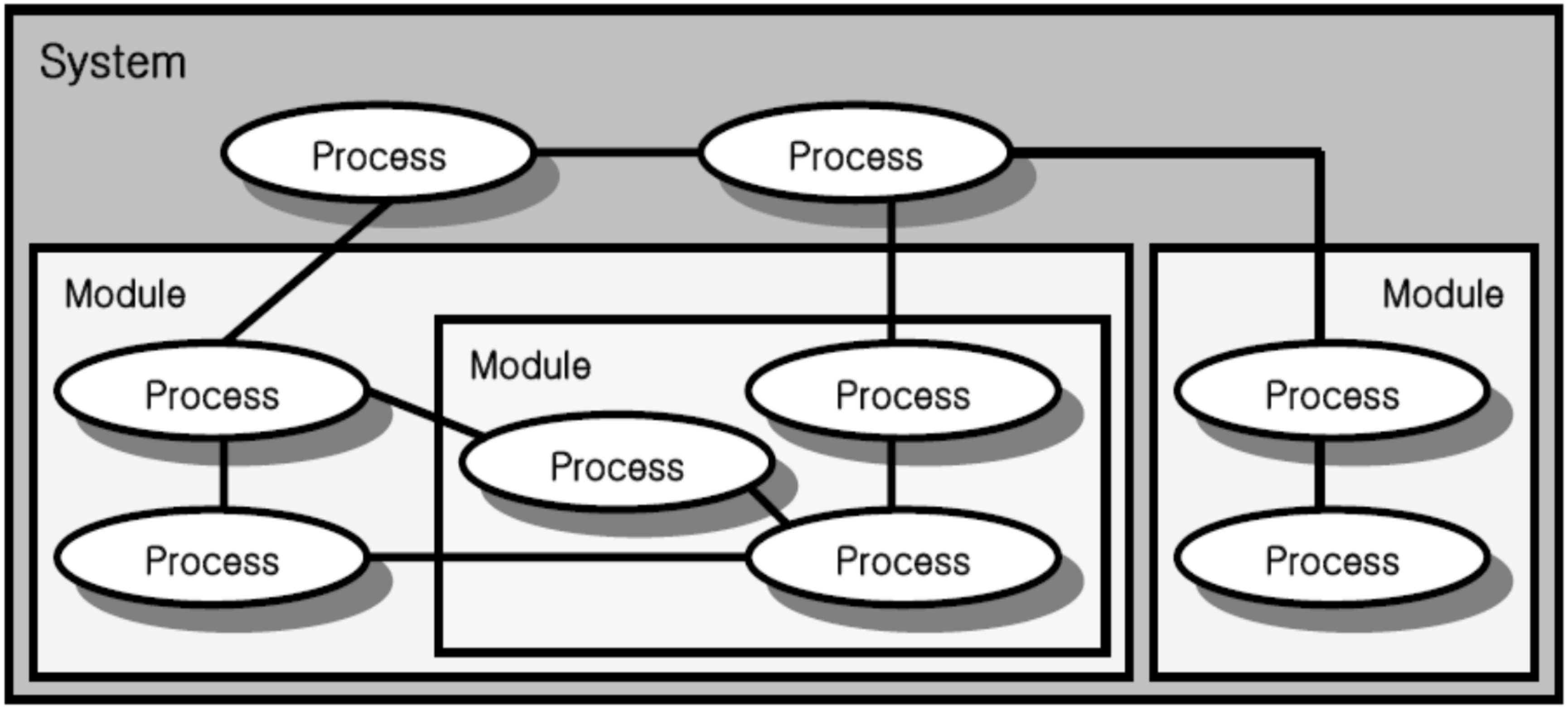
Turun yliopisto
University of Turku

HDL Based Design
ME620138


System
Process
Process
Module
Module
Module
Process
Process
Process
Process
Process
Process
Process

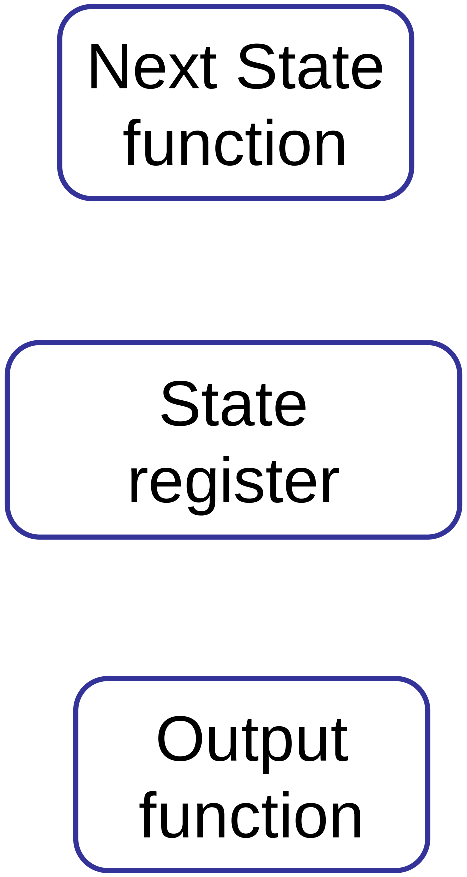
FINITE STATE MACHINES

Mealy and Moore

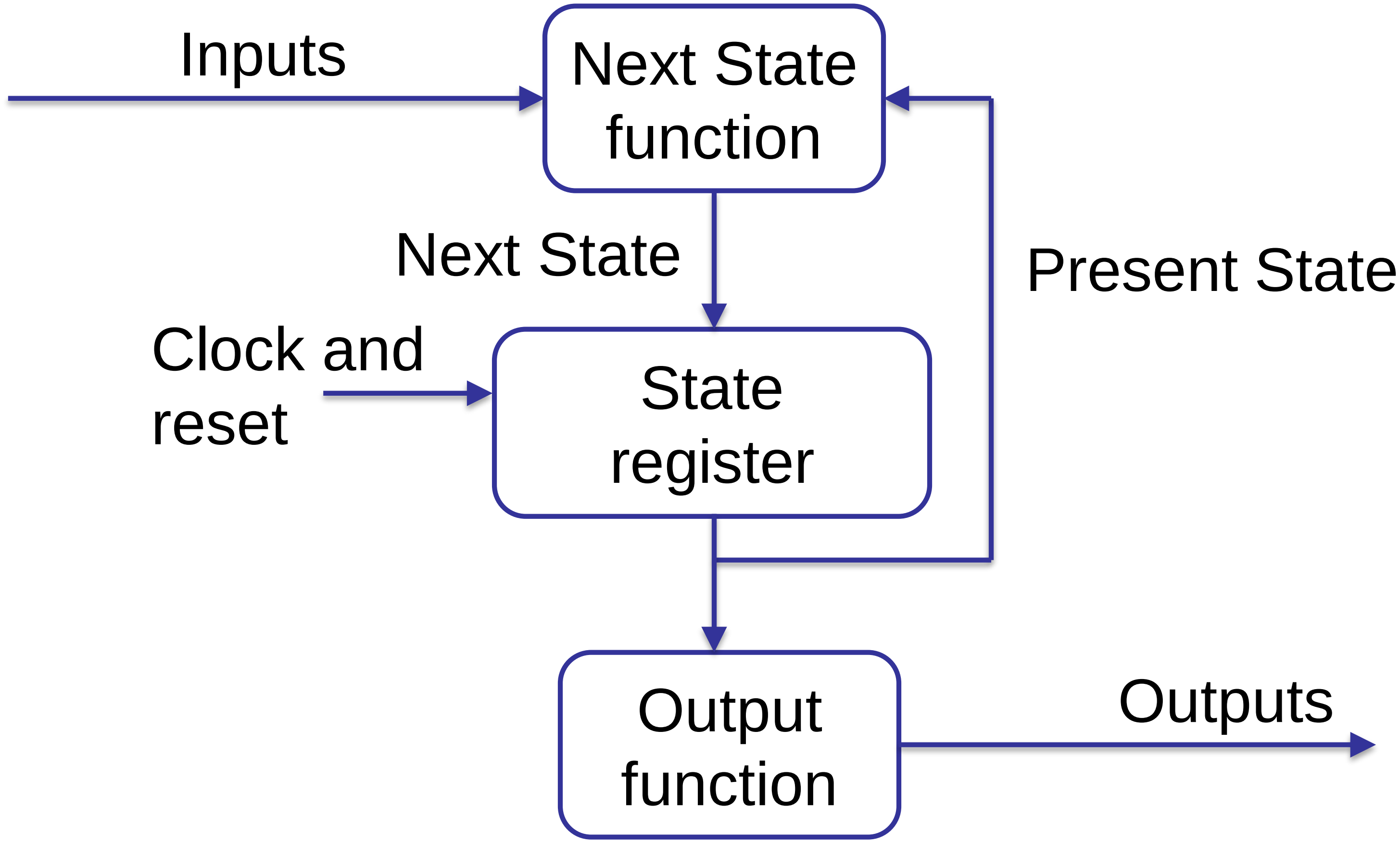
- Moore machine’s outputs are a function of the **present state only**
- Mealy machine’s outputs are a function of the **present state and present inputs**

Turu
Univ

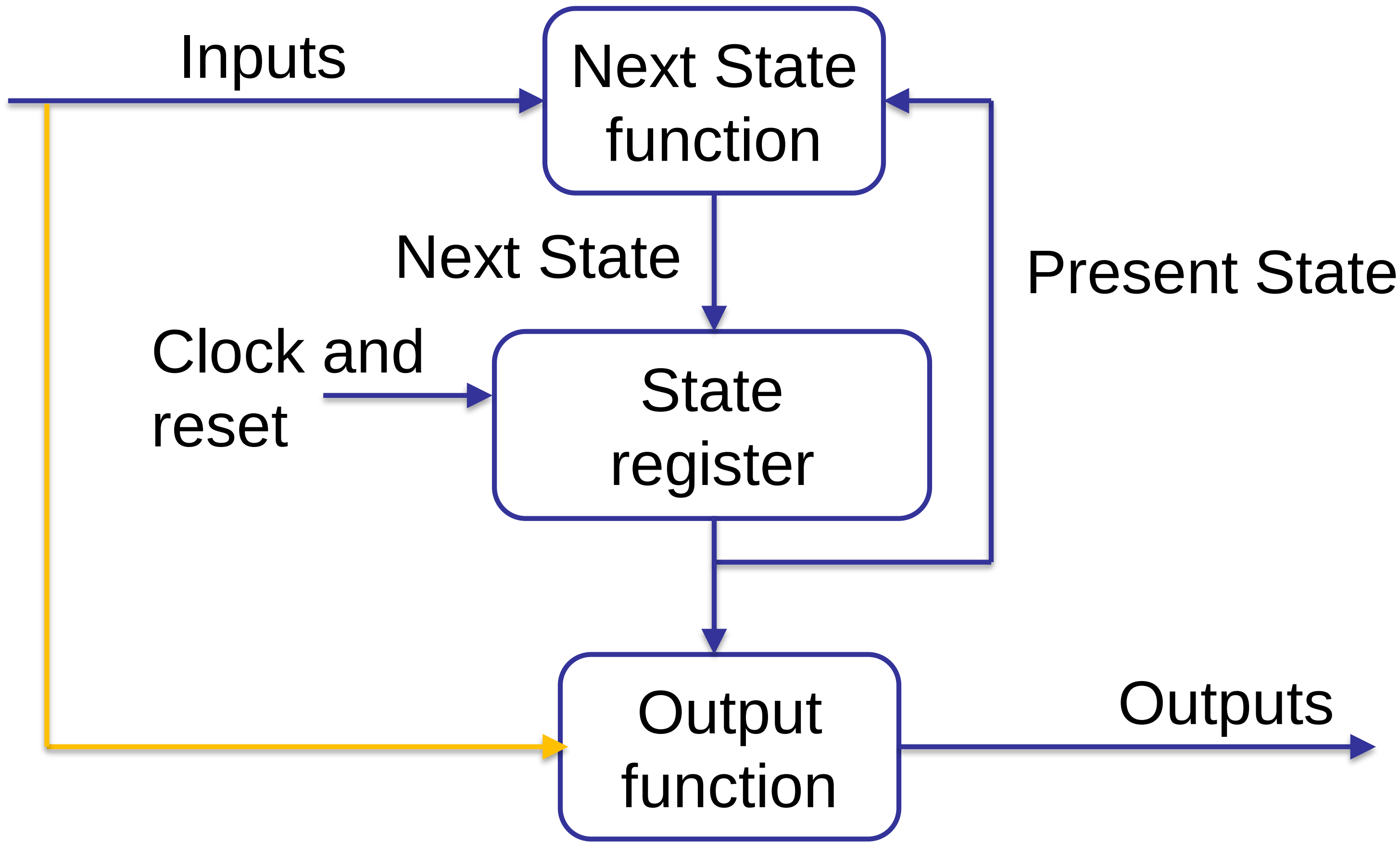
Finite State Machine



Moore Machine

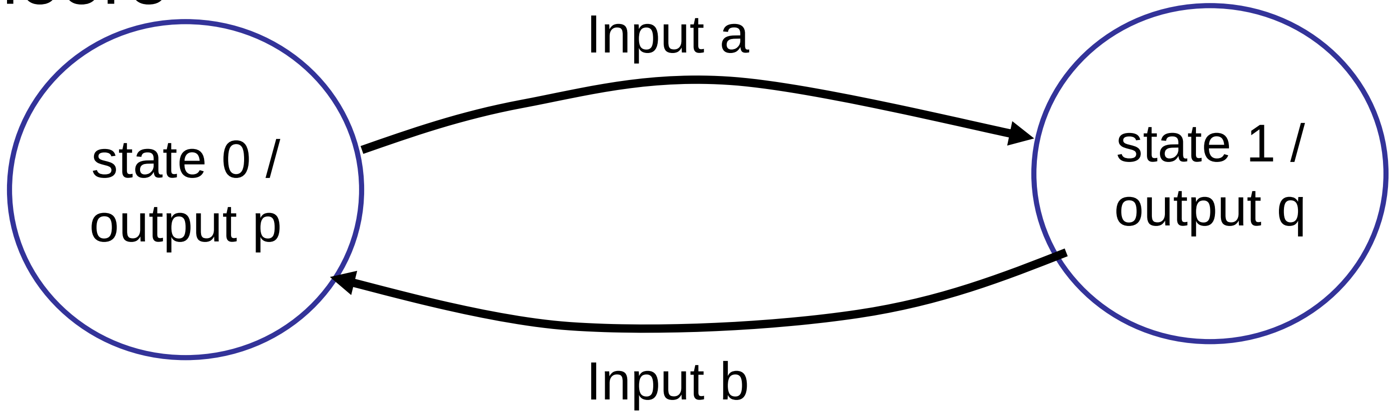


Mealy Machine

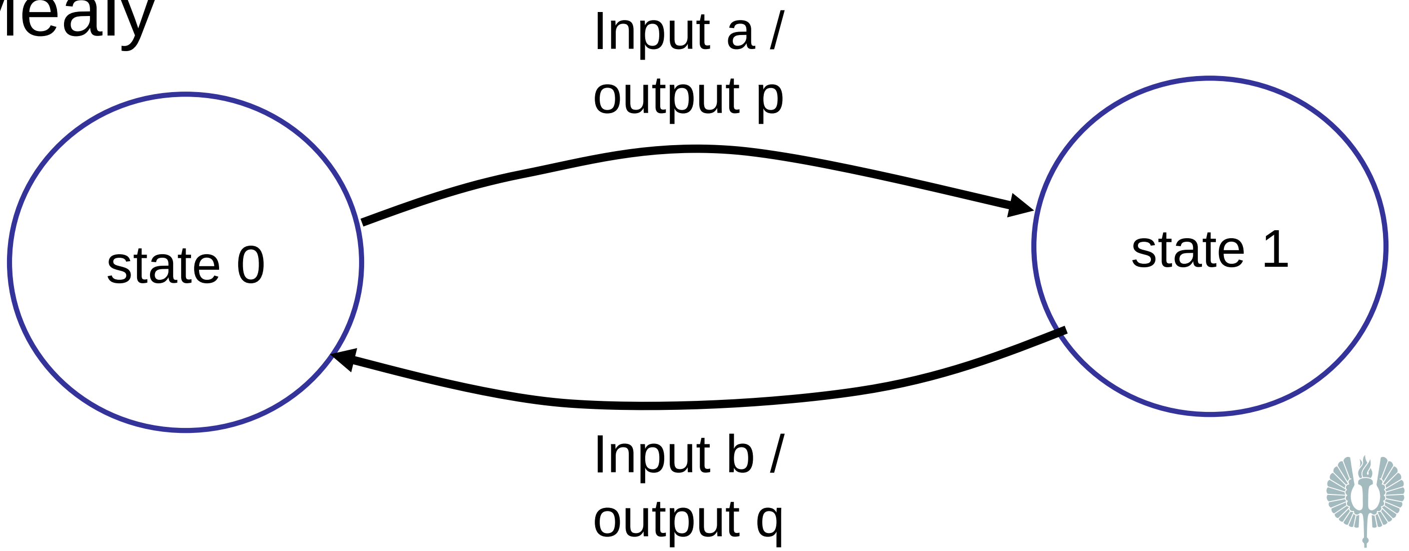


State diagrams offers an abstract graphical representation of the operation of FSM

- Moore

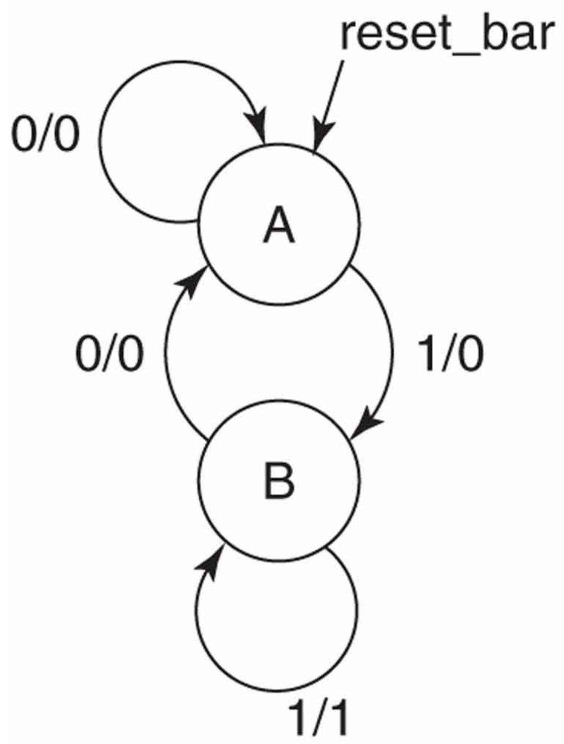


- Mealy

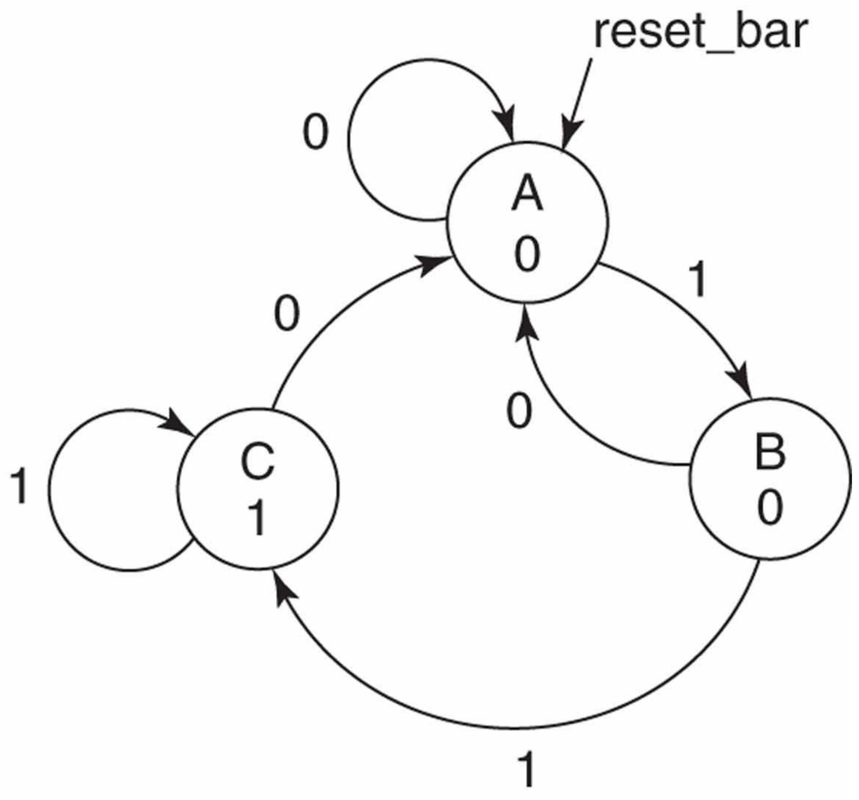


Turun yliopisto
University of Turku

Mealy and Moore

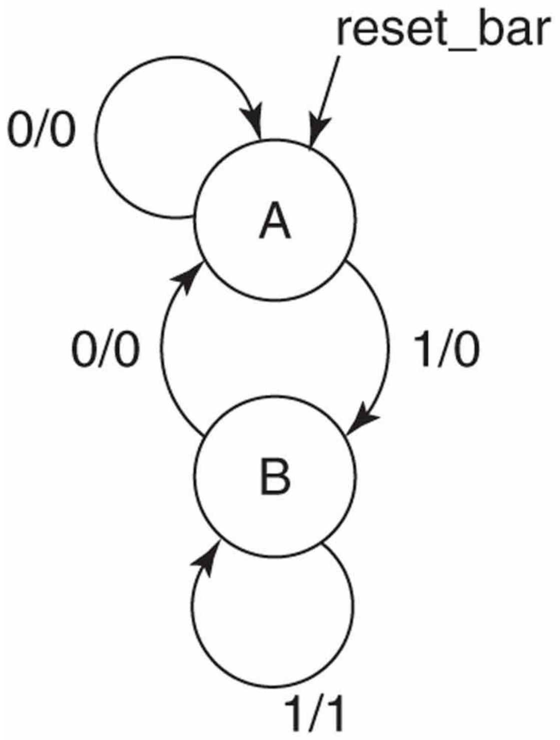


Mealy machine



Moore machine

Mealy and Moore



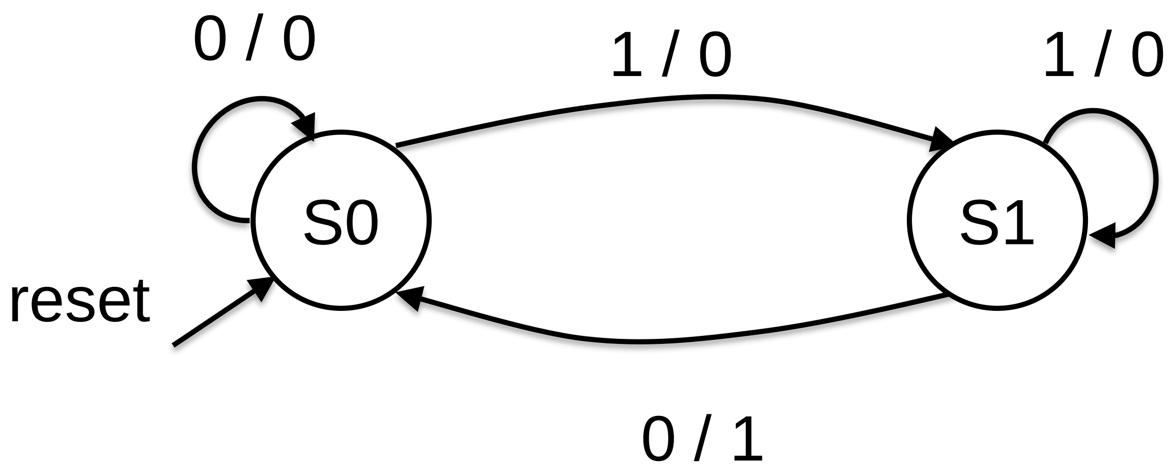
Present State	Input	Next State	Output
A	0	A	0
A	1	B	0
B	0	A	0
B	1	B	1

Mealy machine

Turun yliopisto
University of Turku

Mealy Recognising Sequence "10"

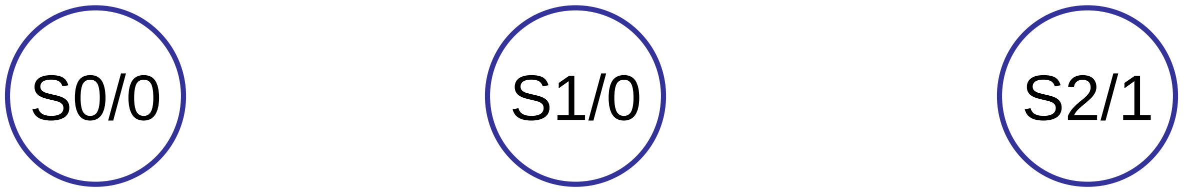
Mealy Recognising Sequence “10”



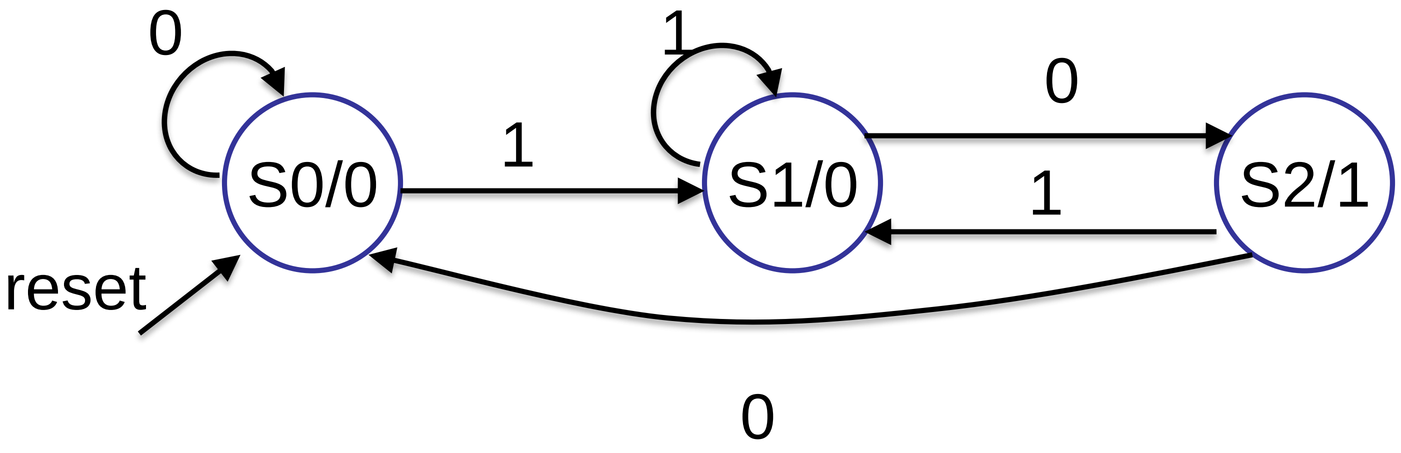
S0:
Nothing
observed

S1:
“1”
observed

Moore Recognising Sequence “10”



Moore Recognising Sequence “10”

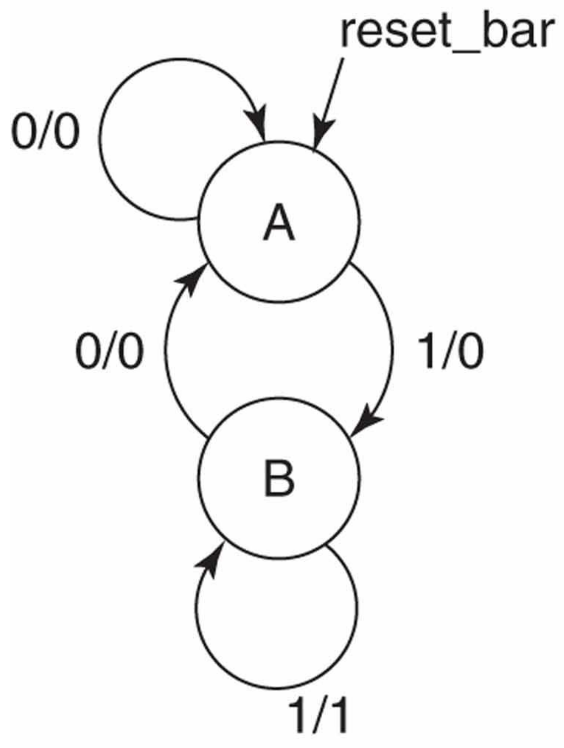


S0:
Nothing
observed

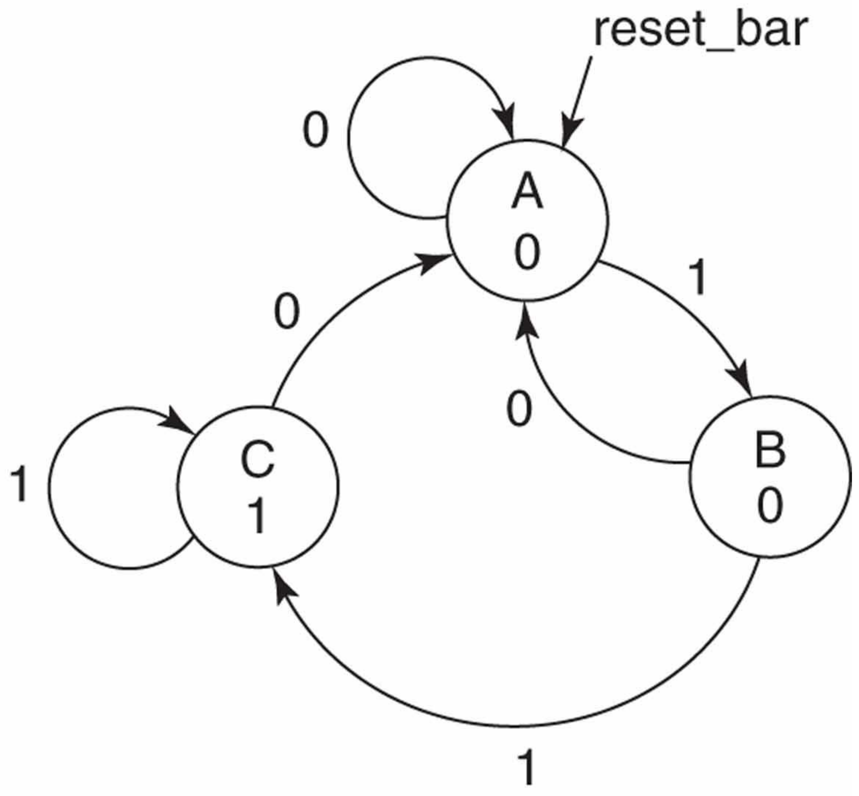
S1:
“1”
observed

S2:
“10”
observed

Mealy and Moore



Mealy machine



Moore machine

Mealy: output <= ‘1’ **when** (state = B **and** input = ‘1’) **else** ‘0’;

Moore: output <= ‘1’ **when** state = C **else** ‘0’;

VHDL outline for Moore Machine

```
type statemachine is (S0, S1, … , SX);
signal state: statemachine;

Moore: process (clock, reset)
begin
        if (reset = ‘1’) then                   state <= S0;
        elsif (clock = ‘1’ and clock’event) then
                case state is
                when S0 =>
                        if input = ‘1’ then     state <= S1;
                        else                    state <= S0;
                        end if;
                when S1 =>
                . . .
                end case;
        end if;
end process;

output <= ‘1’ when state = SX else ‘0’;
```

VHDL outline for Mealy Machine

```
type statemachine is (S0, S1, … ,  SX);
signal state: statemachine;

Mealy: process(clock, reset)
begin
        if (reset = ‘1’) then                   state <= S0;
        elsif (clock = ‘1’ and clock’event) then
                case state is
                when S0 =>
                        if input = ‘1’ then     state <= S1;
                        else                    state <= S0;
                        end if;
                when S1 =>
                . . .
                end case;
        end if;
end process;

output <= ‘1’ when (state = SX and input = ‘0|1’) else ‘0’;
```

Mealy vs Moore

- Can be functionaly equivalent
- Mealy usually requires smaller number of states
- Mealy reacts one clock cycle sooner than Moore
- Moore does not have a combinational path from input to output

Read Ex4

Start doing it

Turun yliopisto
University of Turku