EE214: Implementation to Finite State Machines

Madhav Desai

March 15, 2019

A sequential system

- Sequences: $\{x(k)\}$, $\{y(k)\}$ for $k = 0, 1, 2, \ldots$.
- Sequential system: a map from sequences to sequences.

$$y(k+1) \;=\; y(k) \oplus x(k),\;\; k \geq 0,\;\; y(0) = y_0.$$

- Causal sequential system: a map $F : X \to Y$ such that is $\mathbf{x} = \{x(k)\} \in X$, and $\{y(k)\} = F(\mathbf{x})$, then $y(k)$ is determined only by $x(0), x(1), \ldots x(k)$.
- Finite memory sequential system.

A Mealy machine

$$M = (\Sigma, \Lambda, Q, \delta, \lambda)$$

where

- Σ is a finite set of input symbols.
- Λ is a finite set of output symbols.
- Q is a finite set of states.
- $\delta : \Sigma \times Q \longrightarrow Q$ is a next-state function.
- $\lambda : \Sigma \times Q \longrightarrow \Lambda$ is the outptut function.

A Mealy machine: relations

$$
\begin{aligned}
q(k+1) &= \delta(x(k), q(k)) \\
y(k) &= \lambda(x(k), q(k))
\end{aligned}
$$

A Mealy machine: an example

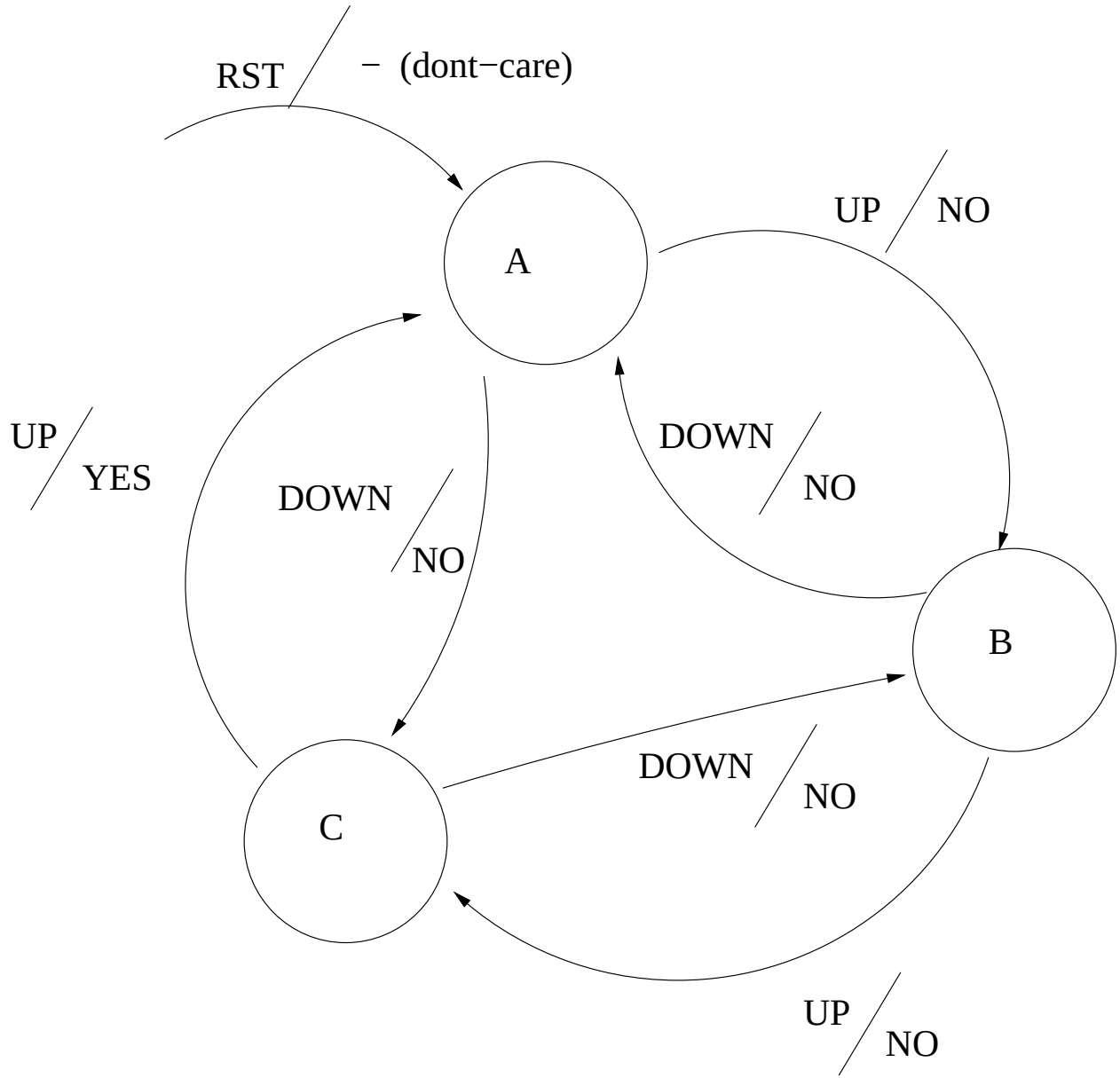


Figure: A simple FSM

The Mealy machine: implementation

We need to define the instants $k = 0, 1, 2,$. Use a periodic square wave.

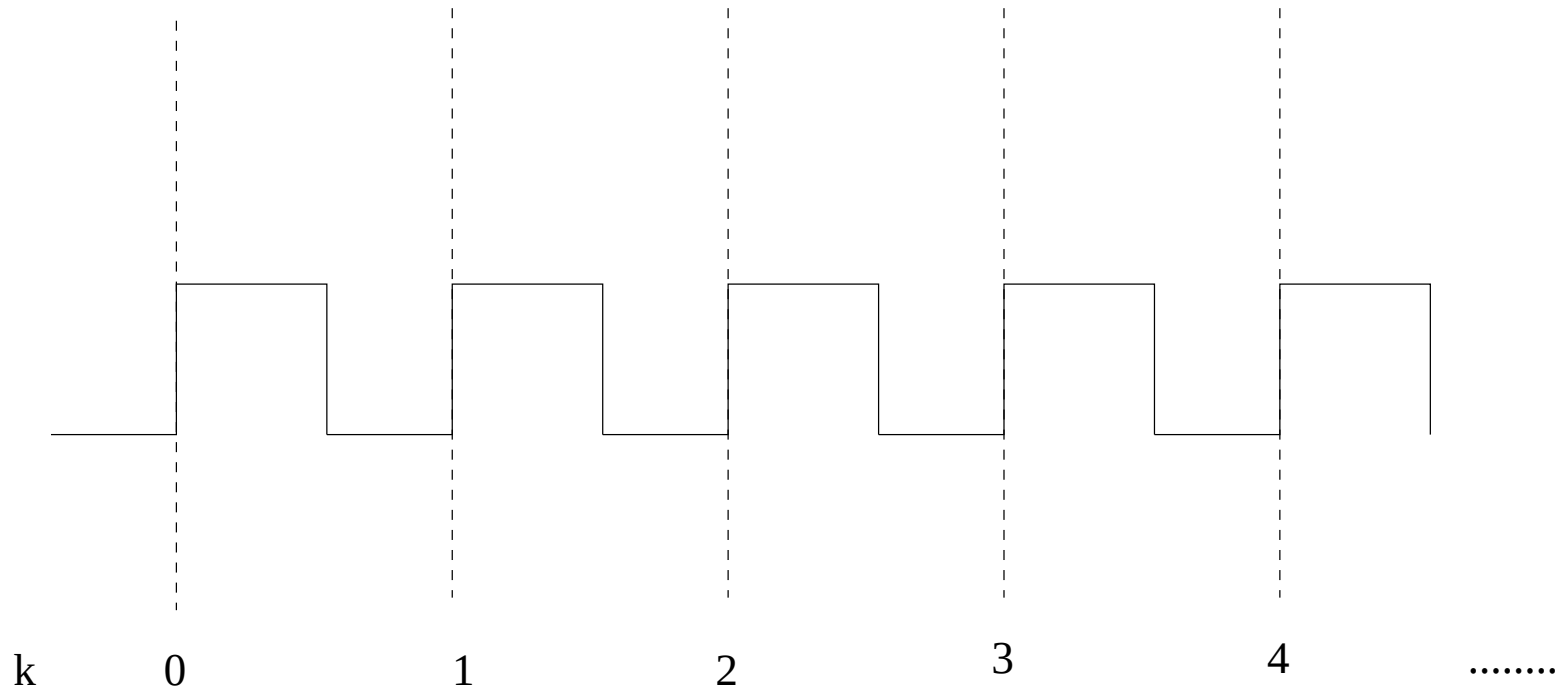


Figure: Clock

The Mealy machine: implementation

Symbols mapped to Boolean variables (encoding).

```
Input variables r u
Input coding  RST -> 1_, Up -> 01, Down -> 00

State variables  q1 q0
State coding   A -> 00, B -> 01, C -> 10

Output variables  y
Output coding   YES -> 1, NO -> 0
```

The Mealy machine: implementation

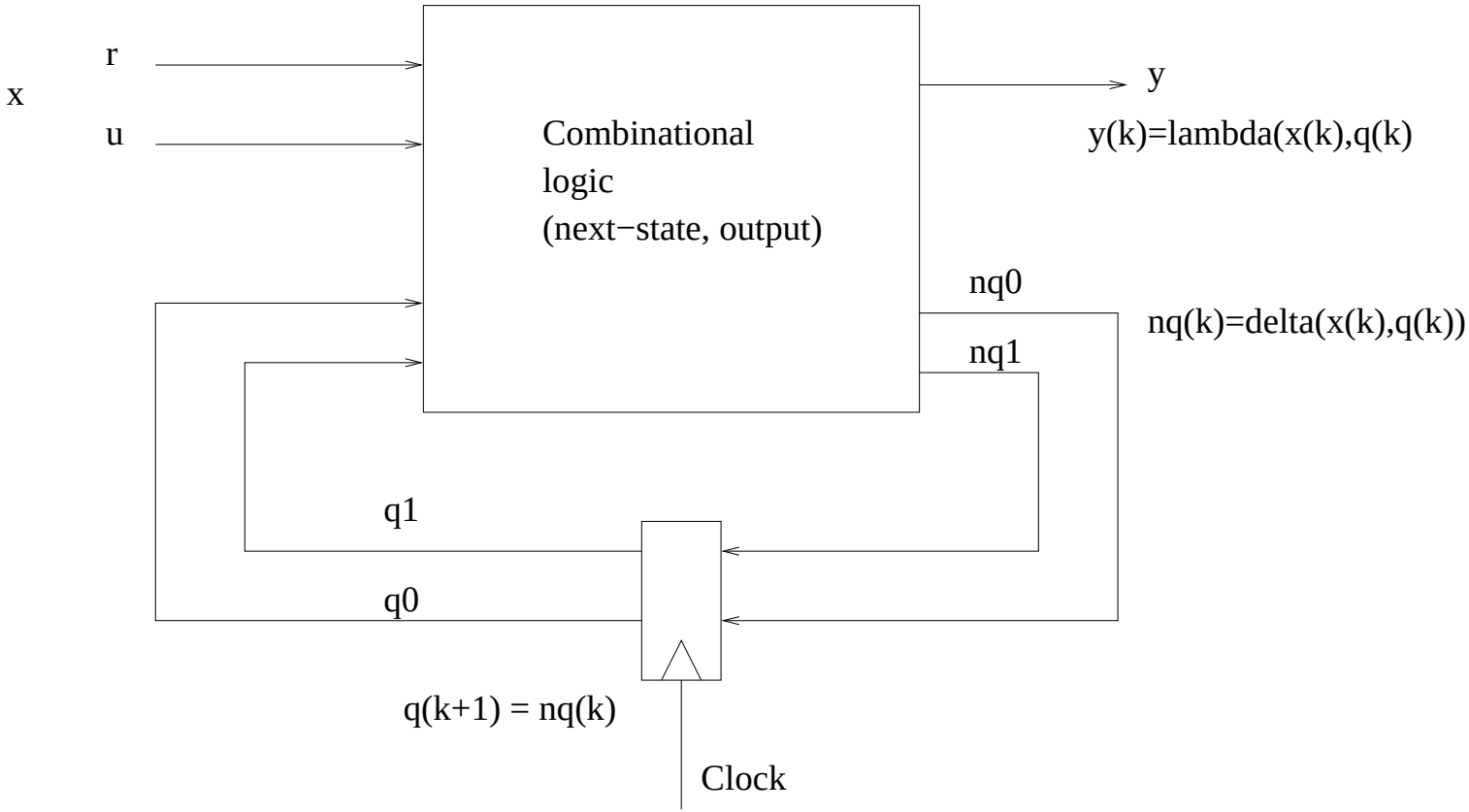
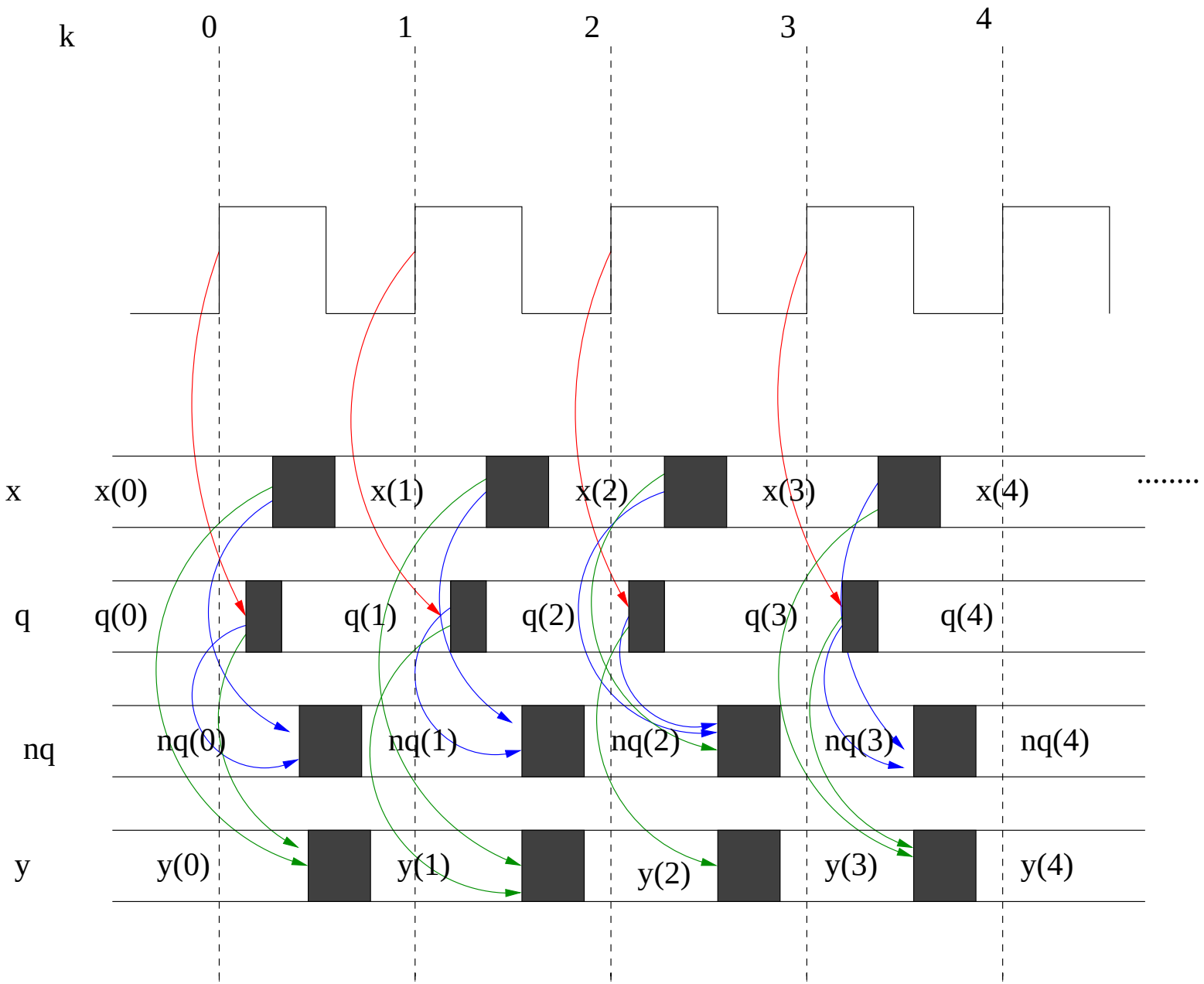


Figure: FSM Implementation

Vhdl Description of the FSM

- Minimize the next state and output functions and implement using logic gates. Connect up everything (using flip-flops).
- Implement using variables (FsmWithVars.vhdl).
- Implement using symbols (FsmWithSymbols.vhdl).

Timing Diagram



Trace file construction

inputs	outputs
clock=0 x(0)	y(0)
clock=1 x(0)	ignore
clock=0 x(1)	y(1)
clock=1 x(1)	ignore
clock=0 x(2)	y(2)
clock=1 x(2)	ignore
etc...	

From specification to implementation

- Input-symbols are (RST, a, b).
- Output-symbols are $(match, nomatch)$.
- The output sequence $y(k) = match$ if and only if the sequence $x(k-3)x(k-2)x(k-1)x(k)$ is either $abab$ or $baba$.

How to go about it?

- Identify the set of states.
- Build the state transition graph of the FSM.
- Implement the FSM using VHDL.
- Write a test trace file which ensures that every arc in the state transition graph is taken.