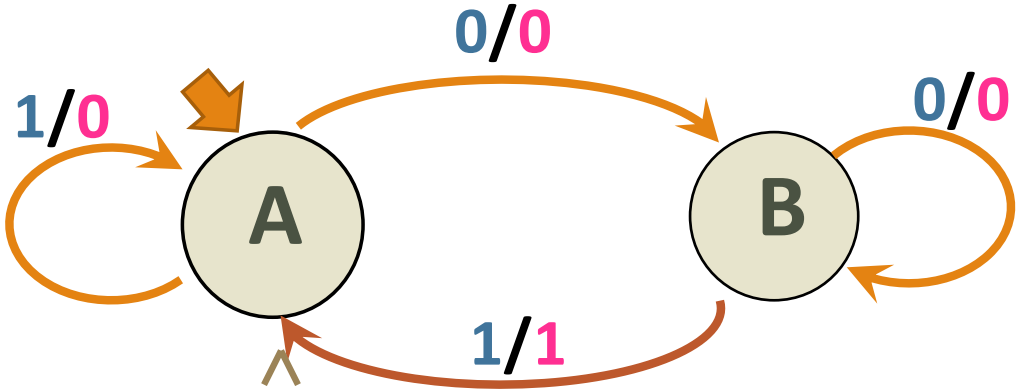
Finite State Machine



- inputs from external world
- outputs to external world
- internal state
- combinational logic

FSM Example





Input: 1 or 0

Output: 1 or 0

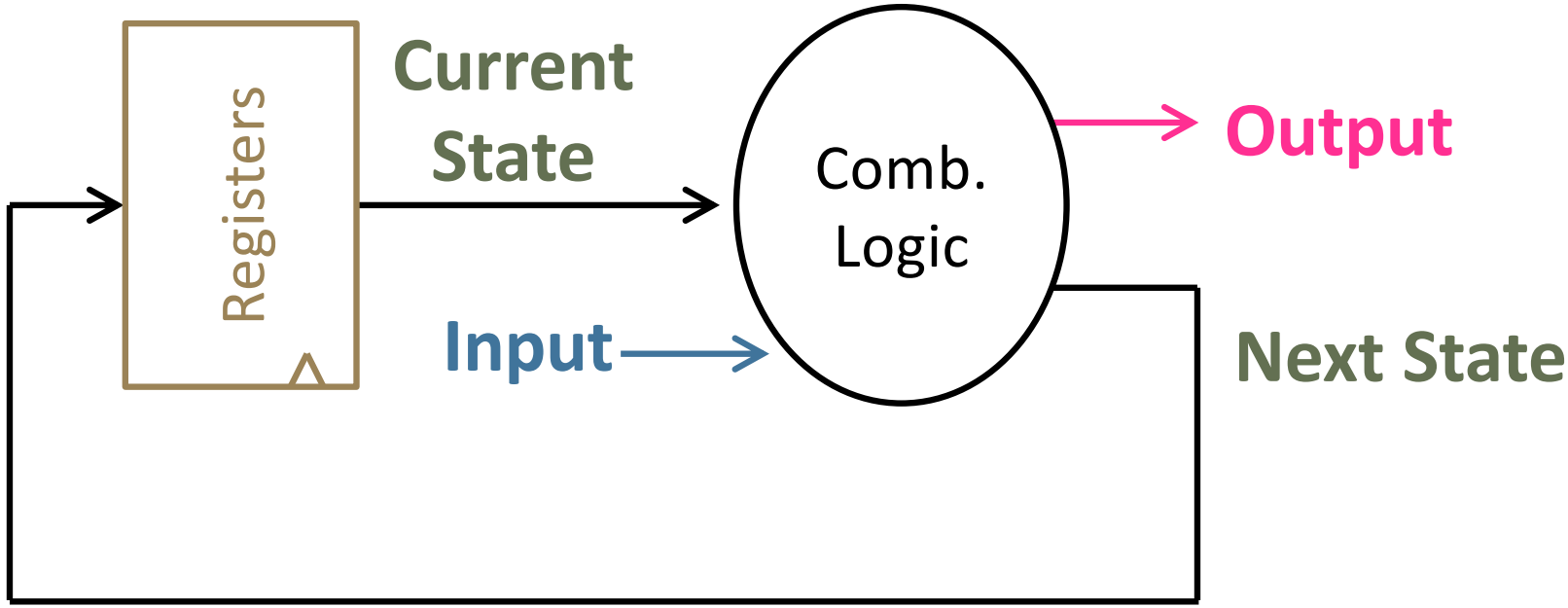
States: A or B

What input pattern is the FSM "looking for"?

[Mealy Machine]

Mealy Machine

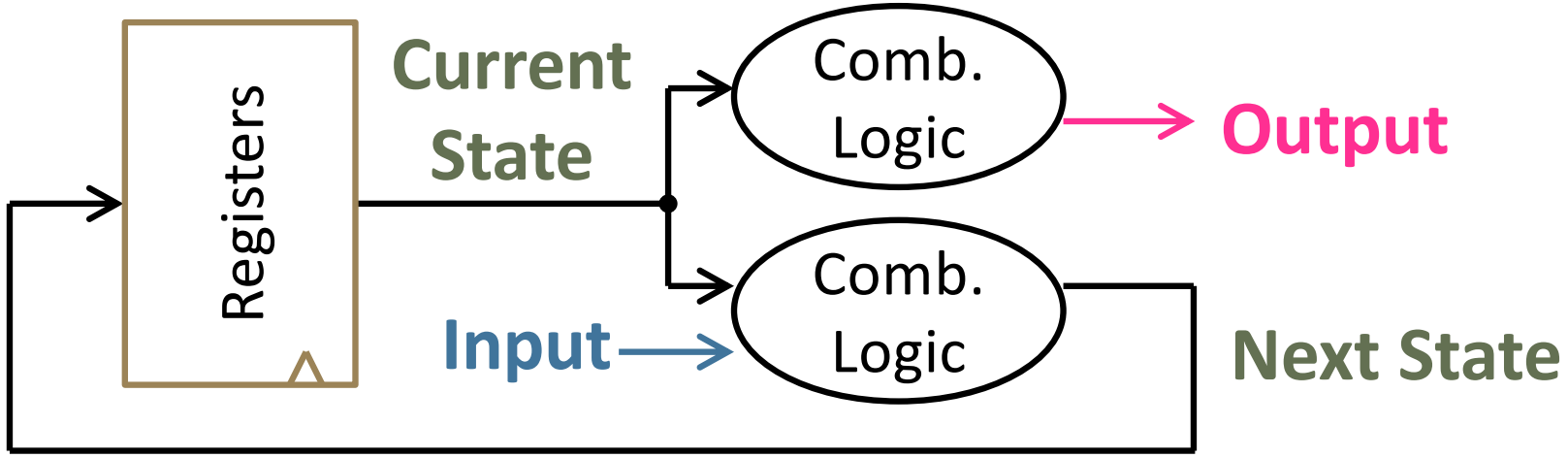
General Case: Mealy Machine

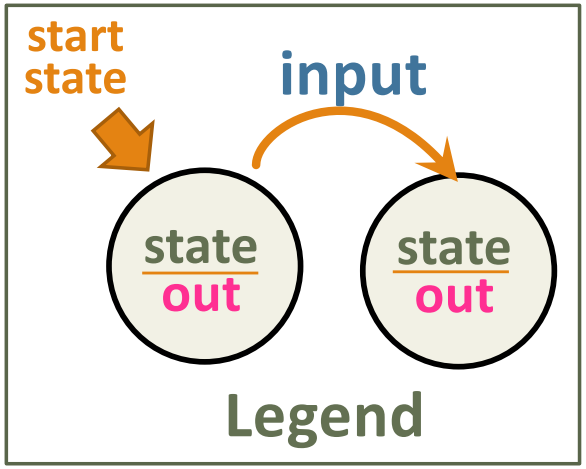


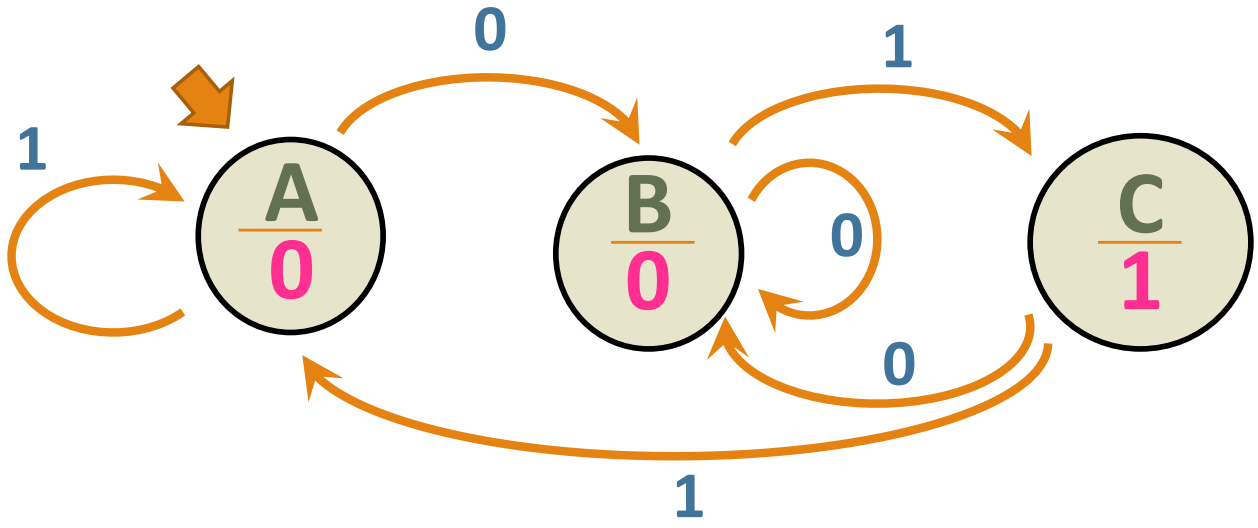
Outputs and next state depend on both current state and input

Moore Machine

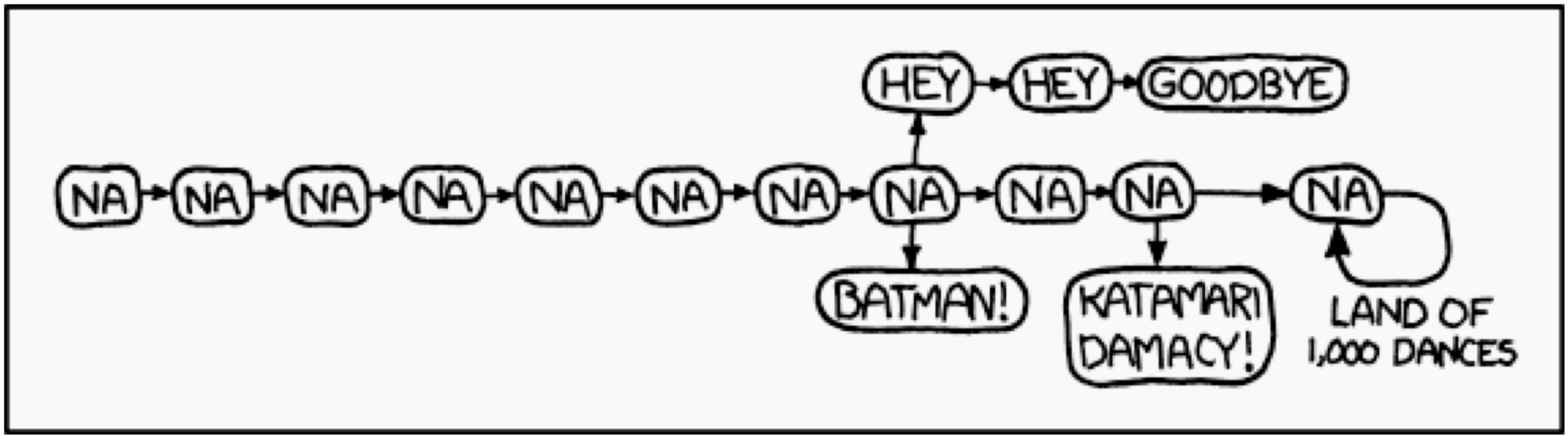
Outputs depend only on current state





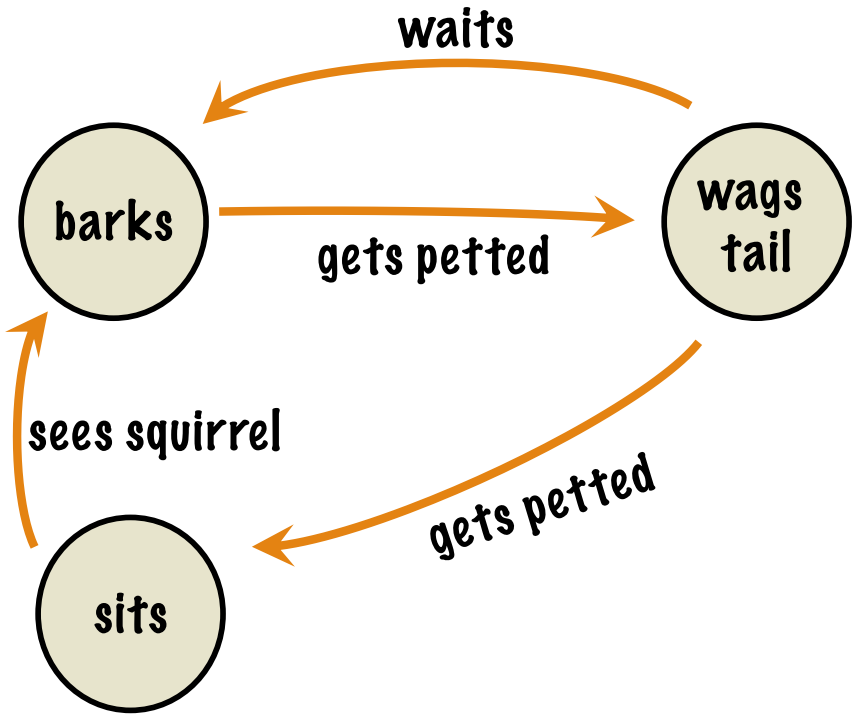


xkcd

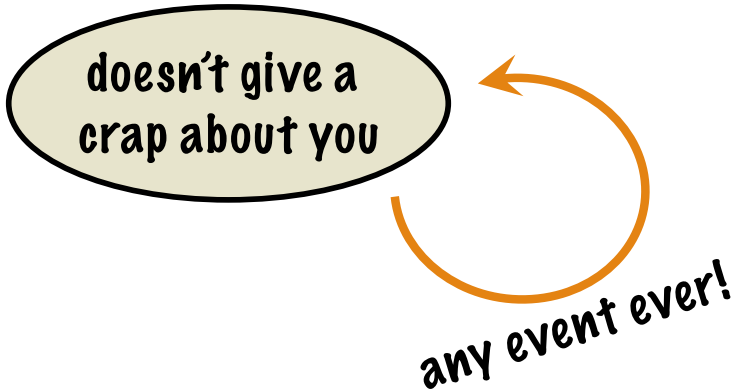

HEY
HEY
GOODBYE
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
BATMAN!
KATAMARI DAMACY!
LAND OF 1,000 DANCES

Why FSMs?

They help us reason about complex behavior.

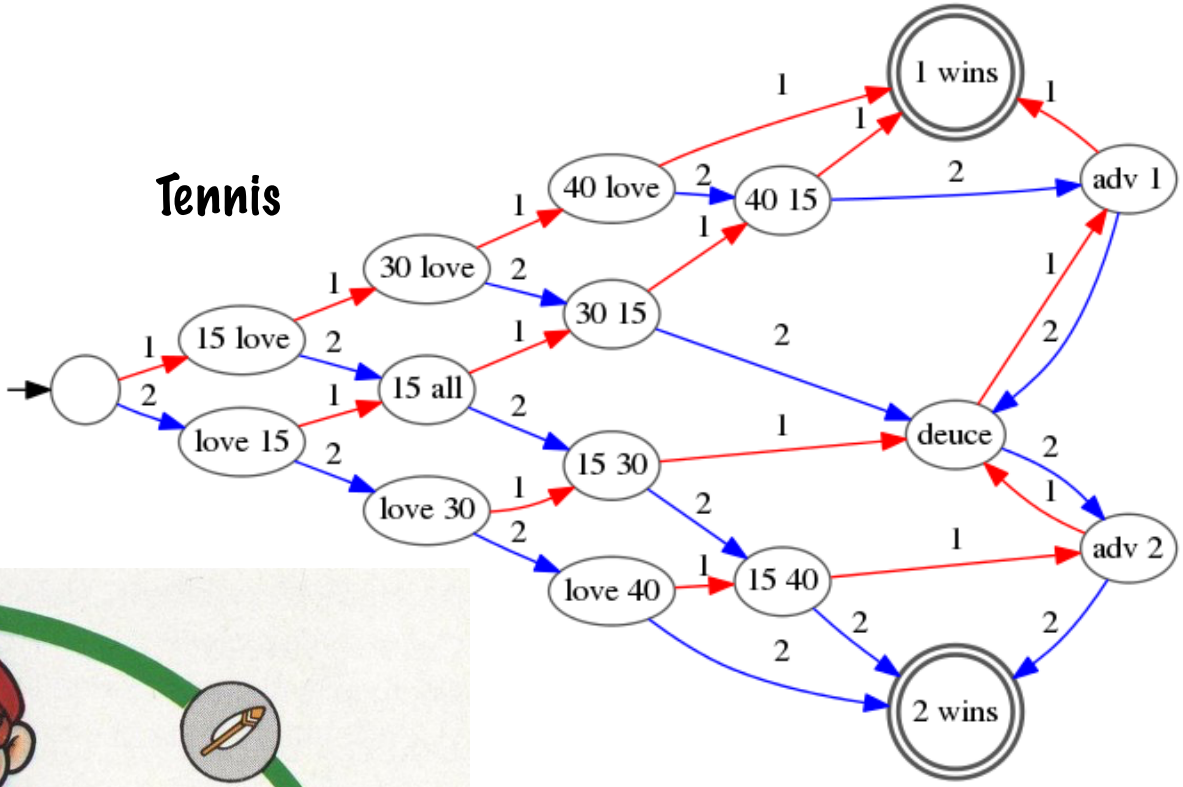


Clicker Question:
What kind of machine is this?
(A) Mealy
(B) Moore
(C) Neither

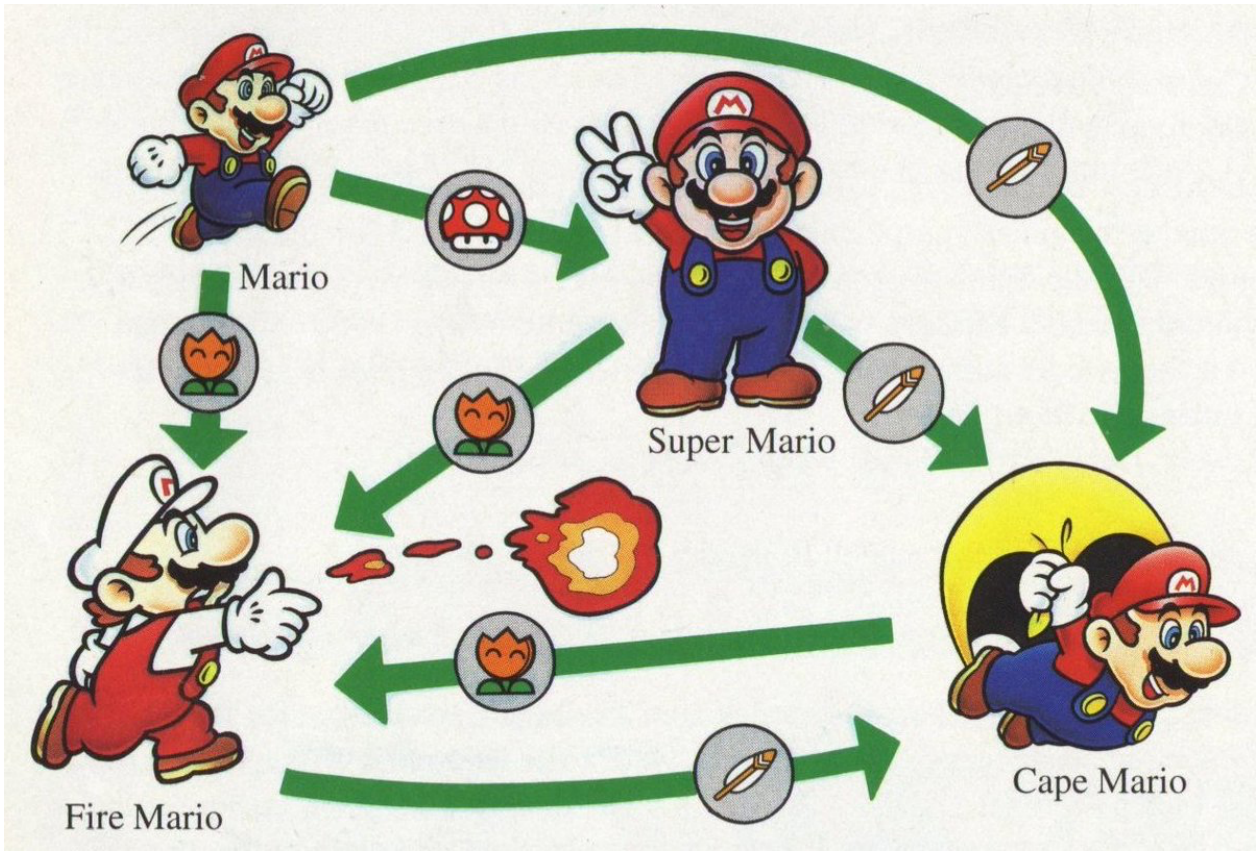


https://learnyousomeerlang.com/finite-state-machines

Other FSMs



http://blog.mikemccandless.com/2014/08/scoring-tennis-using-finite-state.html

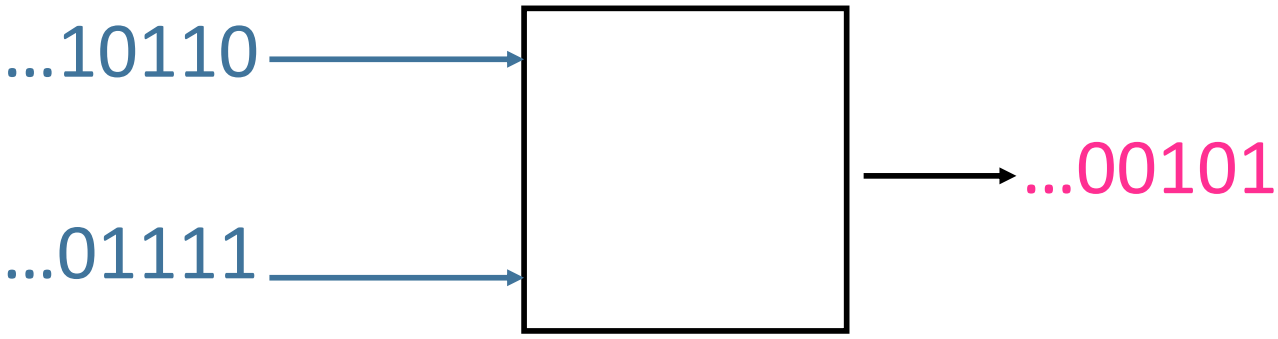


https://www.c-sharpcorner.com/article/understanding-state-design-pattern-by-implementing-finite-state/

Activity: Build a Circuit for a Serial Adder

Add two infinite input bit streams

- streams sent with least-significant-bit (lsb) first



Clicker Question:
How many states are needed to represent this FSM?

(a) 0
(b) 1
(c) 2
(d) 3
(e) 4

Strategy for Building an FSM

(1) Draw a state diagram

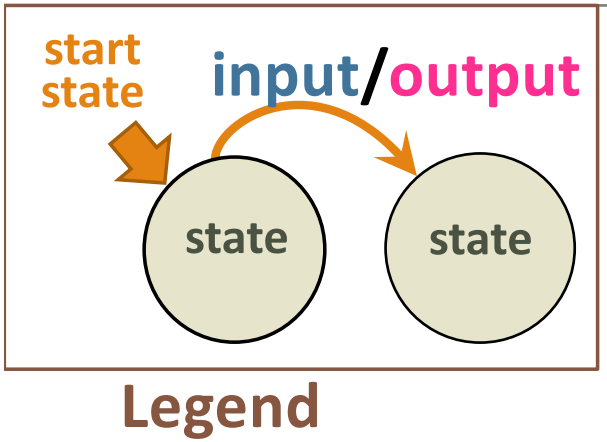
(2) Write output and next-state tables

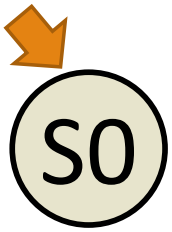
(3) Encode states, inputs, and outputs as bits

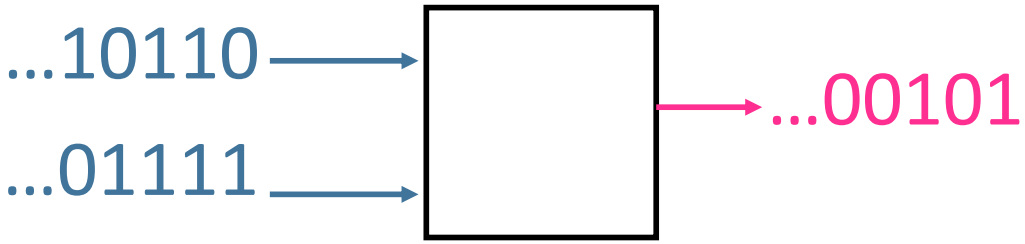
(4) Determine logic equations for next state and outputs

(5) Draw the circuit

Step 1: State Diagram







States:

Inputs: a and b (drawn as 2-bit input ab)

Output: z

Strategy for Building an FSM

(1) Draw a state diagram

(2) Write output and next-state tables

(3) Encode states, inputs, and outputs as bits

(4) Determine logic equations for next state and outputs

(5) Draw the circuit

Step 2: Output & Next State Tables

a	b	Curr State s	z	Next State s′

Strategy for Building an FSM

(1) Draw a state diagram

(2) Write output and next-state tables

(3) Encode states, inputs, and outputs as bits

(4) Determine logic equations for next state and outputs

(5) Draw the circuit

Step 3: Create Bit Encoding

a	b	Curr State s	z	Next State s'

Encode states as bits

S0 =

S1 =

2 states → 1-bit enough

(1-hot also an option)

Copy from previous

Make a binary encoding instead of names

Strategy for Building an FSM

(1) Draw a state diagram

(2) Write output and next-state tables

(3) Encode states, inputs, and outputs as bits

(4) Determine logic equations for next state and outputs

(5) Draw the circuit

Step 4: Create Logic Equations

Determine logic equations for next state and outputs

$S' =$

$Z =$

Strategy for Building an FSM

(1) Draw a state diagram

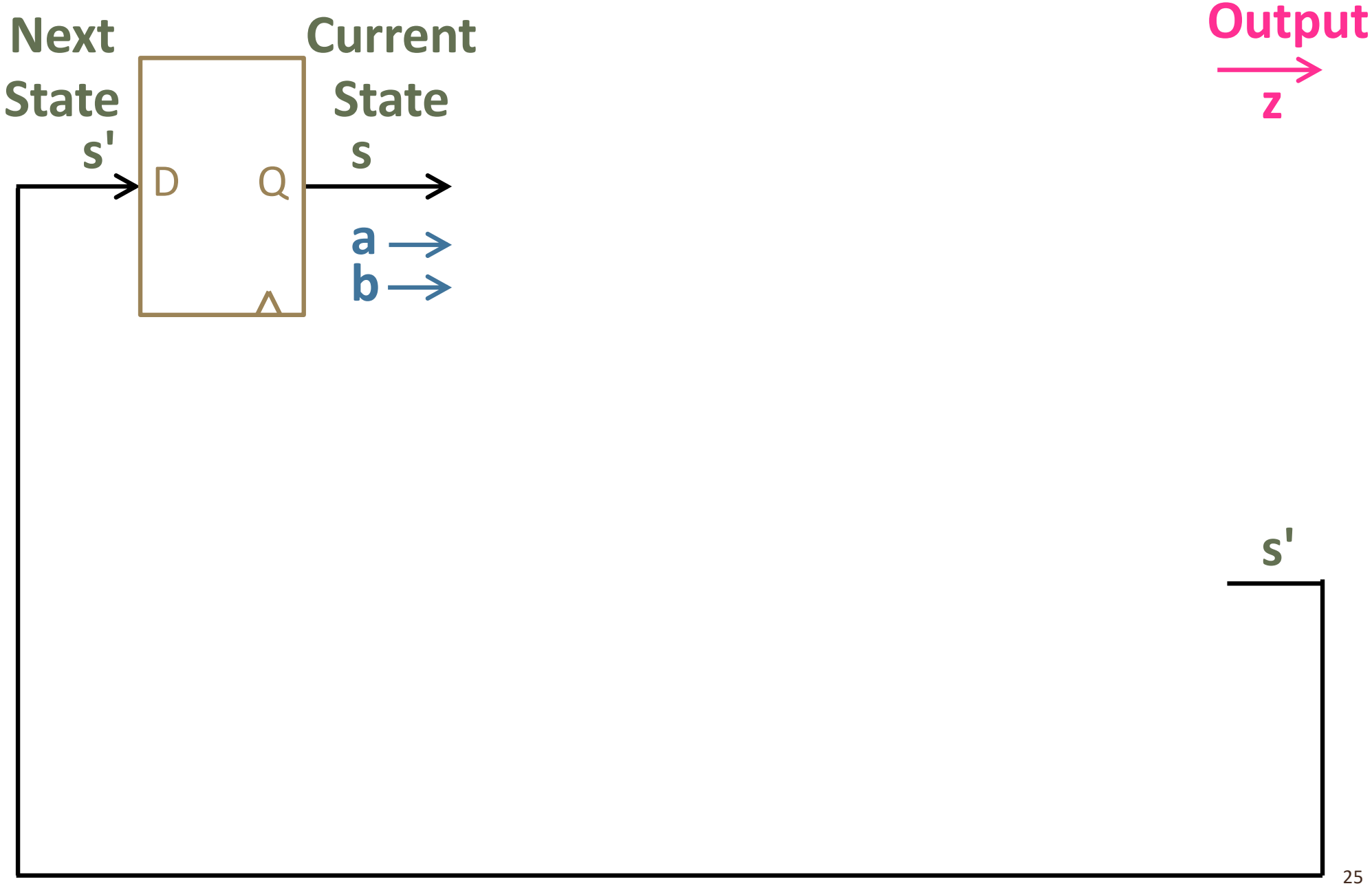
(2) Write output and next-state tables

(3) Encode states, inputs, and outputs as bits

(4) Determine logic equations for next state and outputs

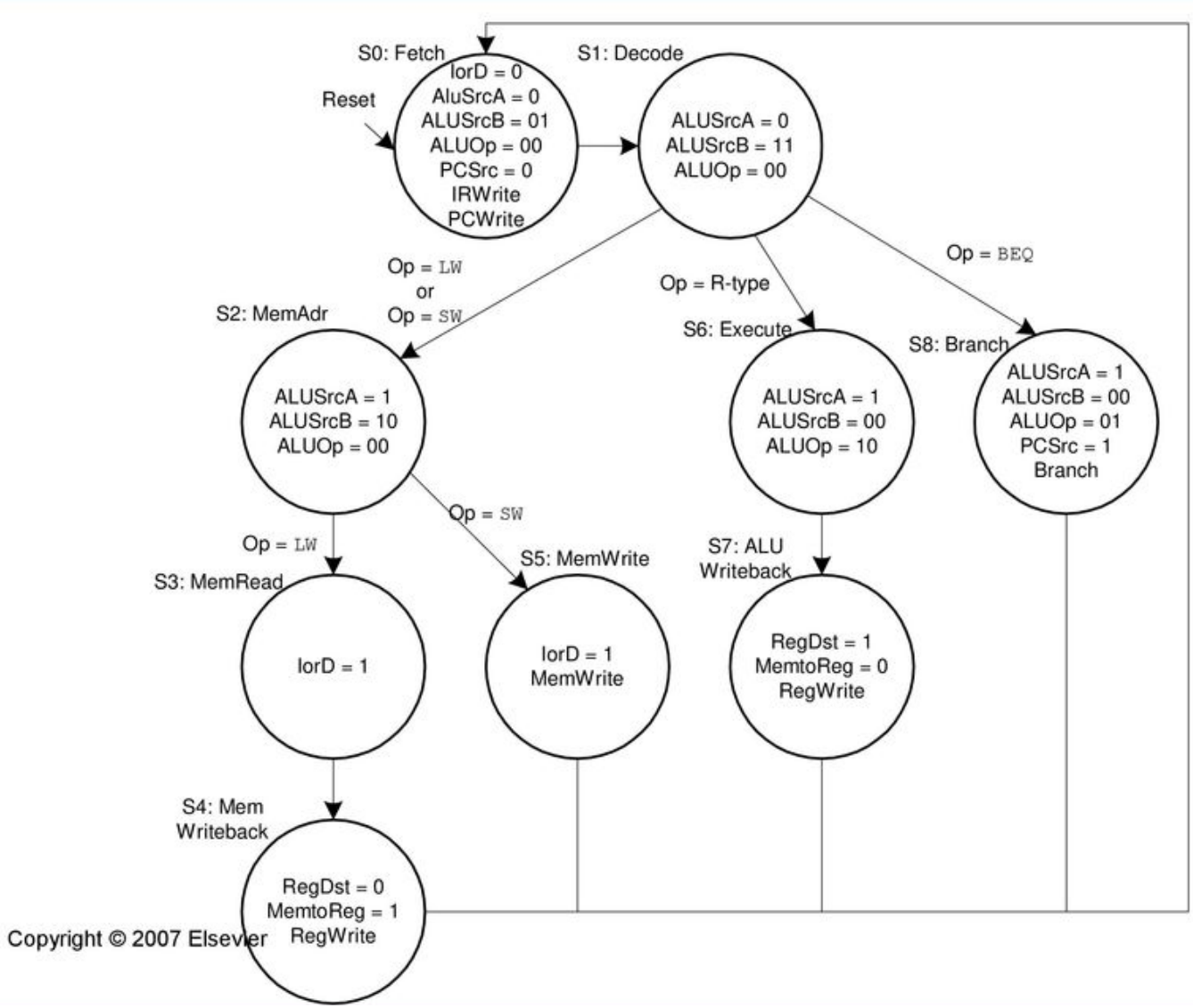
(5) Draw the circuit: **Simplify first!**

Step 5: Draw the Circuit



FSMs in a Processor?

- multi-cycle (non-pipelined) processor



FSMs in a Processor?

- multi-cycle (non-pipelined) processor
- handling cache misses, branch mispredictions, interrupts
- tracking the state of data in your cache (cache coherency)

Consider a finite state machine that takes two inputs, A and B, and generates a single output, Z. Inputs are unsigned binary numbers, entered into the FSM one digit at a time, beginning with the most significant digit. The output, Z, should be the larger of the two numbers. Example: A = 1000 and B = 0101, then Z = 1000 (the value of A).

Draw the state transition diagram (states & arrows) that expresses this FSM. Use the notation AB for inputs (10 means A = 1 and B = 0).

(1) Draw a state diagram

(2) Write output and next-state tables

(3) Encode states, inputs, and outputs as bits

(4) Determine logic equations for next state and outputs

(5) Draw the circuit

Clicker Question:

How many states are needed to represent this FSM?

(a) 0
(b) 1
(c) 2
(d) 3
(e) 4