Lecture 7
EE 421 / CS 425
Digital System Design

Fall 2024

Shahid Masud

Topics

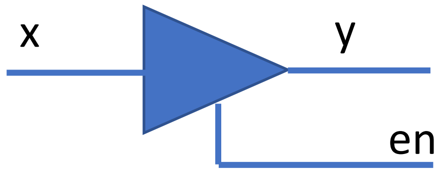
- Three State Drivers (or Buffers)
- D and T Flipflops
- Some Examples of DFF based Circuits:
    - Shift Registers
    - Counters
- Analyzing DFF based Circuits
- Synthesizing DFF Based Counters or Sequencers

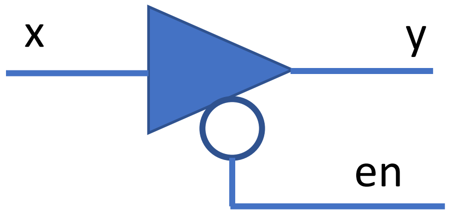
Remember:
Quiz 2 Next Week

Three State Buffers for Processor Buses

- Buses are multi-wire signal paths that connect multiple functional units in a computer system, e.g., Address bus, peripheral bus, etc.
- Simplifies design and architecture of a computer system
- Tradeoff is that the access to the bus has to be managed to avoid conflicts
- Three state hardware devices provide a dynamic interface between a bus and a circuit, providing signals when enabled else an open circuit
- The input goes to the output of the tri-state buffer when enable signal is asserted

Three State Bus Drivers

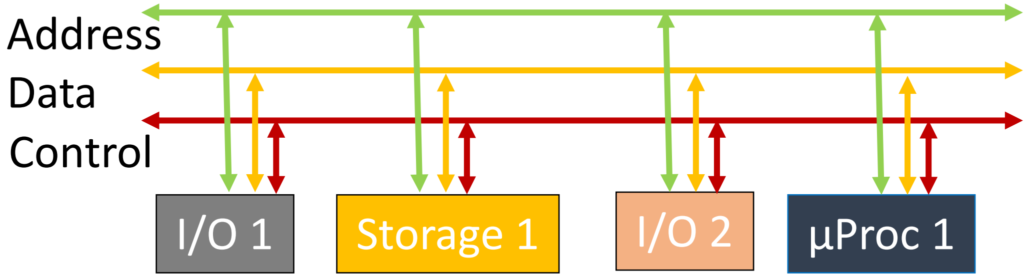




x	en	y
0	0	Hi_Z
0	1	0
1	0	Hi_Z
1	1	1

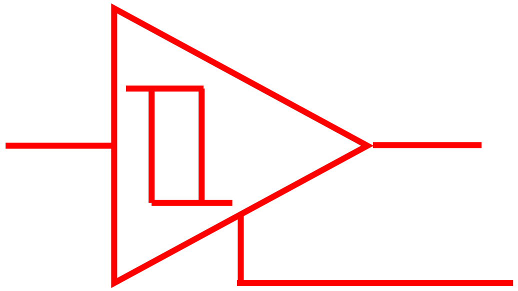
x	en	y
0	0	0
0	1	Hi_Z
1	0	1
1	1	Hi_Z

Bus Architecture

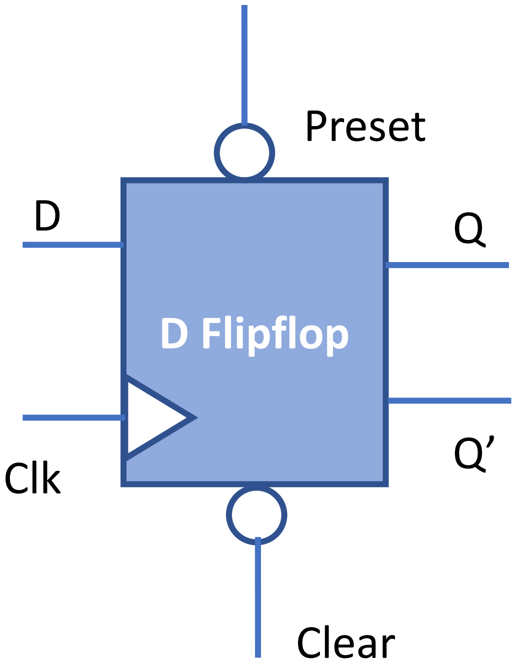


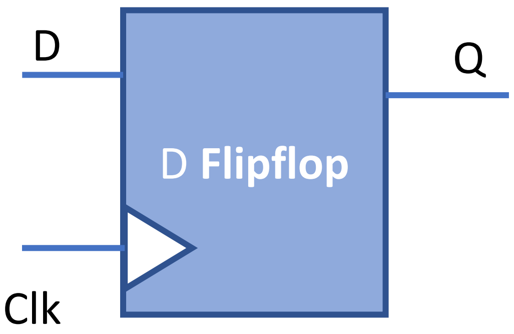
Typical Bus Structure utilizing 3-state buffers

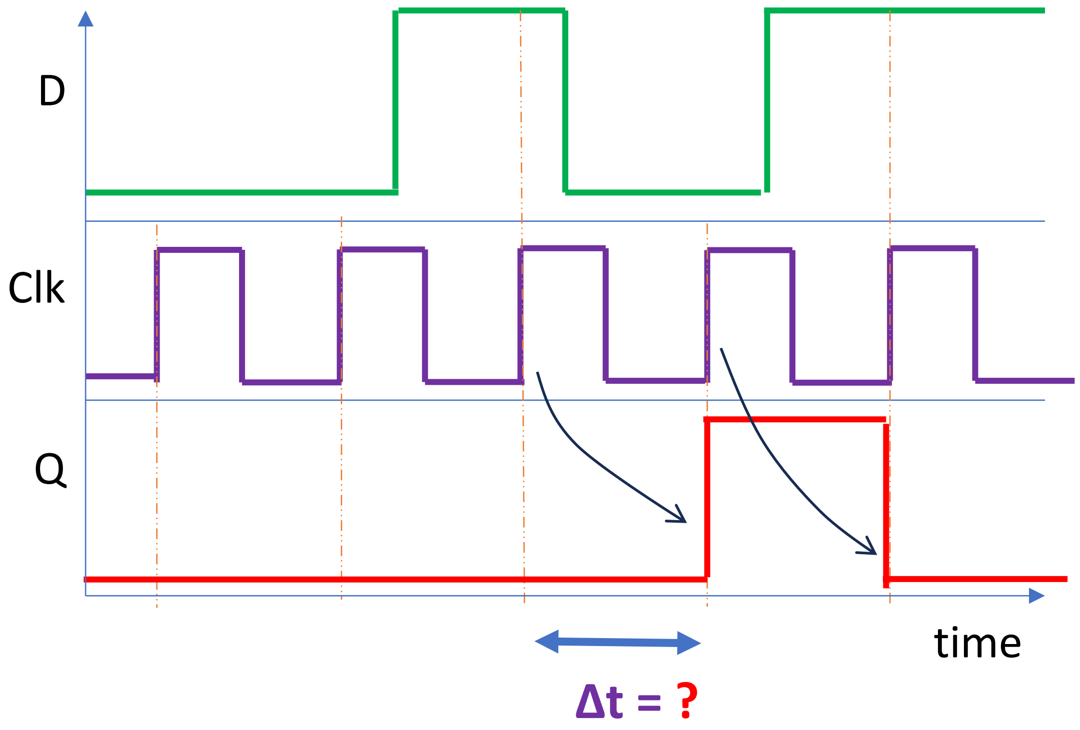
Inverting and Non-Inverting 3-State Buffers / Drivers Exist
The Enable can be active high or active low
Buffers with Hysterisis Loop are available for better noise immunity



D Flipflops – Behaviour and Equation







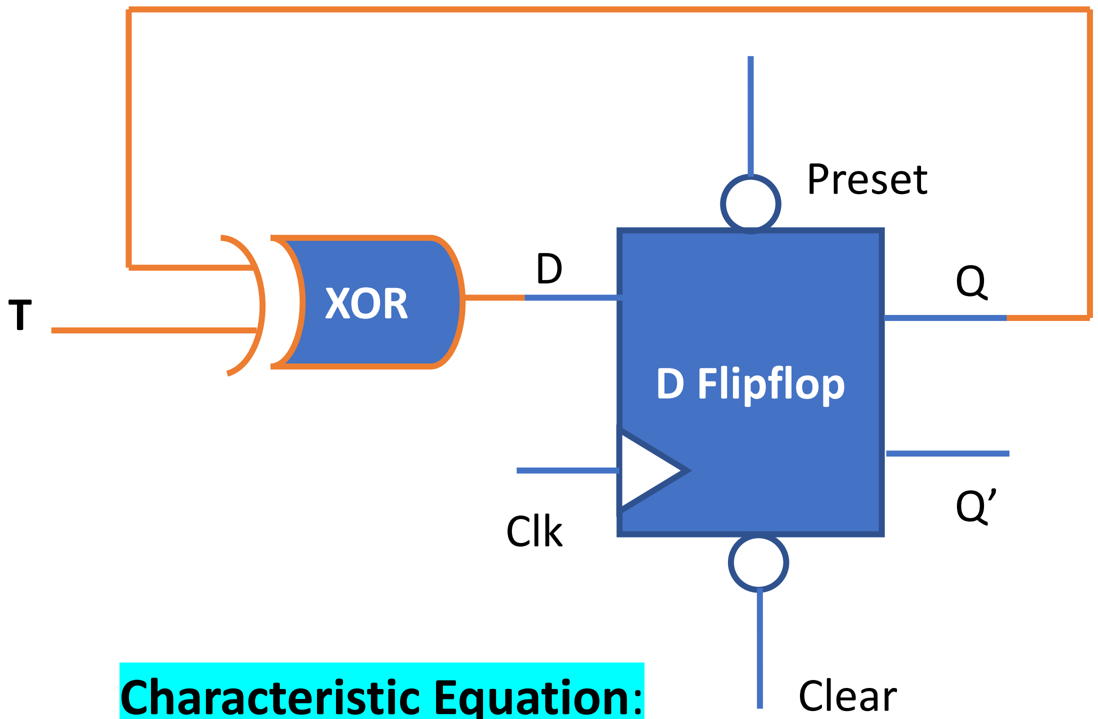
Characteristic Equation:

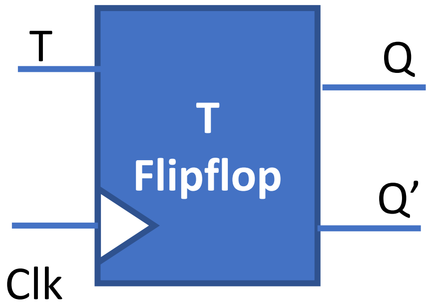
Q(t+1) = D

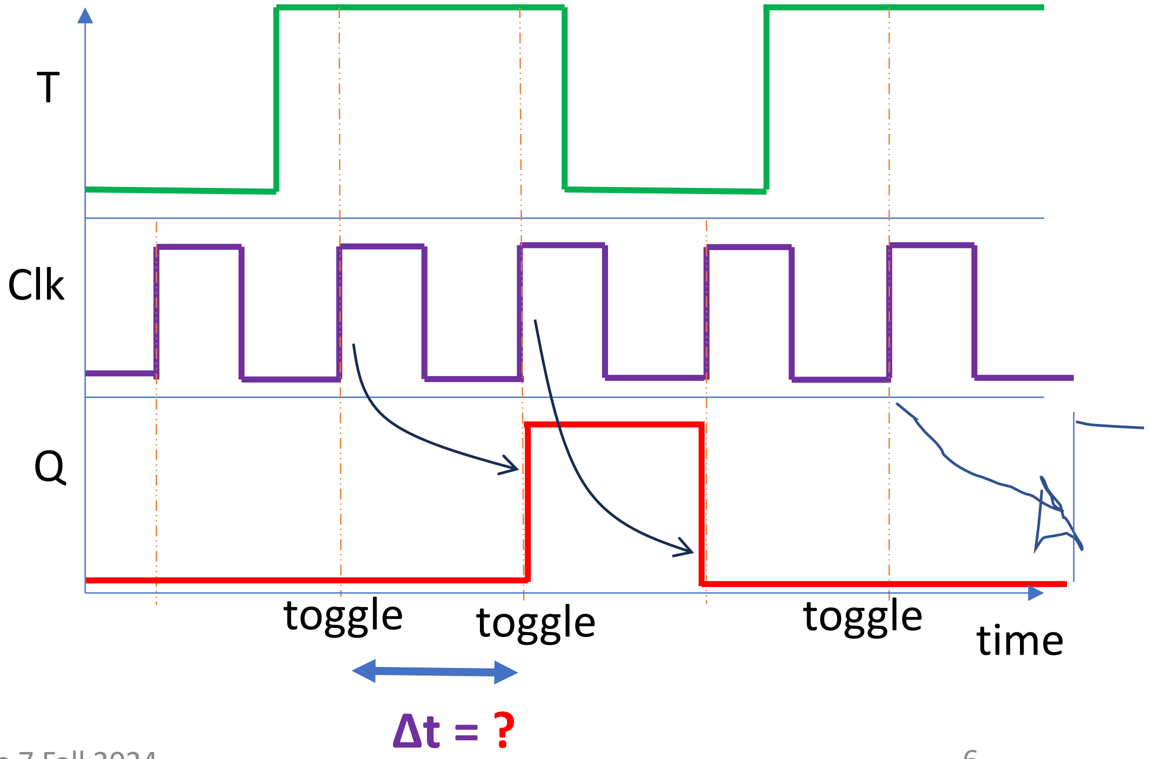
Q after clock pulse gets value of D

D	Q(t+1)
0	**0; Reset**
1	**1; Set**

T Flipflop – Behaviour and Equation







Characteristic Equation:

$$Q(t+1) = T \oplus Q$$
$$= T'Q + TQ'$$

T	Q(t+1)
0	**Q(t); No change**
1	**Q'(t); Toggle or Complement**

In this course,

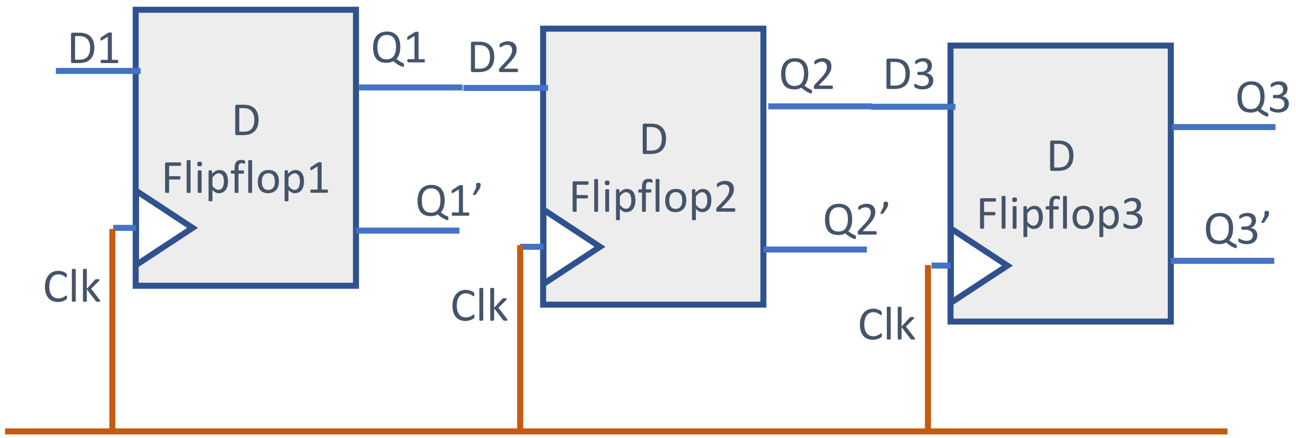
We will mostly use D Flipflops
and sometimes T Flipflops

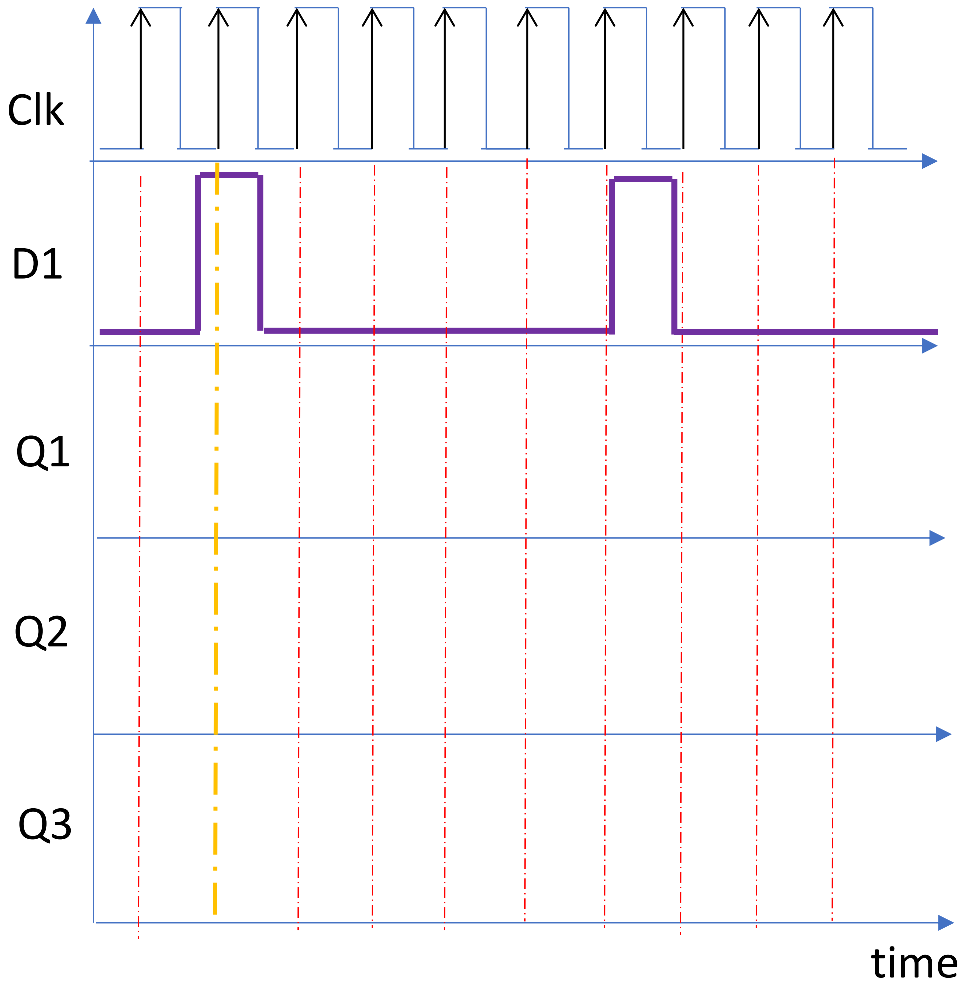
We want to study complex and high speed designs

Sequential Circuits: Counters, Sequencers, State Machines

- Counters: count in a fixed sequence, up or down
- Sequencers: sequence through a pre-defined arbitrary sequence
- State Machines: Sequential and Combinational Circuits work together to go one of many next states based on current inputs

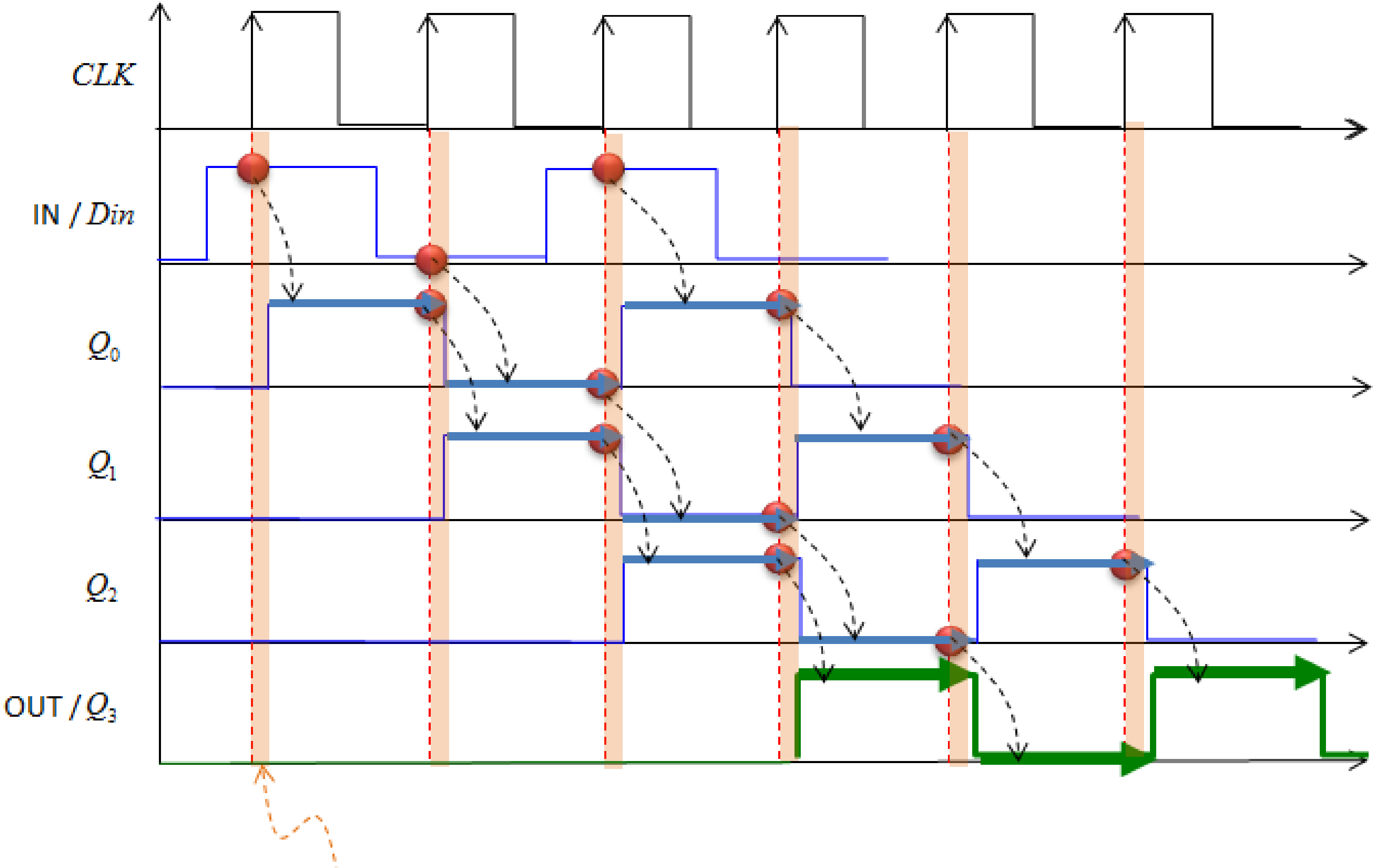
Moving Through Register Chain





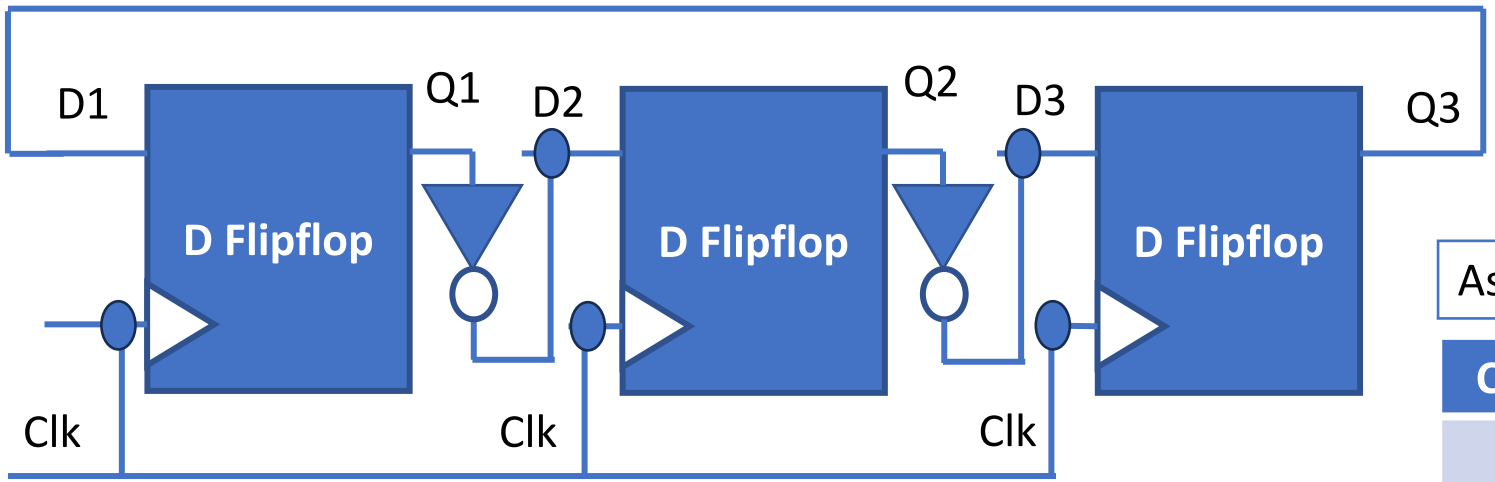
3 DFF Shift Register Timing Diagram from a website

Look closely at the Area close to Clock transition, Shaded pink lines



We assume that there is a small delay at Q state of each flipflop to make the timing diagram easily understandable.
But in reality, the situation would be a little more complicated than this simple time delay.

Cyclic Shift Register using D flipflops



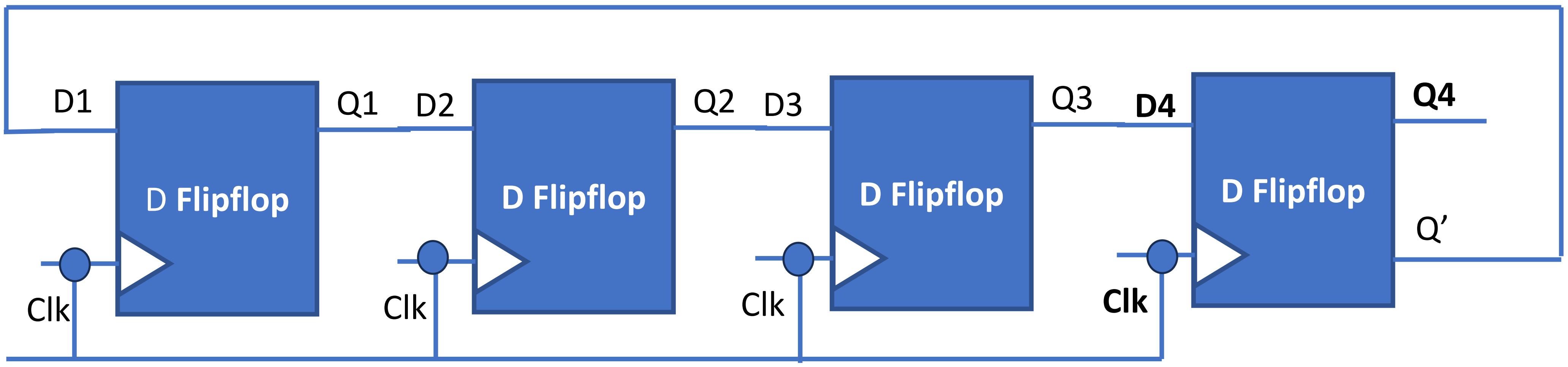
Assume all DFFs are Cleared to **0** at Startup

Clock No.	Q1	Q2	Q3
1	0	0	0
2	0	1	1
3	1	1	0
4	0	0	0
5	0	1	1
6	1	1	0
7	0	0	0
CONTINUE	....	....	....

After the clock edge:

Q1(t+1) becomes same as Q3(t)
Q2(t+1) becomes inverse of Q1(t)
Q3(t+1) becomes inverse of Q2(t)

Tail Ring Counter Four Stage



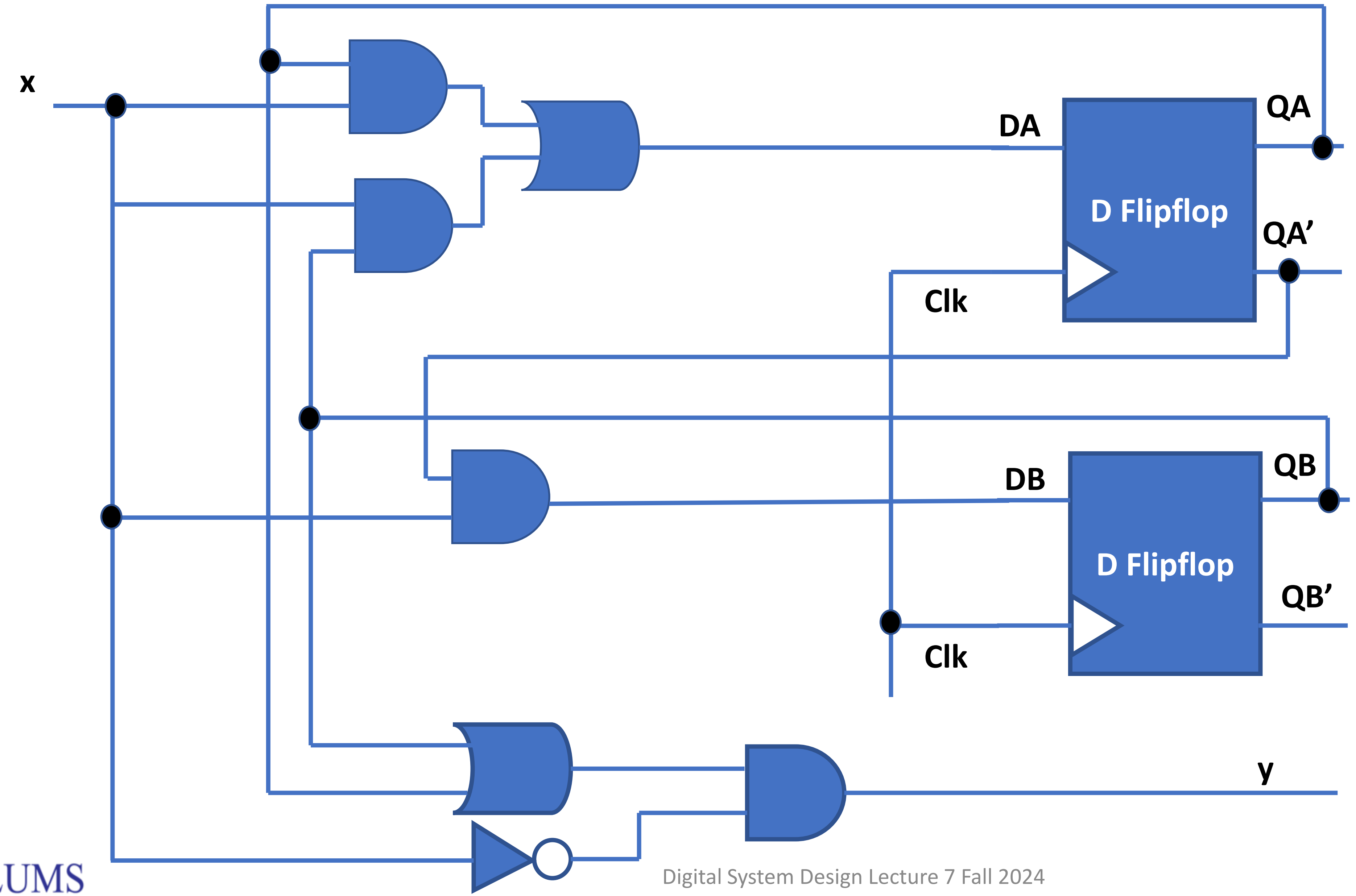
State Transition Table

Sequence No.	Q1	Q2	Q3	Q4
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1
9	0	0	0	0

Analyze Sequential Circuits

- Concept of State Equations
- Concept of State Tables
- Logic Steering through Combinational Logic
- Arbitrary Counting and Sequencing

Example Sequential Circuit for Analysis



Determine State Equations

QA(t+1) = QA(t)x(t) + QB(t) x(t)

QB(t+1) = QA’(t) x(t)

y(t) = [QA(t) + QB(t)]x’(t)

Can be written in a simplified way:

QA(t+1) = QA.(x) + QB.(x)

QB(t+1) = QA’.(x)

y= (QA + QB).(x’)

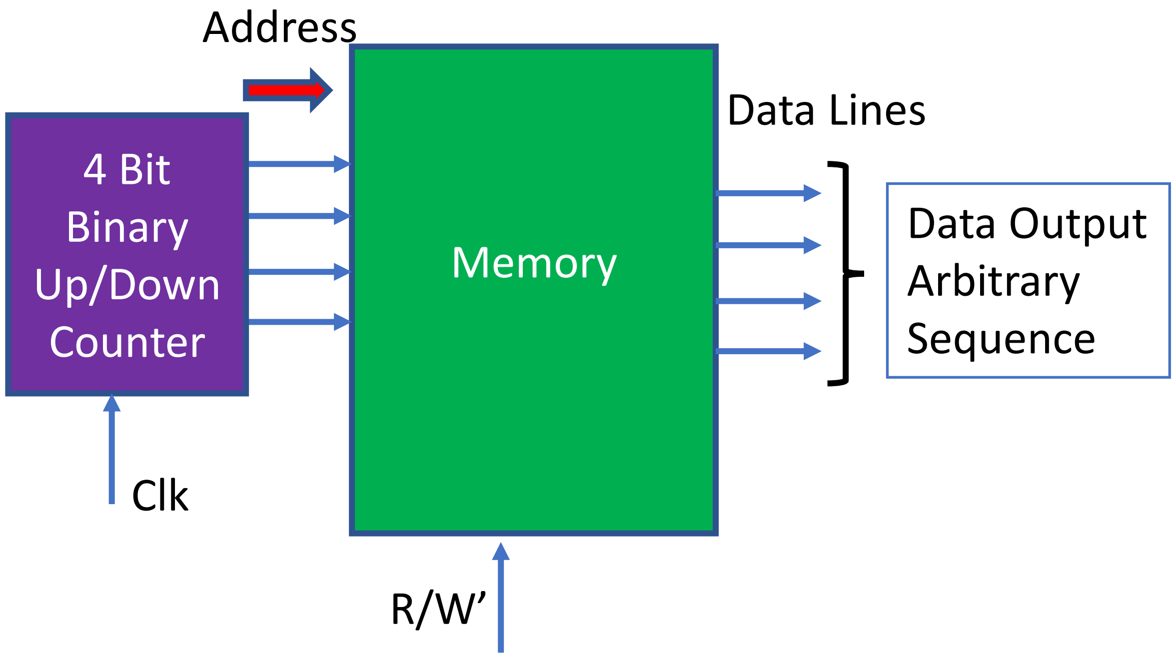
Determine State Table from State Equations

Present State	Present State	Input	Next State	Next State	Output
QA(t)	QB(t)	x	QA(t+1)	QB(t+1)	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Compact State Table

Present State		Next State				Output	
		When x = 0		When x = 1		x = 0	x = 1
A(t)	B(t)	A(t+1)	B(t+1)	A(t+1)	B(t+1)	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Using Memory to design an arbitrary sequencer



Desired Sequence is Stored in Memory Locations

Design of Synchronous Counter using DFF

- Express each number of sequence in binary code
- Use appropriate number of DFFs to represent entire sequence
- Make a State Table representing Present State and Next State
- Determine Excitation Logic Expression for Next States using K-Maps
- Complete the DFF Based Circuit using Excitation Expressions

Example Design of a Synchronous Counter (Sequencer)

Make a Synchronous Counter for sequence {0→3 →6 →9 →12 →0}

Step 1: Describe Behavior Through Truth Table

Desired Sequence In Binary form:

Clock No.	Sequence	Binary			
		Q3	Q2	Q1	Q0
1	0	0	0	0	0
2	3	0	0	1	1
3	6	0	1	1	0
4	9	1	0	0	1
5	12	1	1	0	0
6	0	0	0	0	0

Present State – Next State Table

Clock No.	Sequence	Present State				Next State			
		Q3(t)	Q2(t)	Q1(t)	Q0(t)	Q3(t+1)	Q2(t+1)	Q1(t+1)	Q0(t+1)
1	0	0	0	0	0	0	0	1	1
2	3	0	0	1	1	0	1	1	0
3	6	0	1	1	0	1	0	0	1
4	9	1	0	0	1	1	1	0	0
5	12	1	1	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	1

DFF make it convenient as Present State to Next State is only separated by a Clock Pulse

Make K-Maps to determine Next State using Next State Outputs as K-Map Entries

K-Map for D0 (Input that will give correct next state)

Q3Q2, Q1Q0	00	01	11	10
00	1		0	
01				1
11	0			
10		0		

State Equation:
D0 = Q3'Q2'Q1'Q0' + Q3'Q2Q1Q0'

K-Map for D1

Q3Q2, Q1Q0	00	01	11	10
00	1		1	
01				0
11	0			
10		0		

State Equation:

D1=Q3'Q2'Q1'Q0' + Q3'Q2'Q1Q0

K-Map for D2

Q3Q2, Q1Q0	00	01	11	10
00	0		1	
01				0
11	0			
10		1		

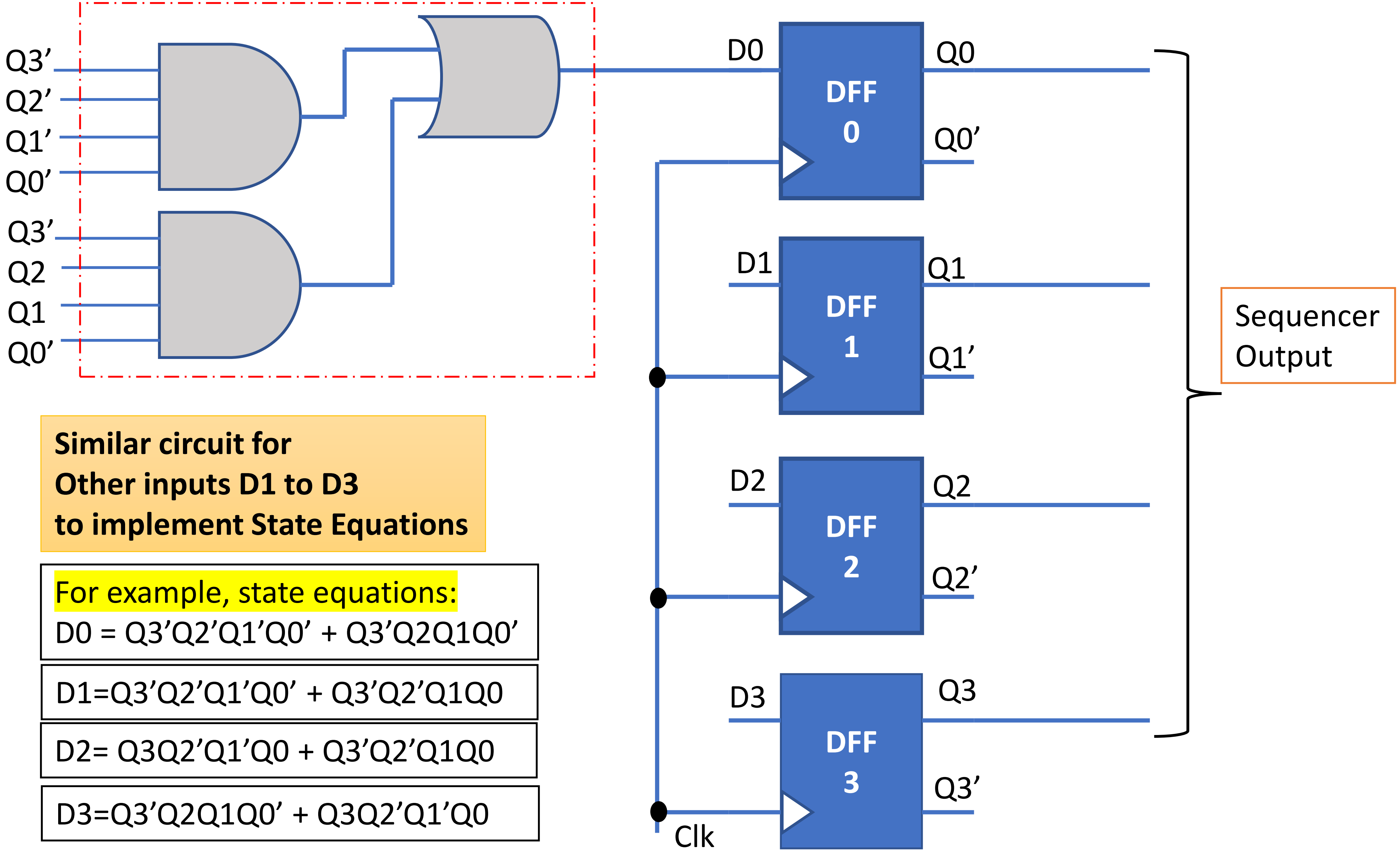
State Equation:
D2= Q3Q2’Q1’Q0 + Q3’Q2’Q1Q0

K-Map for D3

Q3Q2, Q1Q0	00	01	11	10
00	0		0	
01				1
11	0			
10		1		

State Equation:
D3=Q3'Q2Q1Q0' + Q3Q2'Q1'Q0

Final Circuit Design of Sequencer



Similar circuit for Other inputs D1 to D3 to implement State Equations

For example, state equations:

D0 = Q3'Q2'Q1'Q0' + Q3'Q2Q1Q0'

D1=Q3'Q2'Q1'Q0' + Q3'Q2'Q1Q0

D2= Q3Q2'Q1'Q0 + Q3'Q2'Q1Q0

D3=Q3'Q2Q1Q0' + Q3Q2'Q1'Q0