

CpE358/CS381

**Switching Theory and
Logical Design**

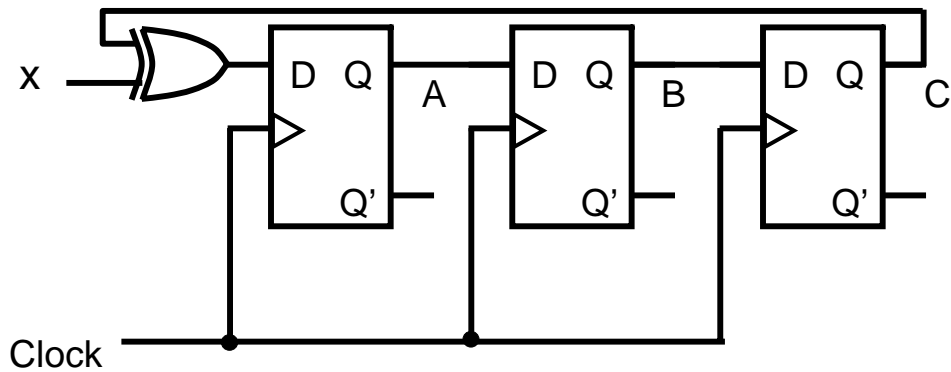
Class 8

Today

- Fundamental concepts of digital systems (Mano Chapter 1)
- Binary codes, number systems, and arithmetic (Ch 1)
- Boolean algebra (Ch 2)
- Simplification of switching equations (Ch 3)
- Digital device characteristics (e.g., TTL, CMOS)/design considerations (Ch 10)
- Combinatoric logical design including LSI implementation (Chapter 4)
- Flip-flops and state memory elements (Ch 5)
- **Sequential logic analysis and design** (Ch 5)
- Hazards, Races, and time related issues in digital design (Ch 9)
- Synchronous vs. asynchronous design (Ch 9)
- Counters, shift register circuits (Ch 6)
- Memory and Programmable logic (Ch 7)
- Minimization of sequential systems
- Introduction to Finite Automata

Typical Sequential Circuit

- Description by State Equations



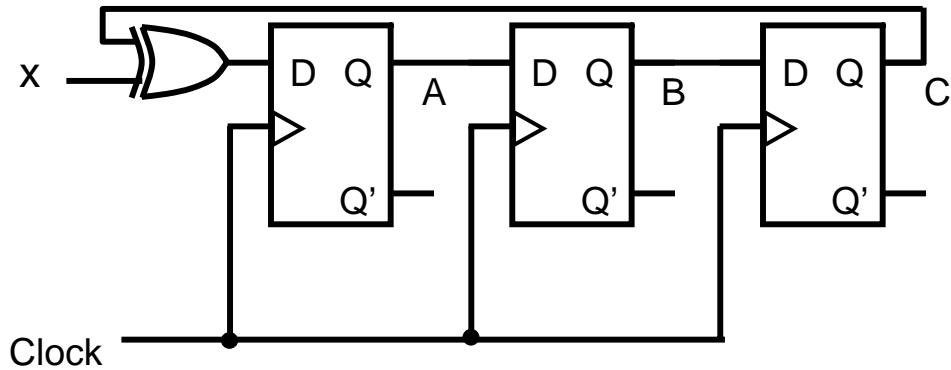
$$A(t+1) = x \oplus C(t)$$

$$B(t+1) = A(t)$$

$$C(t+1) = B(t)$$

Typical Sequential Circuit

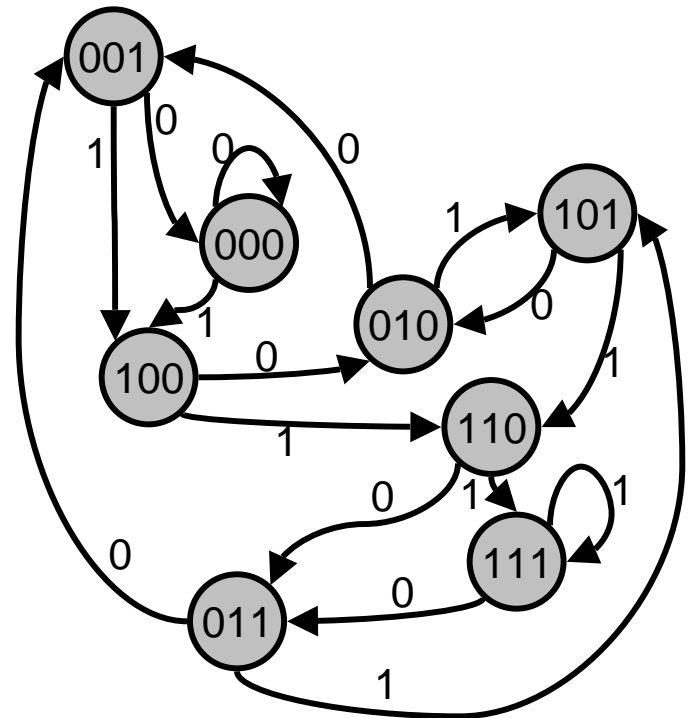
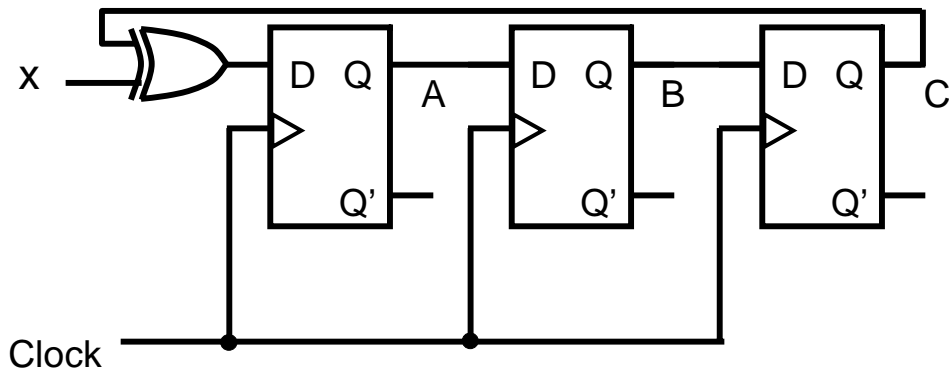
- Description by State Table



Present State			Input X	Next State			Output
A	B	C		A	B	C	
0	0	0	0		0	0	-
0	0	0	1		0	0	-
0	0	1	0		0	0	-
0	0	1	1		0	0	-
0	1	0	0		0	1	-
0	1	0	1		0	1	-
0	1	1	0		0	1	-
0	1	1	1		0	1	-
1	0	0	0		1	0	-
1	0	0	1		1	0	-
1	0	1	0		1	0	-
1	0	1	1		1	0	-
1	1	0	0		1	1	-
1	1	0	1		1	1	-
1	1	1	0		1	1	-
1	1	1	1		1	1	-

Typical Sequential Circuit

- Description by State Diagram

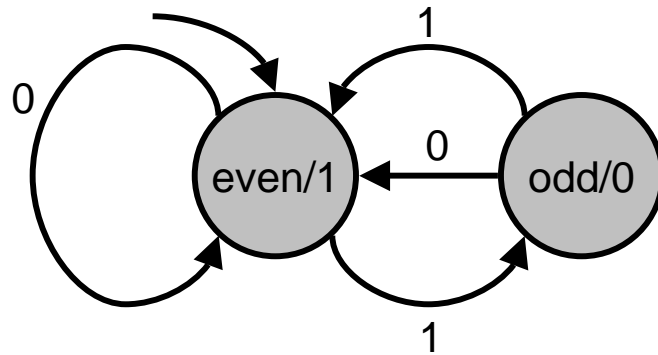


Sequential System Design

- Start with a description of the system operation:
 - Design a sequential circuit to output a 1 if there have been an even number of 1's received since the last 0. Output 0 otherwise

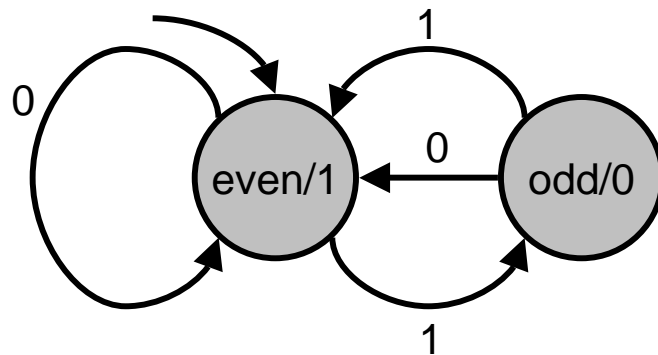
Sequential System Design

- Start with a description of the system operation:
 - Design a sequential circuit to output a 1 if there have been an even number of 1's received since the last 0. Output 0 otherwise
- (Generally) The next step is to draw the state diagram



Sequential System Design

- Start with a description of the system operation:
 - Design a sequential circuit to output a 1 if there have been an even number of 1's received since the last 0. Output 0 otherwise
- (Generally) The next step is to draw the state diagram



- See if any states can be eliminated.
- From the number of states, determine the minimum number of flip-flops needed. Also determine if there is a simple way to assign states that might use more flip-flops, but would make combinatorial logic simpler.
 - In this case, there are two states. $N_{ff} = \log_2(N_{states}) = \log_2(2) = 1$

Sequential System Design

- Write the state equations or state table (or both)

Present State	Input	Next State	Output
even	0	even	1
even	1	odd	0
odd	0	even	1
odd	1	even	1

Sequential System Design

- Write the state equations or state table (or both)

Present State	Input	Next State	Output
even	0	even	1
even	1	odd	0
odd	0	even	1
odd	1	even	1

- Number states in some logical fashion
 - Here, it makes sense to use the states 0 and 1, and let these values correspond to the output, so even=1 and odd=0

Sequential System Design

- Write the state equations or state table (or both)

Present State	Input	Next State	Output
even	0	even	1
even	1	odd	0
odd	0	even	1
odd	1	even	1

- Number states in some logical fashion
 - Here, it makes sense to use the states 0 and 1, and let these values correspond to the output, so even=1 and odd=0
- Assign flip-flops to state number to make decoding straightforward
 - There is only one flip-flop in this case, so the assignment is direct

Sequential System Design

- Write the state equations or state table (or both)

Present State	Input	Next State	Output
even	0	even	1
even	1	odd	0
odd	0	even	1
odd	1	even	1

- Number states in some logical fashion
 - Here, it makes sense to use the states 0 and 1, and let these values correspond to the output, so even=1 and odd=0
- Assign flip-flops to state number to make decoding straightforward
 - There is only one flip-flop in this case, so the assignment is direct
- Relabel state table with state assignments

Present State	Input	Next State	Output
1	0	1	1
1	1	0	0
0	0	1	1
0	1	1	1

Sequential System Design

- Decide on type of flip-flop
 - Let's try a D-type flip-flop

Present State	Input	Next State	Output
1	0	1	1
1	1	0	0
0	0	1	1
0	1	1	1

Sequential System Design

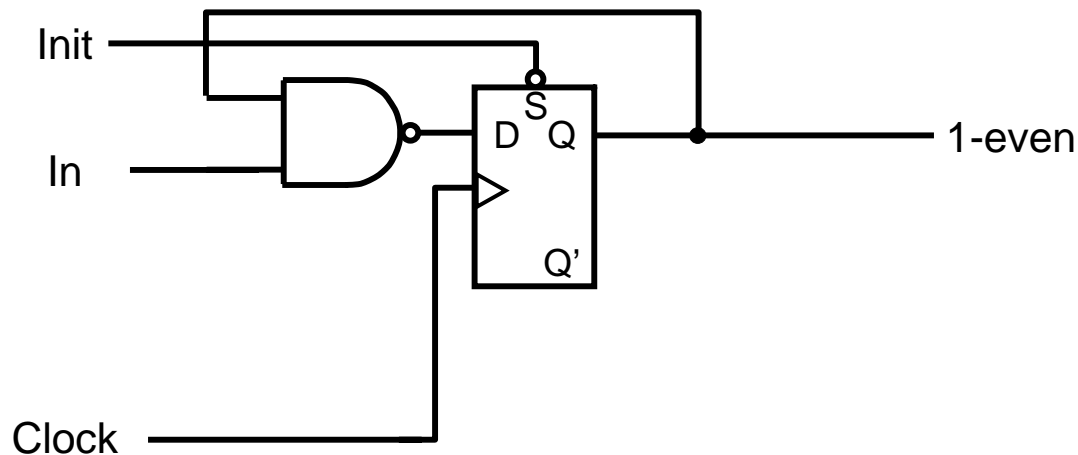
- Decide on type of flip-flop
 - Let's try a D-type flip-flop
- Derive and minimize switching equations for FF inputs
 - $D_{in} = (QI)'$
- And (as necessary) outputs

Present State	Input	Next State	Output
1	0	1	1
1	1	0	0
0	0	1	1
0	1	1	1

Sequential System Design

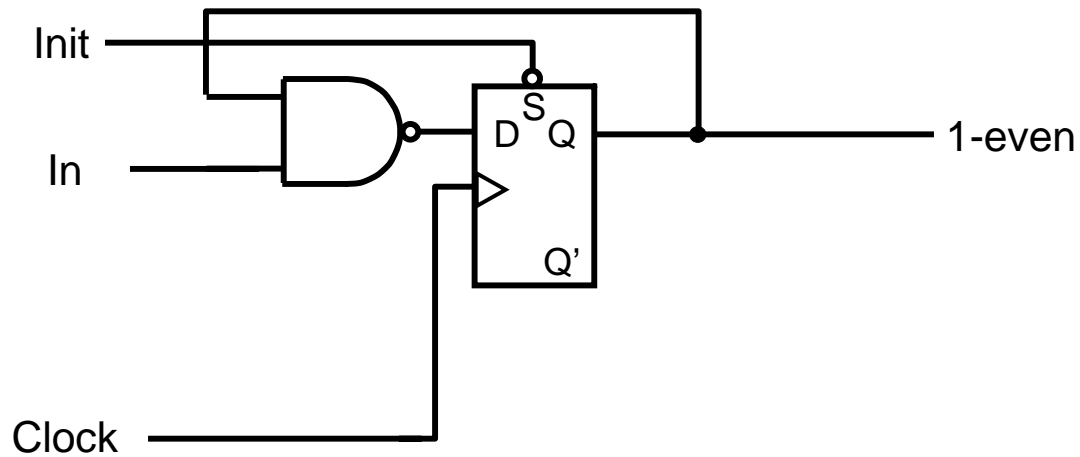
- Decide on type of flip-flop
 - Let's try a D-type flip-flop
- Derive and minimize switching equations for FF inputs
 - $D_{in} = (QI)'$
- And (as necessary) outputs

Present State	Input	Next State	Output
1	0	1	1
1	1	0	0
0	0	1	1
0	1	1	1



Sequential System Design

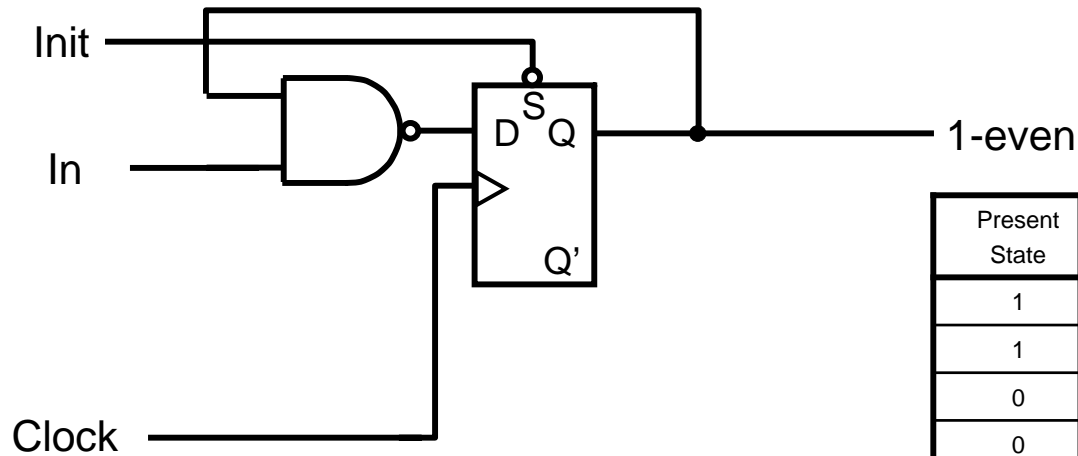
- Verify operation of design



Present State	Input	Next State	Output
1	0	1	1
1	1	0	0
0	0	1	1
0	1	1	1

Sequential System Design

- Verify operation of design

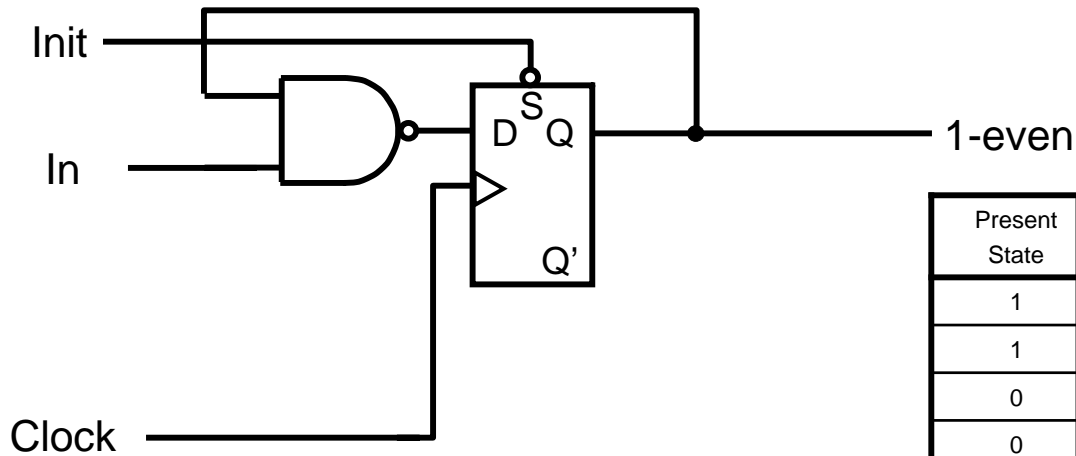


Present State	Input	Next State	Output
1	0	1	1
1	1	0	0
0	0	1	1
0	1	1	1

Present State	Input	(QI)	(QI)'	Next State	Output
1	0	0	1	1	1
1	1	1	0	0	0
0	0	0	1	1	1
0	1	0	1	1	1

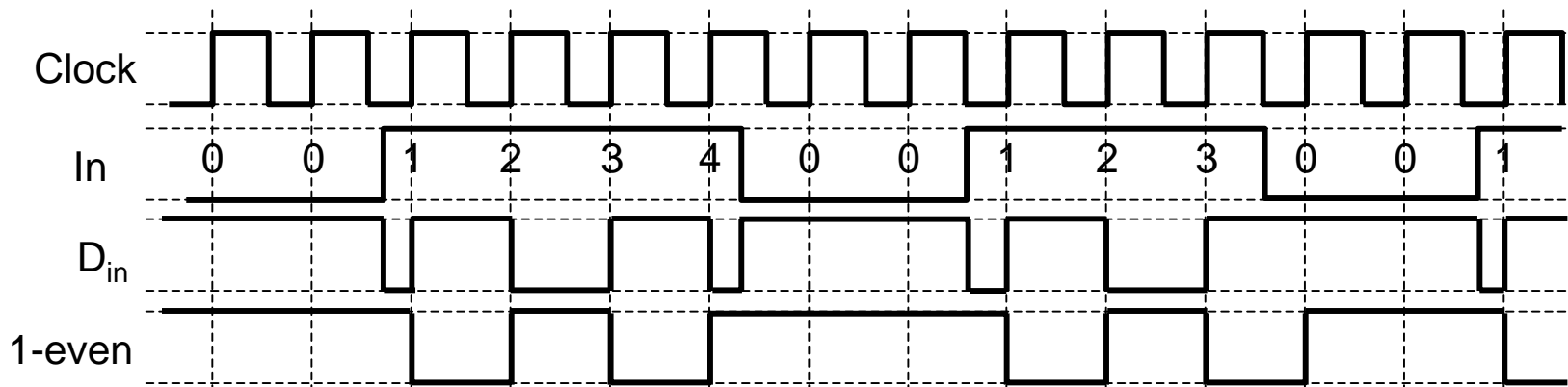
Sequential System Design

- Verify operation of design



Present State	Input	Next State	Output
1	0	1	1
1	1	0	0
0	0	1	1
0	1	1	1

Present State	Input	(QI)	(QI)'	Next State	Output
1	0	0	1	1	1
1	1	1	0	0	0
0	0	0	1	1	1
0	1	0	1	1	1



Homework 8 – due in Class 10

- No homework