

```
While True:
```

```
    ...
```

```
        if ( ):
```

```
            do1()
```

```
        elif ( ):
```

```
            do2()
```

```
    ...
```

```
...
```

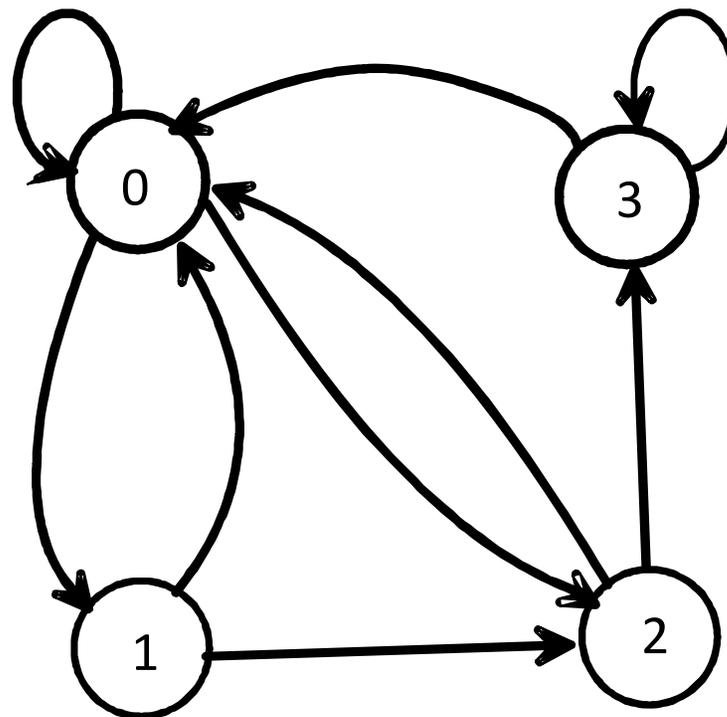
```
do1()
```

```
do2()
```

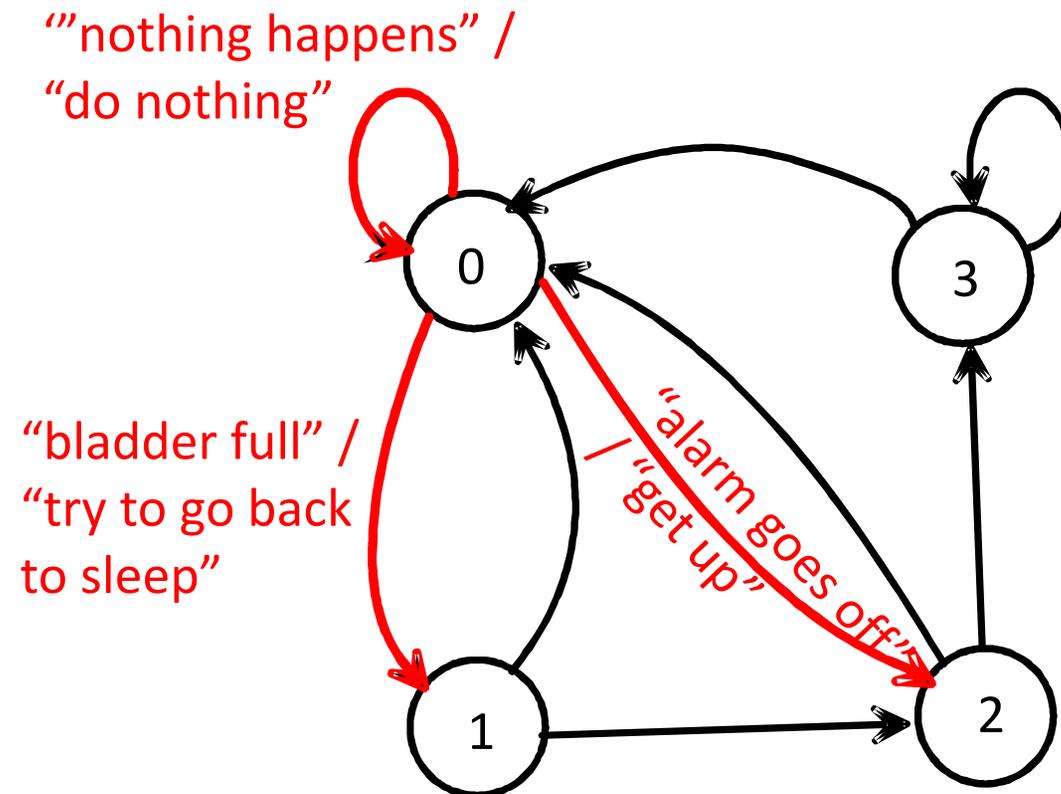
```
...
```

Finite State Machines (FSM)

Finite State Machine (FSM)



Finite State Machine (FSM)

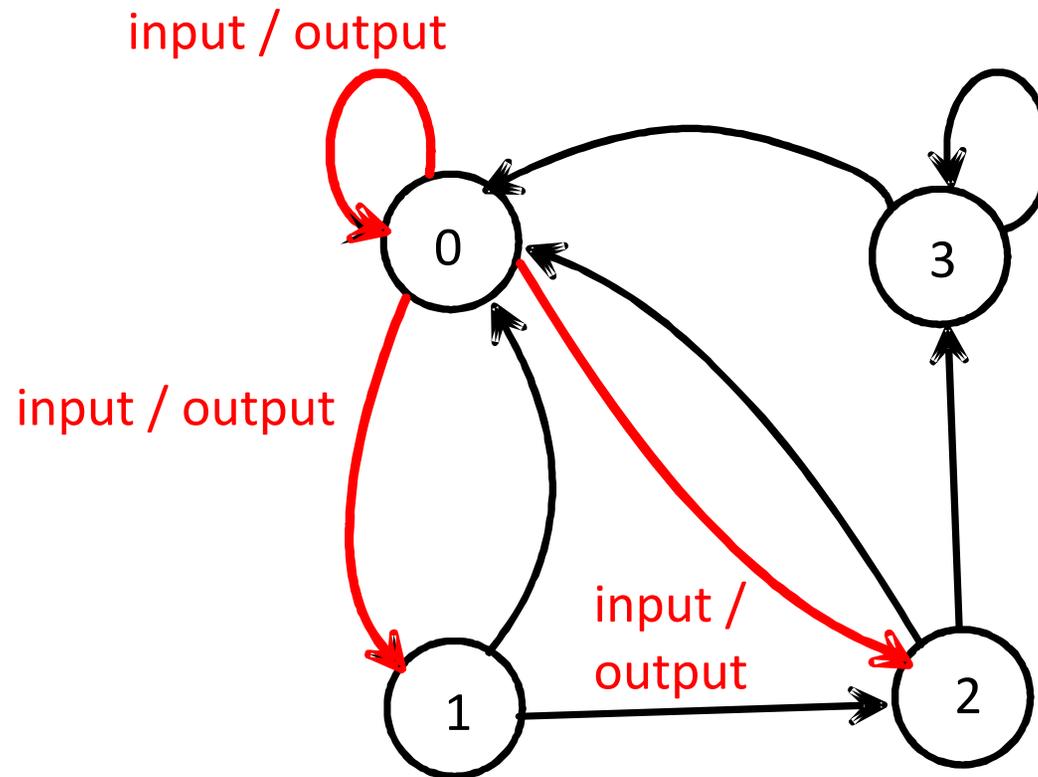


State 0: “in bed, asleep”

State 1: “in bed, awake”

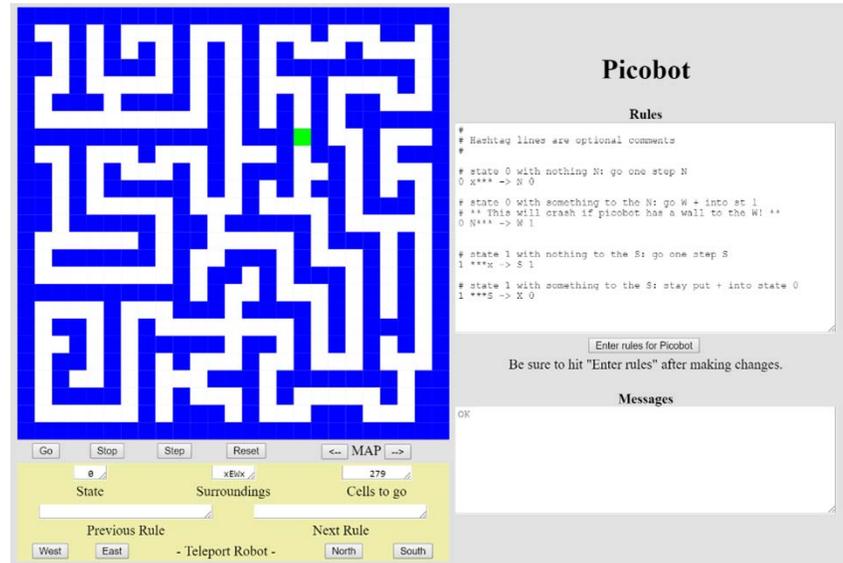
State 2: “in the bedroom, out of bed”

Finite State Machine (FSM)



state input \longrightarrow output new state

Picobot



Picobot

Rules

```
# Hashiq lines are optional comments
#
# state 0 with nothing N: go one step N
0 x*** -> N 0
# state 0 with something to the N: go W + into et 1
# ** This will crash if picobot has a wall to the W! **
0 N*** -> W 1
# state 1 with nothing to the S: go one step S
1 ***x -> S 1
# state 1 with something to the S: stay put + into state 0
1 ***S -> X 0
```

Enter rules for Picobot.
Be sure to hit "Enter rules" after making changes.

Messages

OK

Go Stop Step Reset <- MAP ->

State: 0 Surroundings: xxWx Cells to go: 279

Previous Rule: Next Rule

West East - Teleport Robot - North South

state

surroundings

direction

new state

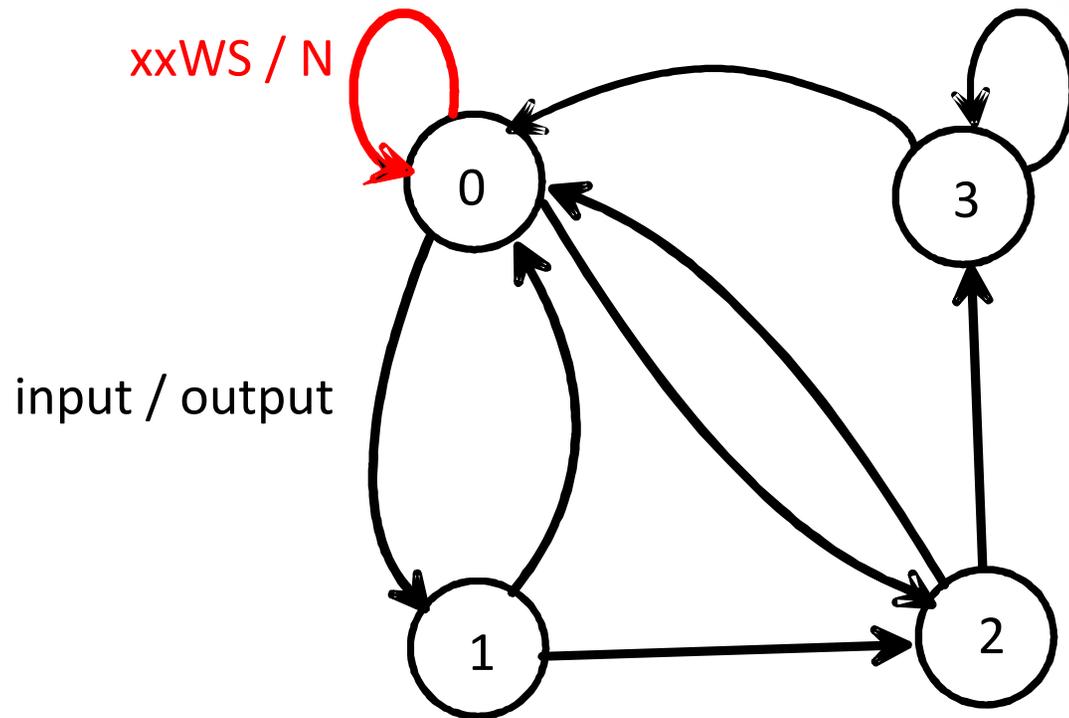
0

xxWS



N

0



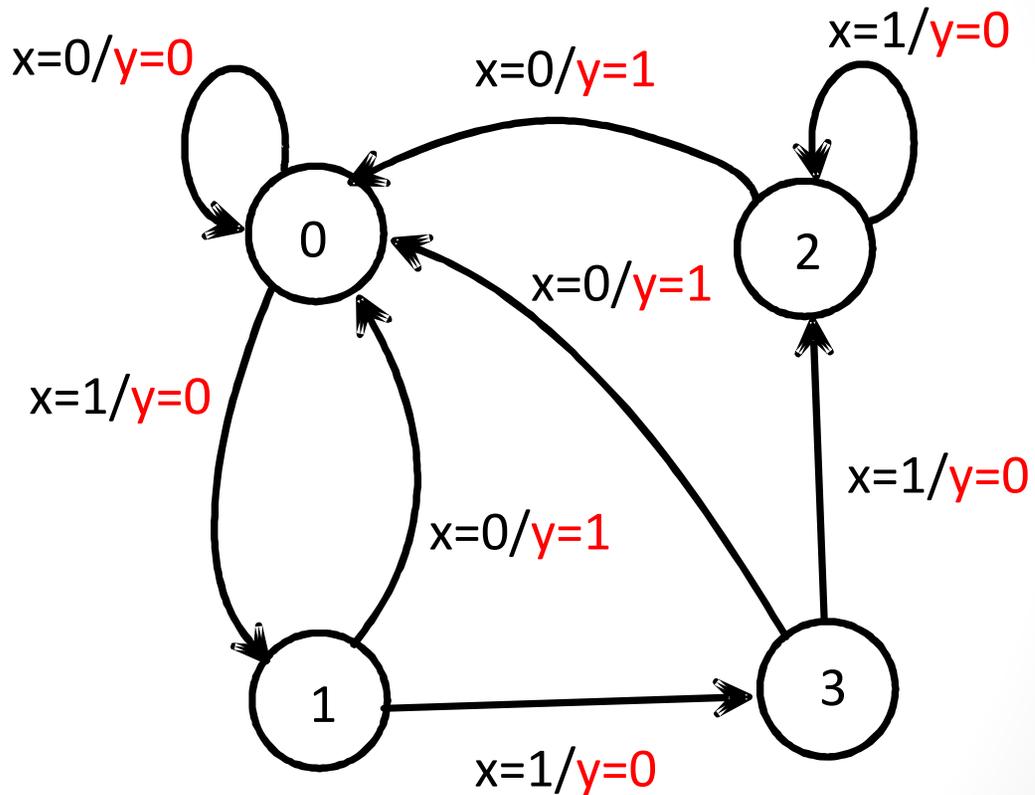
state	surroundings	direction	new state
0	xxWS	→	0

Finite State Machine (FSM)

State: s

Input: x

Output: y



Mealy FSM

```

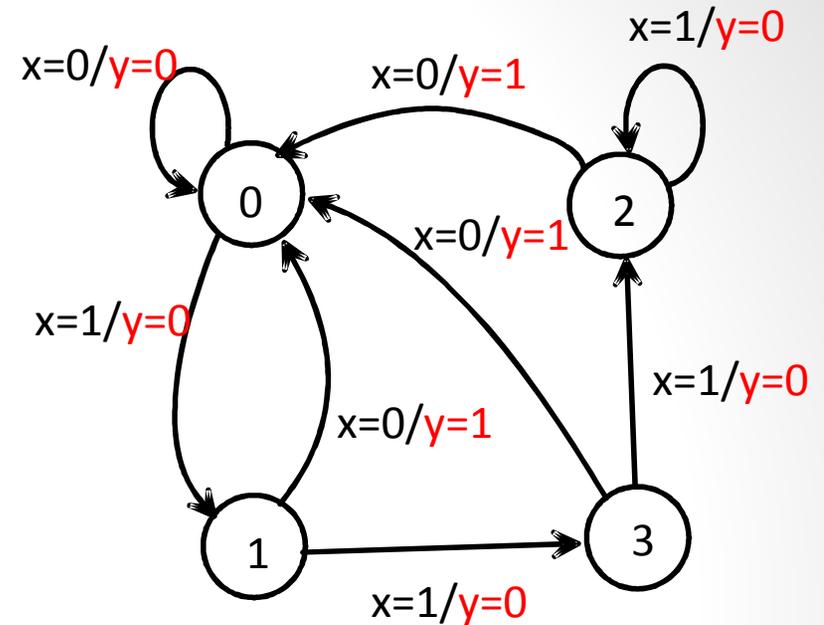
if (state == 0):
    if (x == 0):
        y = 0
        next_state = 0
    elif (x == 1):
        y = 0
        next_state = 1

elif (state == 1):
    if (x == 0):
        y = 1
        next_state = 0
    elif (x == 1):
        y = 0
        next_state = 3

```

...

```
state = next_state
```



Mealy FSM

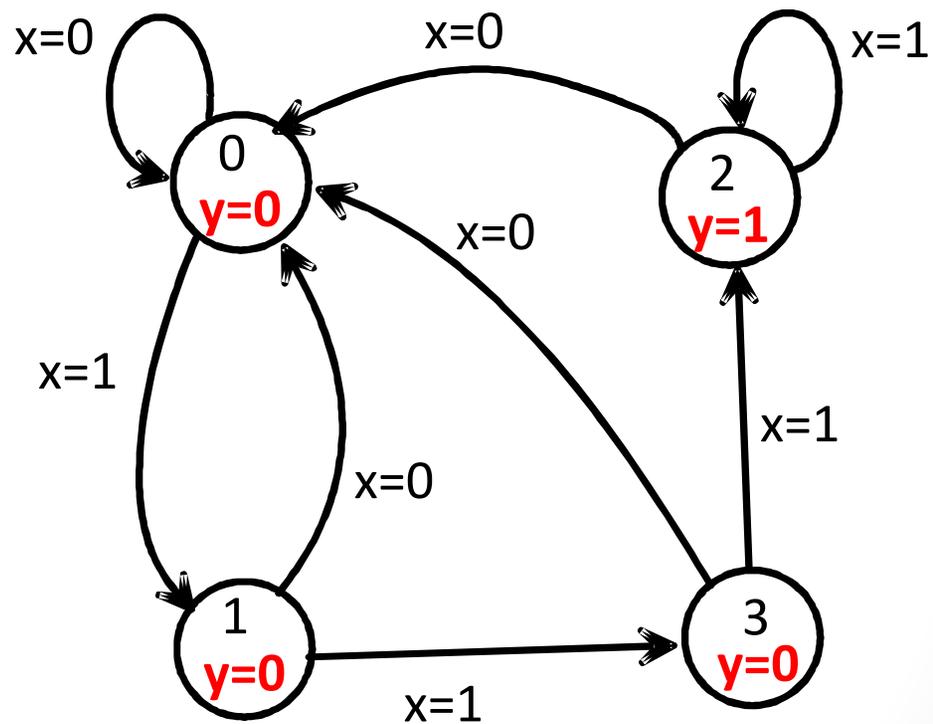
Finite State Machine (FSM)

State: s

Input: x

Output: y

Moore FSM

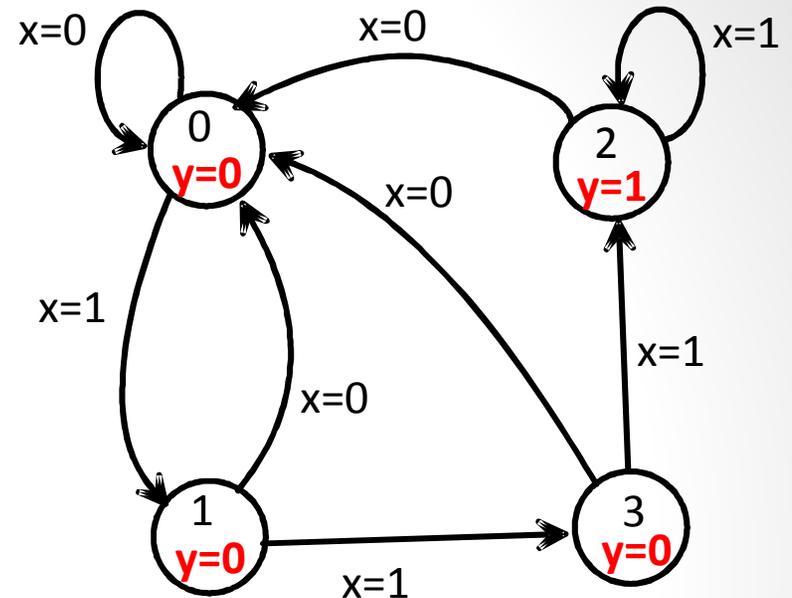


```
if (state == 0):
    y = 0
    if (x == 0):
        next_state = 0
    elif (x == 1):
        next_state = 1

elif (state == 1):
    y = 0
    if (x == 0):
        next_state = 0
    elif (x == 1):
        next_state = 3

...

state = next_state
```



Moore FSM