

Sequential Circuits

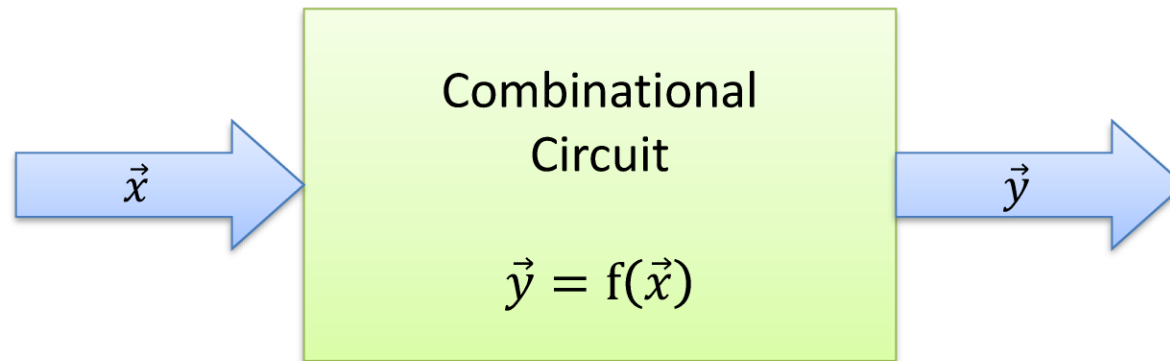
Networks and Embedded Software

Module 3.3.2

by Wolfgang Neff

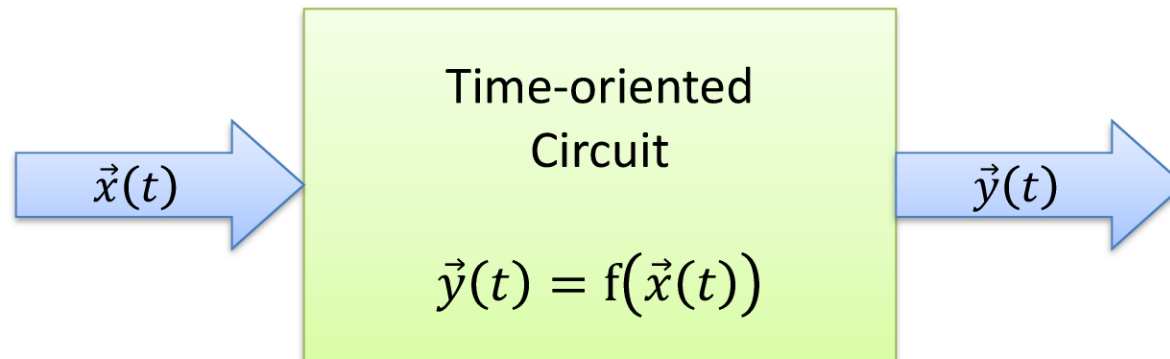
Combinational Circuits

- Output depends on current input, only
 - System has no memory
 - History is of no importance



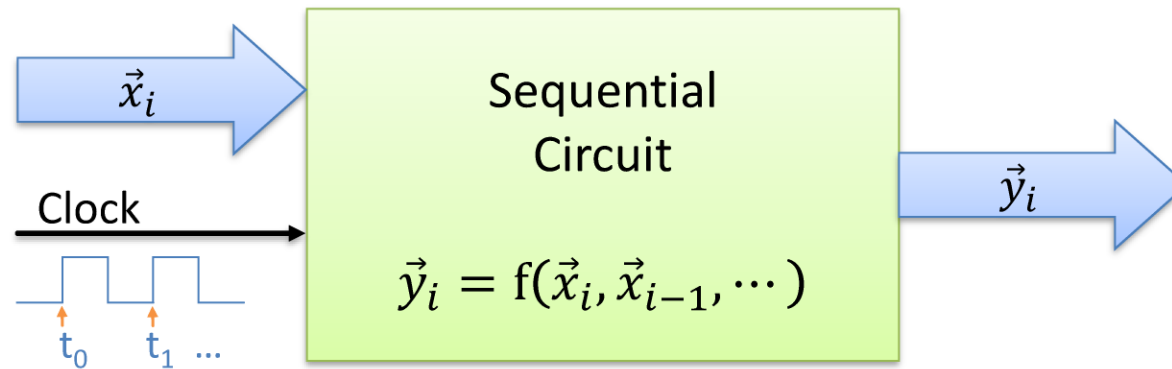
Sequential Circuits (1)

- Output depends on previous inputs
 - System has a memory
 - History is important



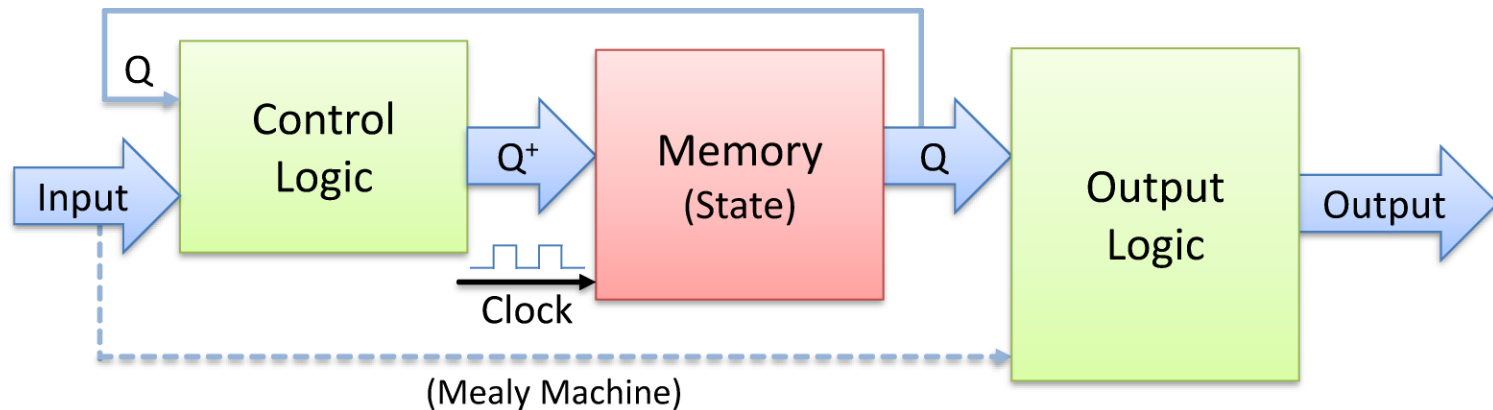
Sequential Circuits (2)

- Time is provided by a clock
 - Time gets discretized
 - Input becomes a sequence of data



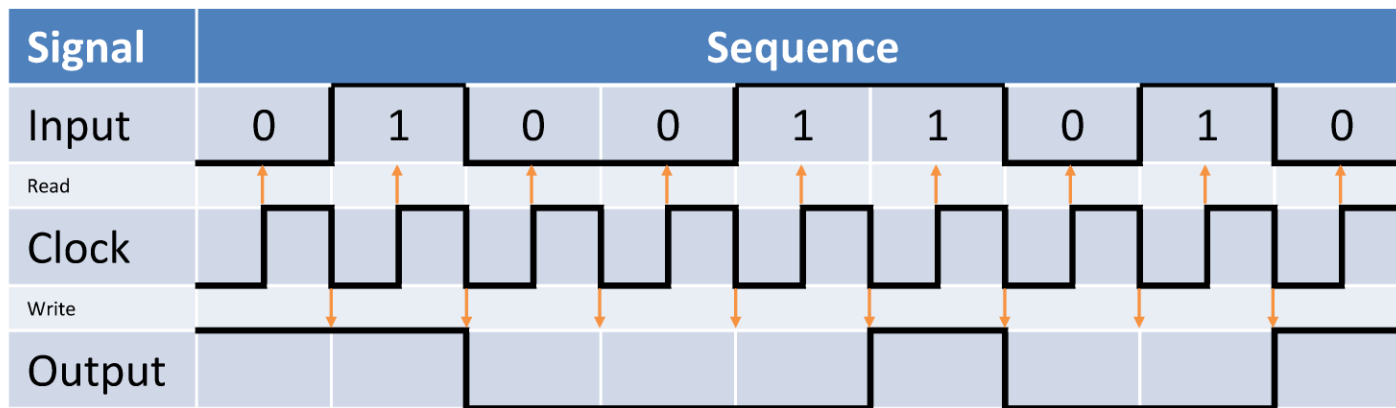
Sequential Circuits (3)

- Two representations are frequently used
 - Mealy Machine (dashed arrow allowed)
 - Moore Machine (dashed arrow not allowed)



Sequential Circuits (4)

- Timing
 - Control logic reads input on rising edge
 - Output logic writes output on falling edge



Pulse Diagram

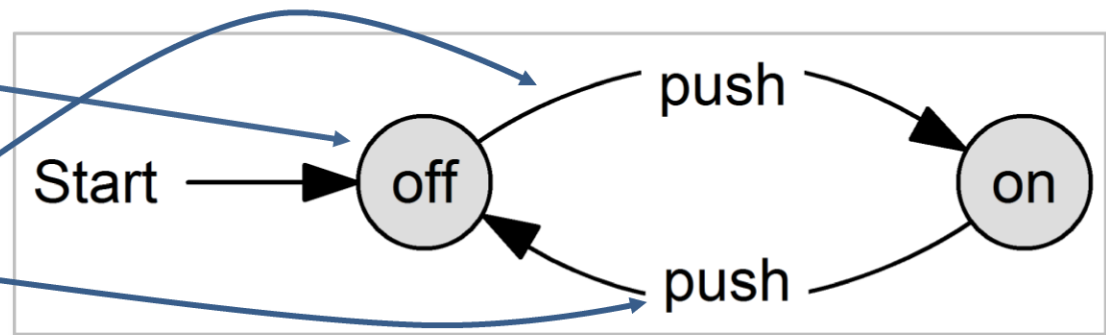
Sequential Circuits (5)

- States are handled by state machines (FSM)
- FSM are represented by state diagrams
- Finite state machines have

– States

– Transitions

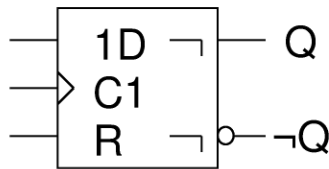
– Conditions



State Diagram of a Switch

Sequential Circuits (6)

- D flip-flops store the state



D	Q ⁺
0	0
1	1

- *1D*: Synchronous data line controlled by clock 1
- *C1*: First clock signal for the circuit
- *R*: Asynchronous reset line
- *Q*: Stored data

Sequential Circuits (7)

- Design process
 - Create state diagram
 - Number of states
 - Transitions and conditions
 - Encode states
 - One D flip-flops per bit
 - Design control logic
 - State/transition table
 - Design output logic

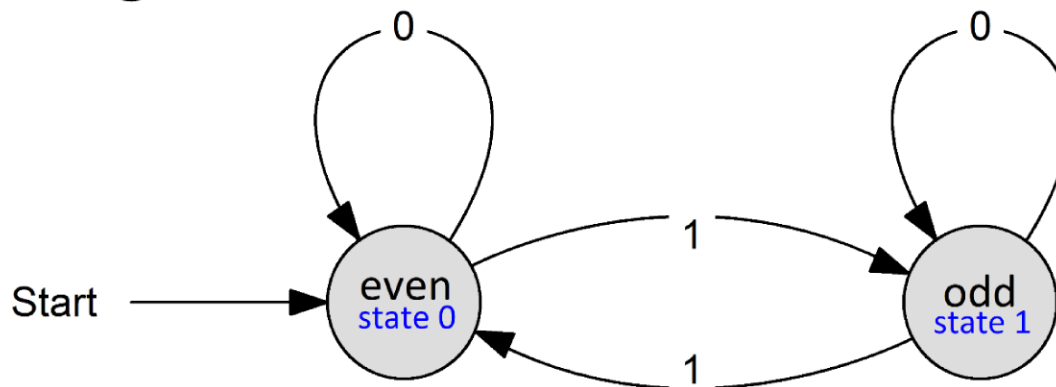
Sequential Circuits (8)

- Example I

- Assignment

Design a sequential circuit which outputs 1 if an even number of 1 was read.

- State diagram



Sequential Circuits (9)

- Example I (continued)
 - Number of states
 - Even, odd \rightarrow 2 states \rightarrow 1 bit \rightarrow 1 D flip-flop
 - State table (transitions and conditions)

Q	a	Q ⁺
Even (0)	0	Even (0)
Even (0)	1	Odd (1)
Odd (1)	0	Odd (1)
Odd (1)	1	Even (0)

a Input
 Q Current State
 Q^+ Next State

Sequential Circuits (10)

- Example I (continued)


- Encoding of states

- Even bit count: 0, 2, 4, 6, ...
 - Odd bit count: 1, 3, 5, 7, ...

State	Encoding
Even	0
Odd	1

- Output logic

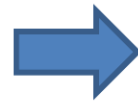
Q	y
0	1
1	0

 $y(Q) = \neg Q$

Sequential Circuits (11)

- Example I (continued)
 - Control logic

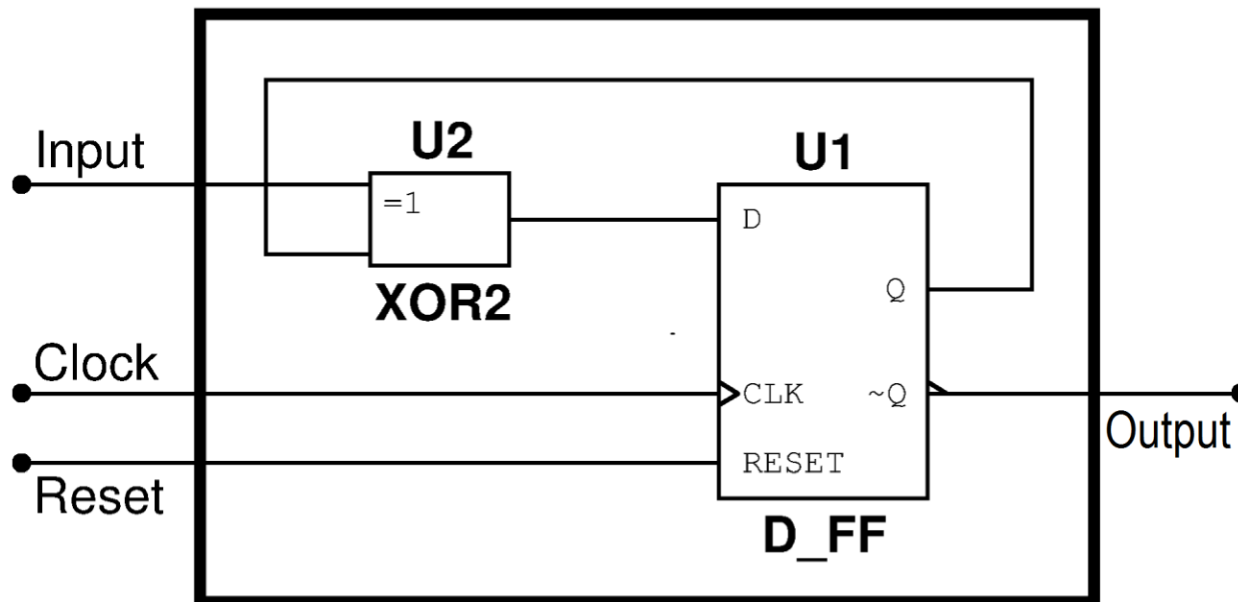
Q	a	Q ⁺
0	0	0
0	1	1
1	0	1
1	1	0



$$Q^+ = Q \oplus a$$

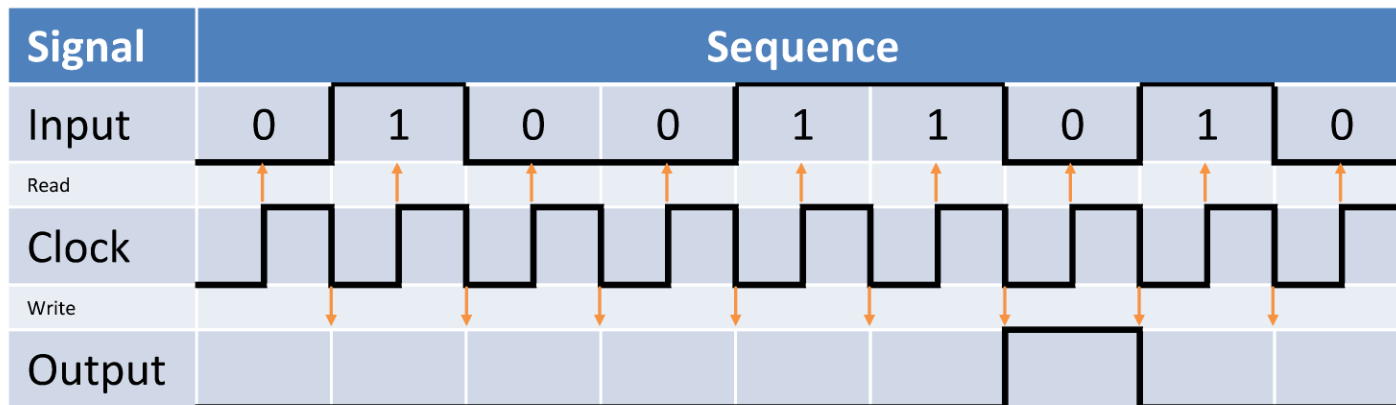
Sequential Circuits (12)

- Example I (finished)
 - The resulting sequential circuit



Sequential Circuits (13)

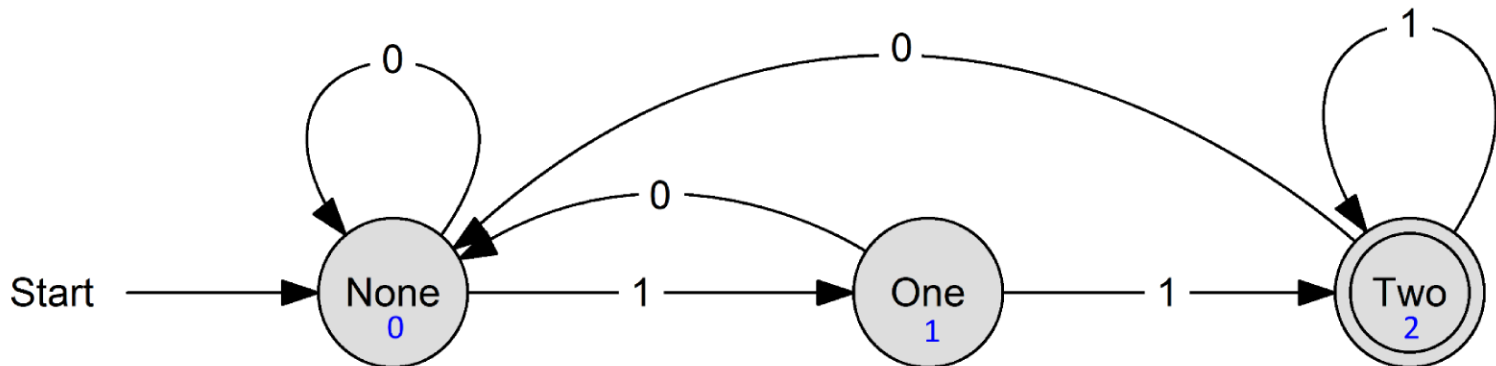
- Example II
 - Assignment
Find two subsequent *1* in a bit sequence



Pulse Diagram

Sequential Circuits (14)

- Example II (continued)
 - Number of states
 - None, one, two ones \rightarrow 3 states \rightarrow 2 bits
 - State diagram



Sequential Circuits (15)

- Example II (continued)
 - Control logic

Q_1	Q_0	a	Q_1^+	Q_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	0
1	1	0	X	X
1	1	1	X	X



State	Encoding
None	$0 \rightarrow 00$
One	$1 \rightarrow 01$
Two	$2 \rightarrow 10$

$$Q_0^+ = a \wedge \neg Q_0 \wedge \neg Q_1$$

$$Q_1^+ = (a \wedge Q_1) \vee (a \wedge Q_0)$$

Sequential Circuits (16)

- Example II (finished)

– Output logic

Q_1	Q_0	y
0	0	0
0	1	0
1	0	1
1	1	X

$$y = Q_1$$

