

Outline

□ Digital CMOS design

- Boolean algebra
- Basic digital CMOS gates
- **Combinational and sequential circuits**
- Coding - Representation of numbers



CMOS Circuits

How to implement Boolean functions
in CMOS technology ?

- A complex function cannot be implemented using a single gate
- Use a network of gates

Boolean network



CMOS Circuits

Example :

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$f = \bar{x}.y.z + x.\bar{y}.\bar{z} + x.\bar{y}.z + x.y.z$$

$$f = (x+y+z) . (x+y+\bar{z}) .$$

$$(x+\bar{y}+z) . (\bar{x}+\bar{y}+z)$$

$$f = x.(yz+\bar{y}) + \bar{x}.(y.z)$$

$$f = \bar{x}.y.z + x.\bar{y}.\bar{z} + x.z$$

$$f = x.\bar{y} + y.z$$

There is not a unique expression



CMOS Circuits

Example :

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Which gates should I use ?

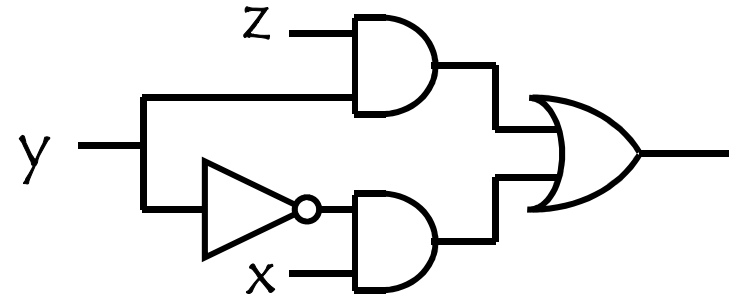
Cell library

$$f = x.\bar{y} + y.z$$

CMOS Circuits

Example :

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



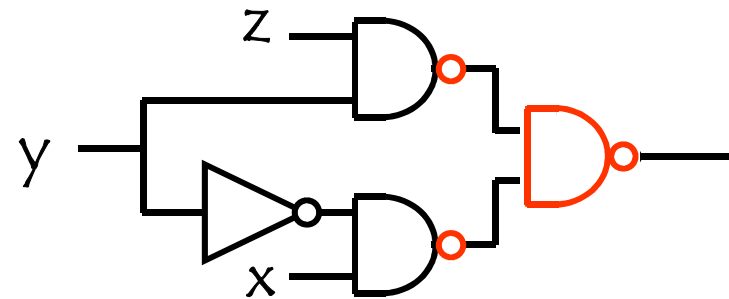
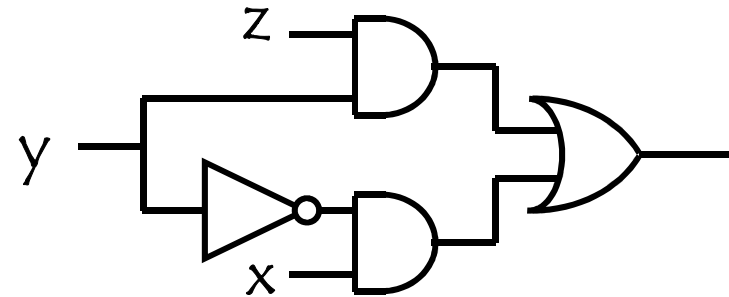
In VLSI, non-inverting gates
do NOT exist

$$f = x \cdot \bar{y} + y \cdot z$$

CMOS Circuits

Example :

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



CMOS Circuits

Example :

$$f = (\bar{x} + y) \oplus x$$

$$f = \overline{(\bar{x} + y)} \oplus \bar{x}$$

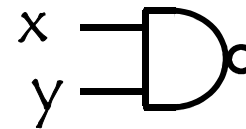
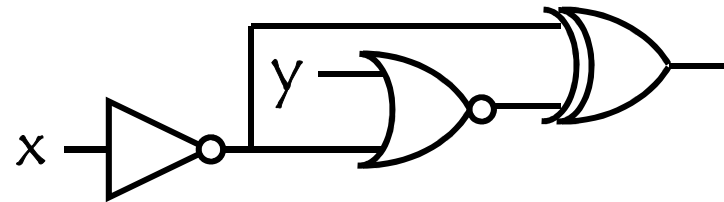
$$f = (\bar{x} + y) \cdot \bar{x} + \overline{(\bar{x} + y)} \cdot x$$

$$f = \bar{x} + y \cdot \bar{x} + x \cdot \bar{y}$$

$$f = \bar{x} + x \cdot \bar{y}$$

$$f = \bar{x} + \bar{y}$$

$$f = \overline{x \cdot y}$$



CMOS Circuits

How to implement Boolean functions
with a gate network ?

- Which expression ?

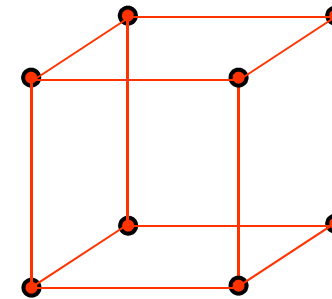
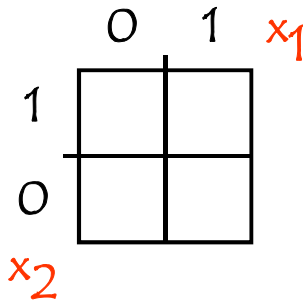
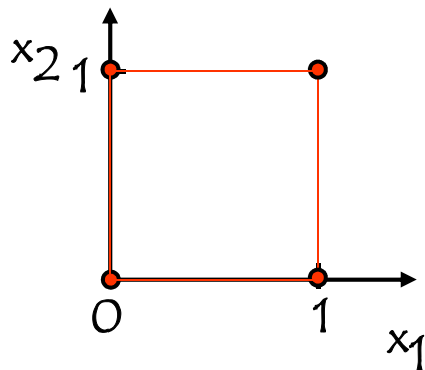


CMOS Circuits

A function can be defined by its Truth table

- Karnaugh representation gives a minimal expression

Representation of the function in a space of dimension n

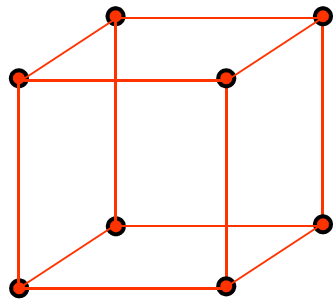


Representation of vectors' adjacency



CMOS Circuits

A Karnaugh table is represented in a flat 2-dimension table



	00	01	11	10
0		○		○
1				

The table shows a 2x4 grid of cells. The top row is labeled with binary vectors 00, 01, 11, and 10. The left column is labeled with 0 and 1. In the row labeled '0', there are small circles (minterms) in the columns for 01 and 10. A red horizontal line connects the circles in the 01 and 11 columns. A blue horizontal line connects the circles in the 11 and 10 columns. A blue curved line connects the circles in the 01 and 10 columns, passing over the 11 column. A red vertical line connects the circle in the 01 column to the cell below it in the row labeled '1'.

vectors' adjacency

CMOS Circuits

Example :

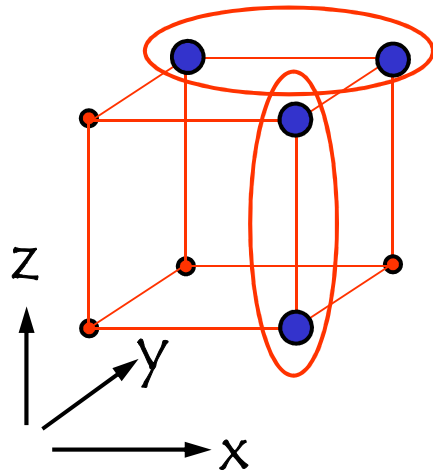
x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

	00	01	11	10	xy
0	0	0	0	1	
1	0	1	1	1	

z

CMOS Circuits

Example :



	00	01	11	10	xy
0	0	0	0	1	
1	0	1	1	1	
z					

$$f = x.\bar{y} + yz$$

CMOS Circuits

Example :

	00	01	11	10	xy
00	0	1	1	1	
01	0	1	1	0	
11	0	1	0	0	
10	0	1	1	1	

zV

$$\bar{x}.y + x.\bar{v} + y.\bar{z}$$

CMOS Circuits

Example :

	000	001	011	010	110	111	101	100	xyv
0	0	1	0	0	0	0	1	0	
1	0	1	1	1	1	0	1	0	

z

The Karnaugh map shows a 2x8 grid of cells. The columns are labeled 000, 001, 011, 010, 110, 111, 101, 100. The rows are labeled 0 and 1. The output values are 0 or 1. Red circles highlight the 1s in the 001, 011, 010, 110, and 101 columns. Red arrows point from these circles to the terms in the Boolean expression below. Blue and black arrows at the top indicate wrap-around connections between 001 and 011, 011 and 010, and 001 and 101.

$$g = \bar{x}.y.z + y.\bar{v}.z + \bar{y}.v$$

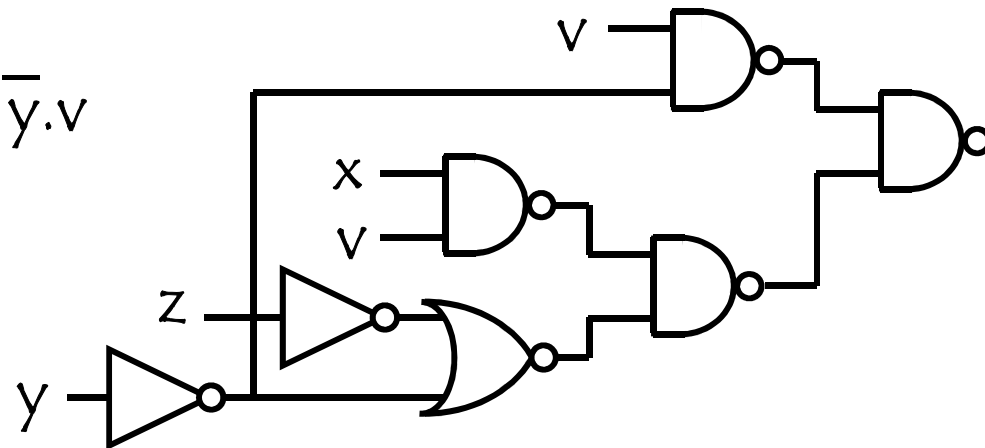
CMOS Circuits

Example :

$$g = \bar{x}.y.z + y.\bar{v}.z + \bar{y}.v$$

$$g = y.z.(\bar{x}+\bar{v}) + \bar{y}.v$$

$$g = \overline{(\bar{y}+\bar{z})}.(\bar{x}.v)} + \bar{y}.v$$



CMOS Circuits

How to implement Boolean functions
with a gate network ?

- Local optimization using Karnaugh tables

A design includes several Boolean functions

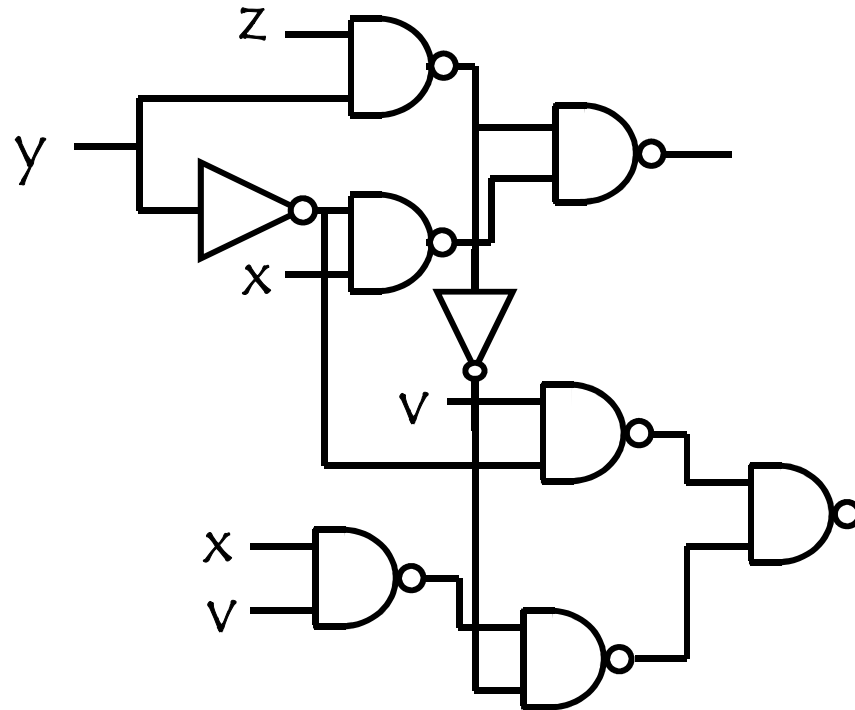


CMOS Circuits

Example :

$$g = \overline{\overline{(y+z)} \cdot \overline{(x \cdot v)}} + \overline{y} \cdot v$$

$$f = x \cdot \overline{y} + y \cdot z$$



CMOS Circuits

How to implement Boolean functions
with a gate network ?

- Local optimization using Karnaugh tables

A design includes several Boolean functions

- Global optimization by sharing sub functions



Synthesis Tool

CMOS Circuits

- **Combinational logic**

The value of the output can be determined knowing the value of the inputs

- **Sequential logic**

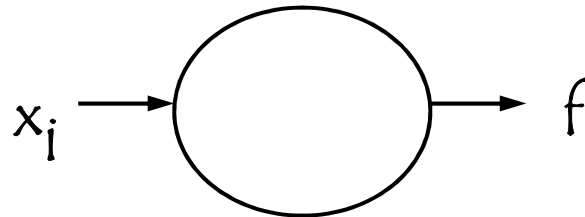
The value of the output depends on the value of the inputs **and the history**

Notion of memory



CMOS Circuits

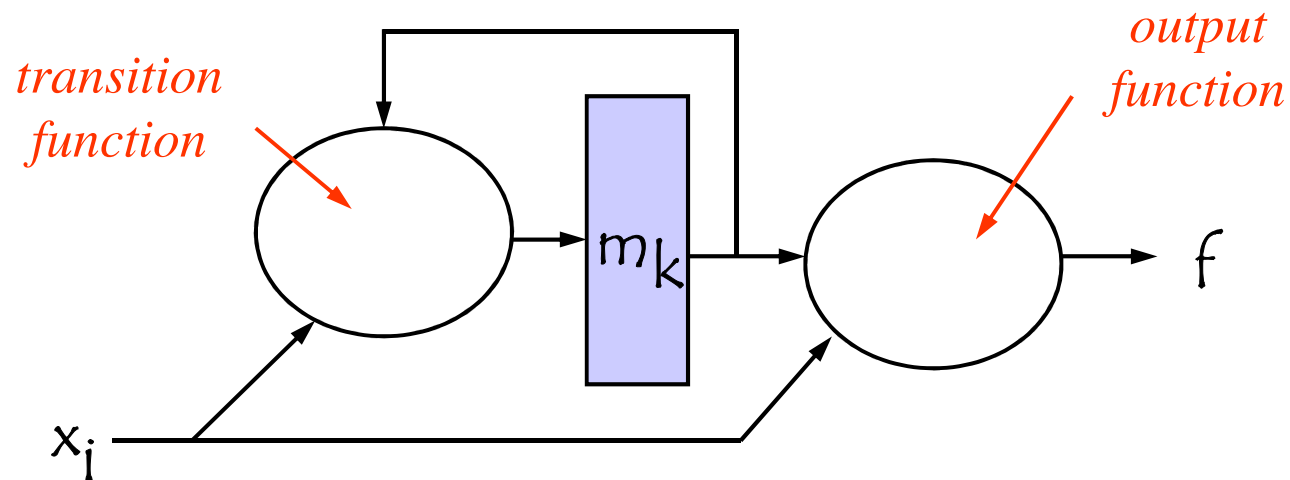
Sequential logic



$$f(x_1, \dots, x_i, \dots, x_n, m_1, \dots, m_k, \dots, m_p)$$
$$m_k(x_1, \dots, x_i, \dots, x_n, m_1, \dots, m_k, \dots, m_p)$$

CMOS Circuits

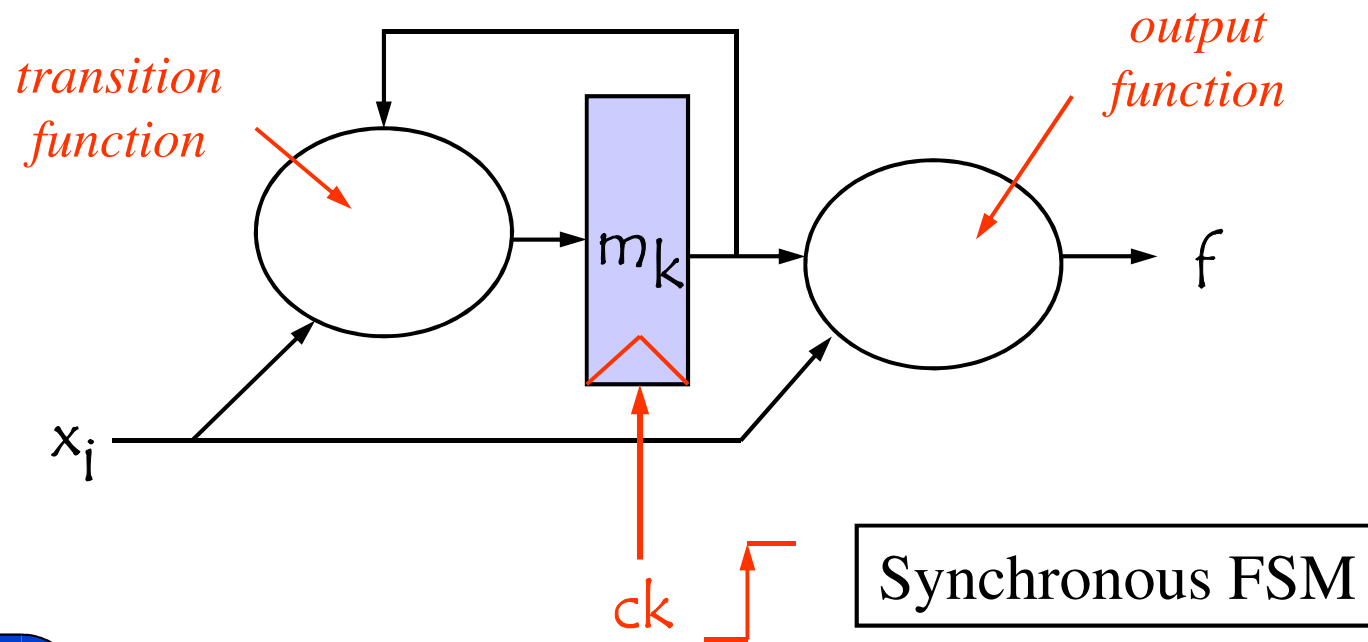
Sequential logic



Finite State Machine (FSM)

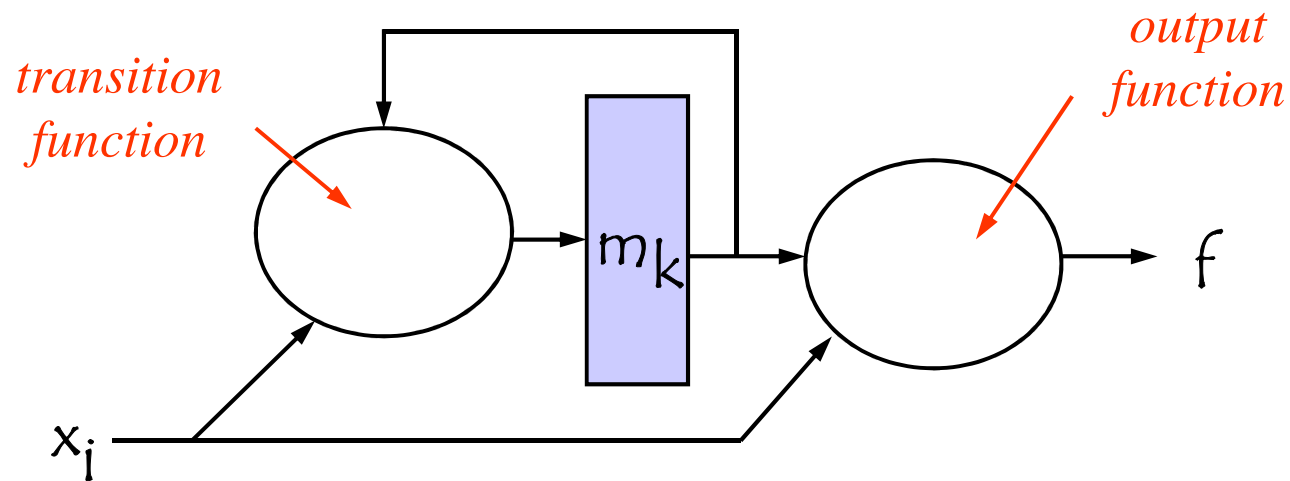
CMOS Circuits

Finite State Machine



CMOS Circuits

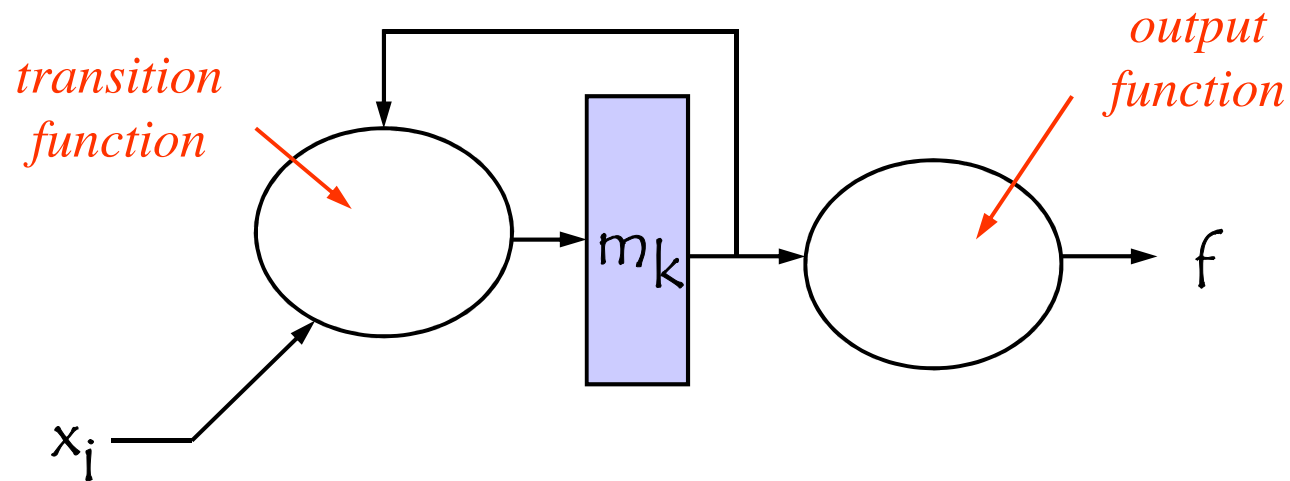
Finite State Machine



Mealy FSM

CMOS Circuits

Finite State Machine



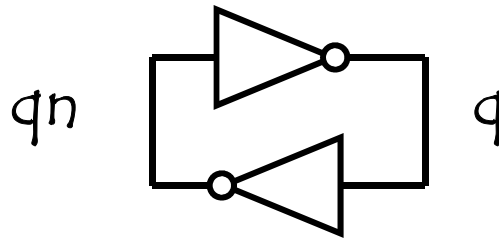
Moore FSM

CMOS Circuits

Memory :

Hold a data (0 or 1)

Write a data (0 or 1)

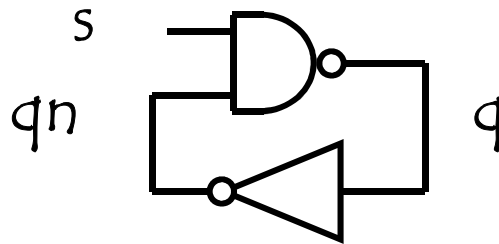


CMOS Circuits

Memory :

Hold a data (0 or 1)

Write a data (0 or 1)



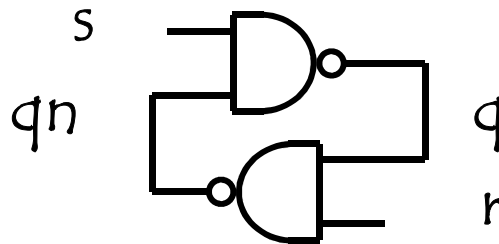
s	q	qn
0	1	0
1	\overline{qn}	\overline{q}

CMOS Circuits

Memory :

Hold a data (0 or 1)

Write a data (0 or 1)

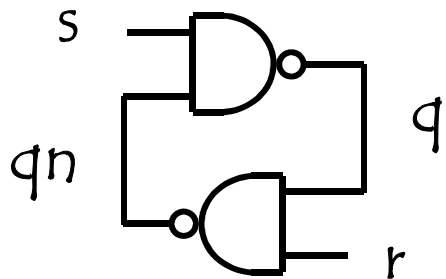


s	r	q	qn
0	1	1	0
1	0	0	1
1	1	\overline{qn}	\overline{q}
0	0	1	1

RS flip flop

CMOS Circuits

Memory :



$$q = \overline{s \cdot qn} \quad qn = \overline{r \cdot q}$$

$$f = q = \overline{s \cdot (r \cdot q)}$$

$$f = \overline{s} + (r \cdot q)$$

$$f = q \cdot (\overline{r+s}) + \overline{q} \cdot \overline{s}$$

$$\frac{\partial f^+}{\partial q} = s \cdot (\overline{s} + r) = s \cdot r$$

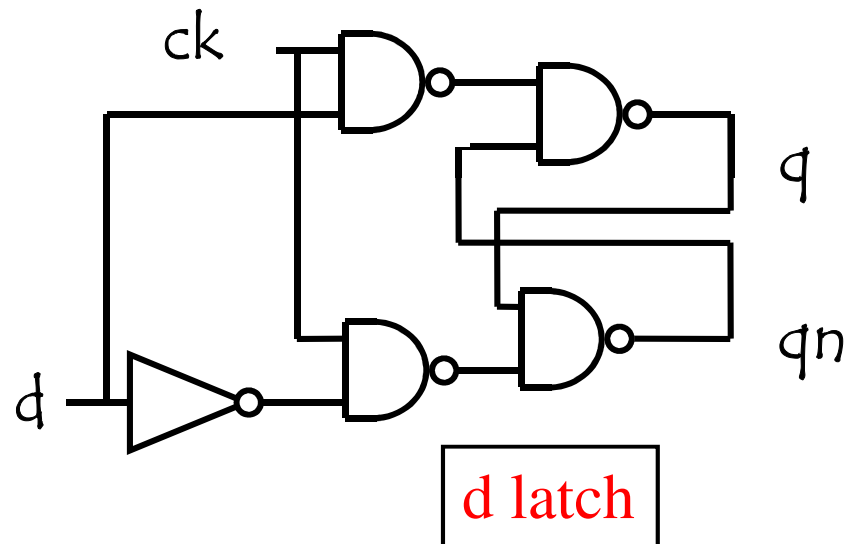
memory

$$\frac{\partial f^-}{\partial q} = \overline{s} \cdot \overline{(\overline{s} + r)} = 0$$

CMOS Circuits

Synchronous Memory :

Write a data d when the clock $ck = 1$



s	r	q	qn
0	1	1	0
1	0	0	1
1	1	\overline{qn}	\overline{q}
0	0	1	1

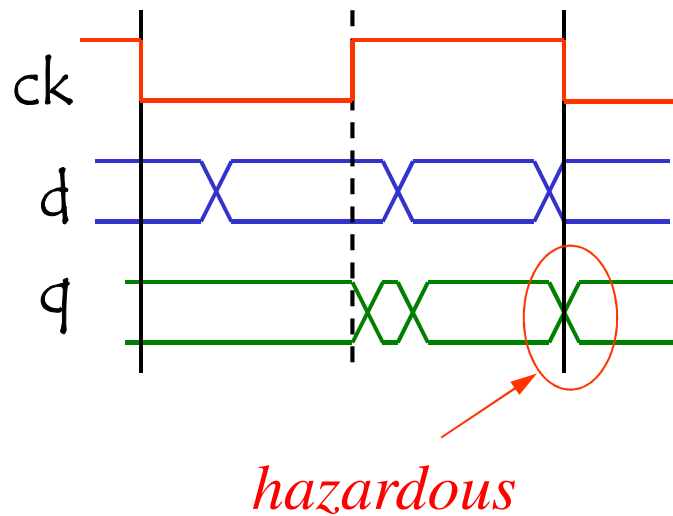
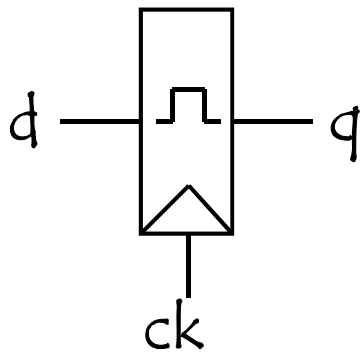
if $ck \cdot d = 1$ $s = 0$

if $ck \cdot \overline{d} = 1$ $r = 0$

CMOS Circuits

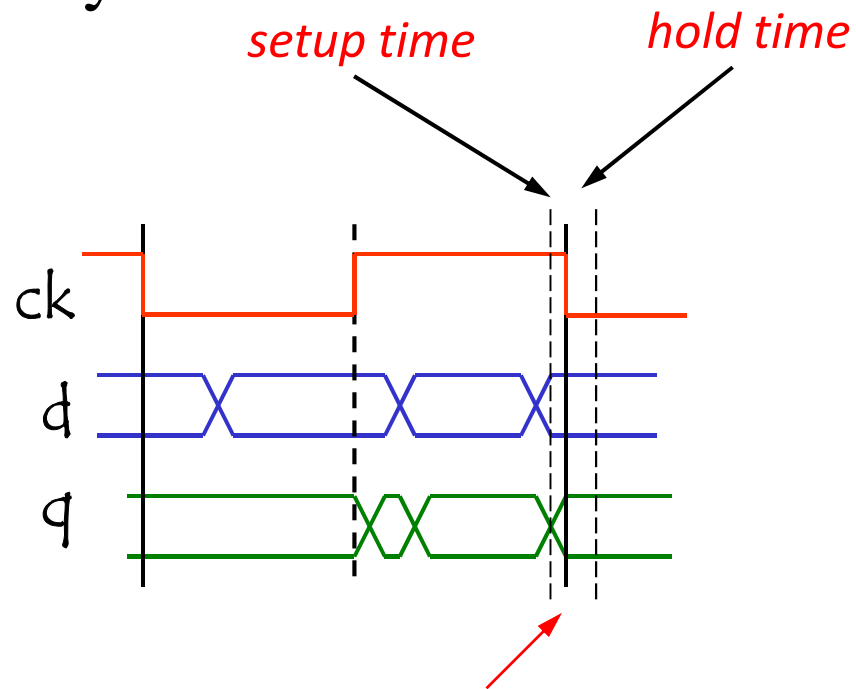
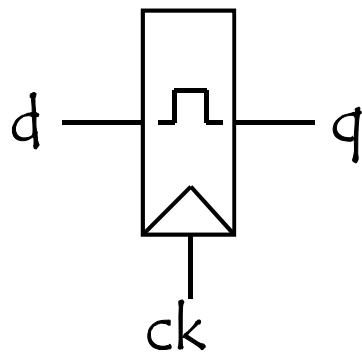
Synchronous Memory :

Write a data d when the clock $ck = 1$



CMOS Circuits

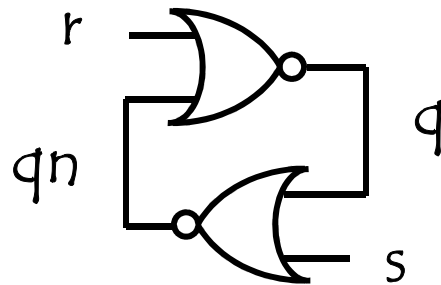
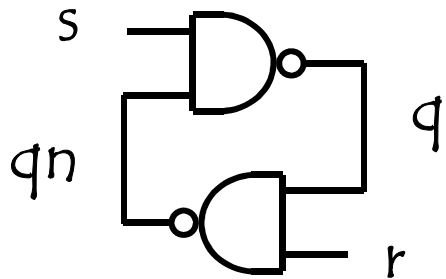
Synchronous Memory :



data should not be changed during this period

CMOS Circuits

Memory :



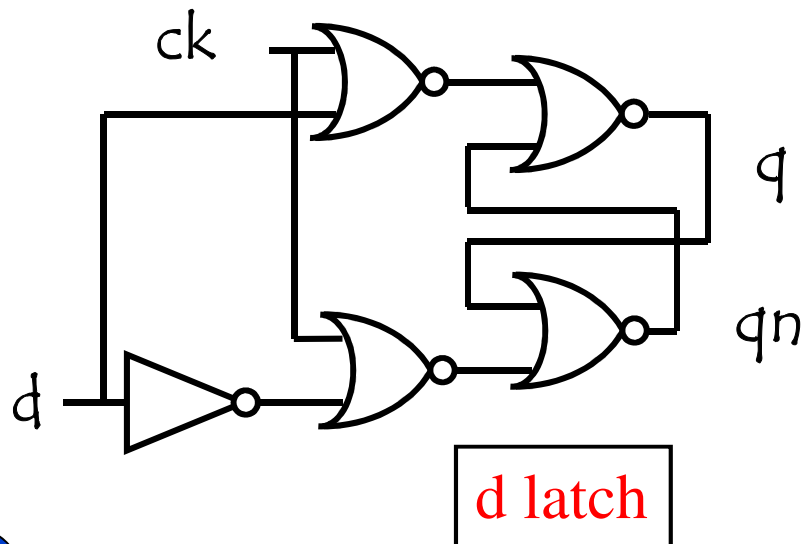
s	r	q	q_n
0	1	0	1
1	0	1	0
1	1	0	0
0	0	\bar{q}_n	\bar{q}

RS flip flop

CMOS Circuits

Synchronous Memory :

Write a data q when the clock $ck = 0$



s	r	q	qn
0	1	0	1
1	0	1	0
1	1	0	0
0	0	$\bar{q}n$	\bar{q}

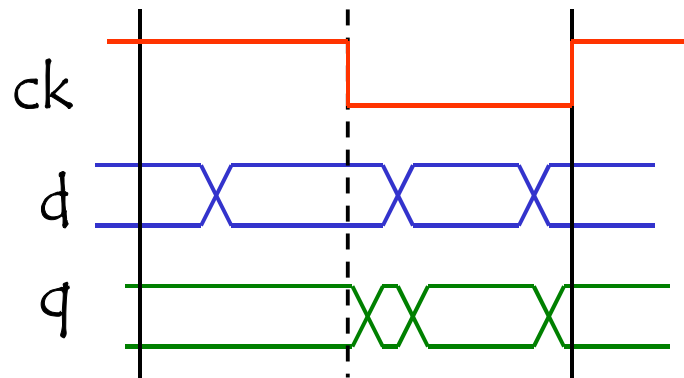
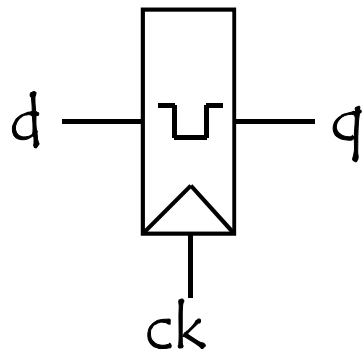
if $\bar{ck}.d = 1$ $s = 1$

if $\bar{ck}.\bar{d} = 1$ $r = 1$

CMOS Circuits

Synchronous Memory :

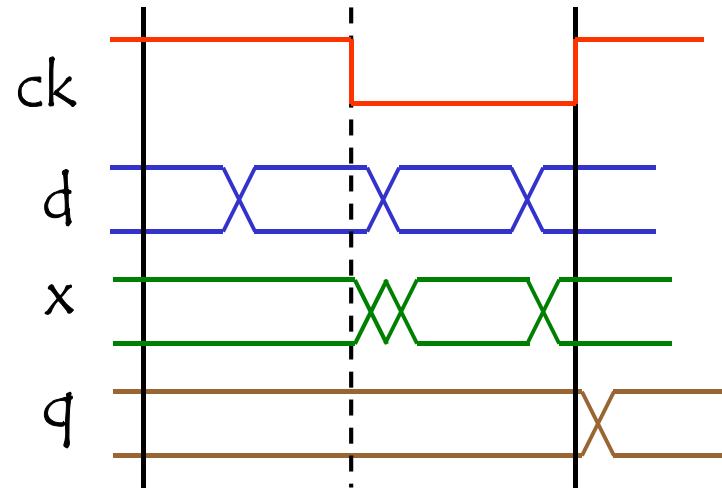
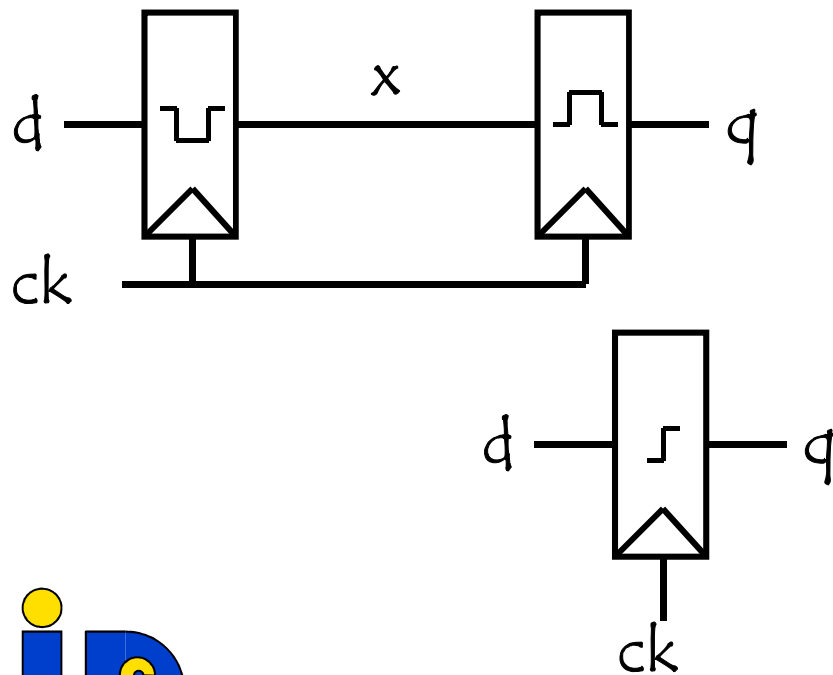
Write a data d when the clock $ck = 0$



CMOS Circuits

Synchronous Memory :

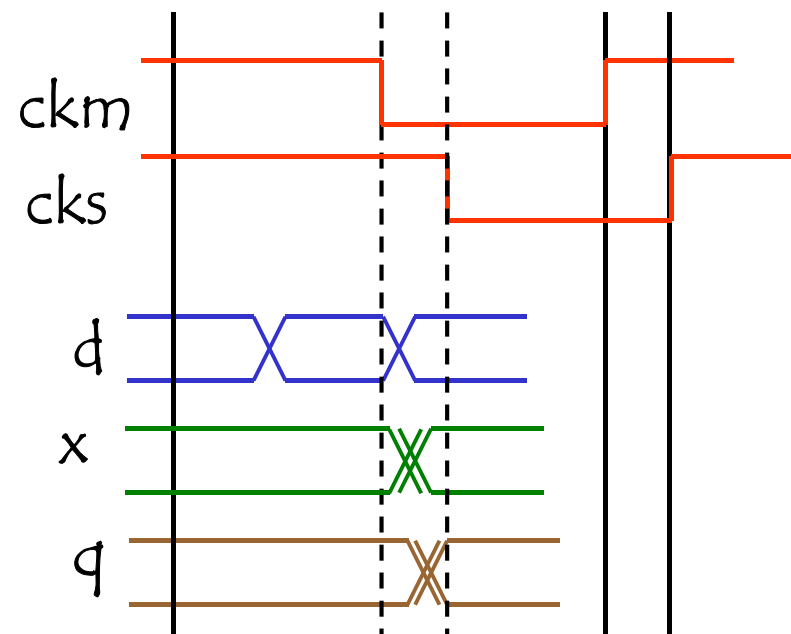
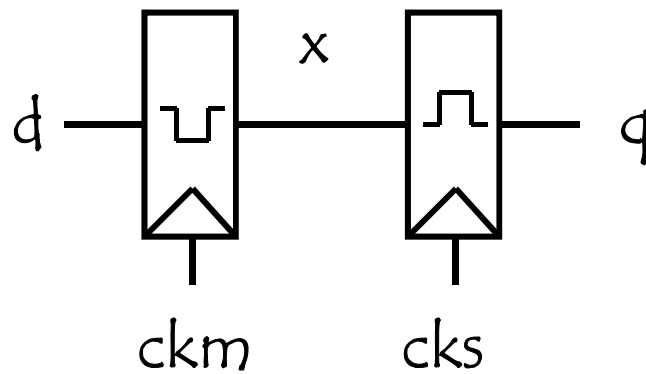
Write a data d on the rising **edge** of the clock ck



CMOS Circuits

Synchronous Memory :

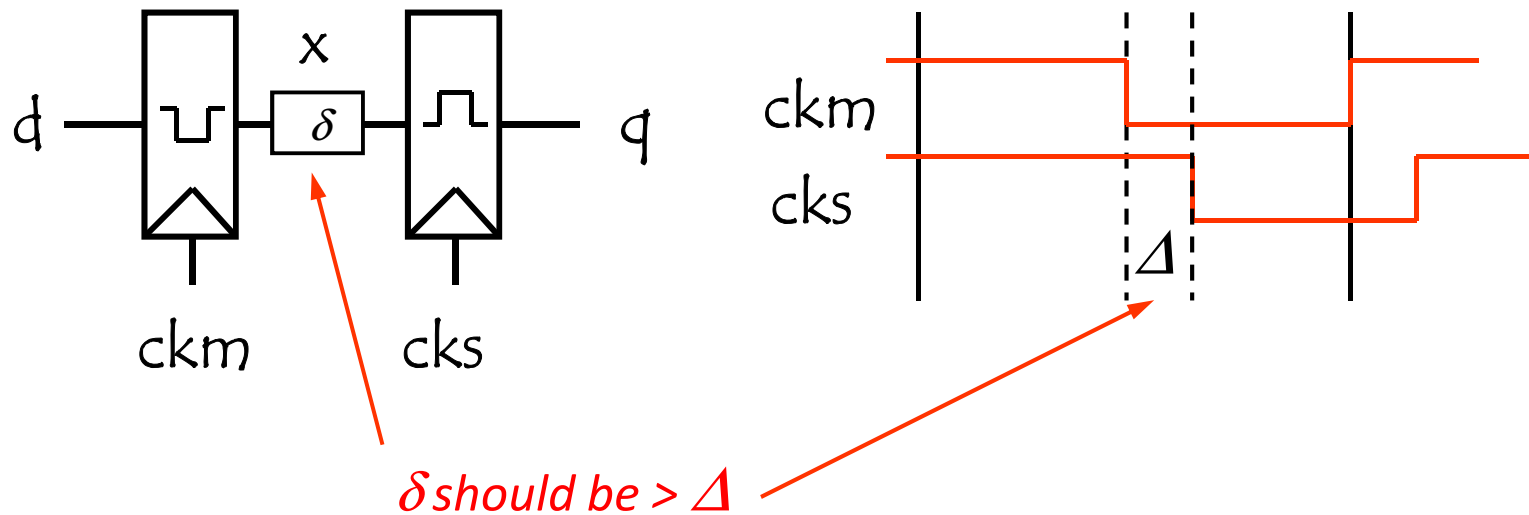
Write a data d on the rising edge of the clock ck



CMOS Circuits

Synchronous Memory :

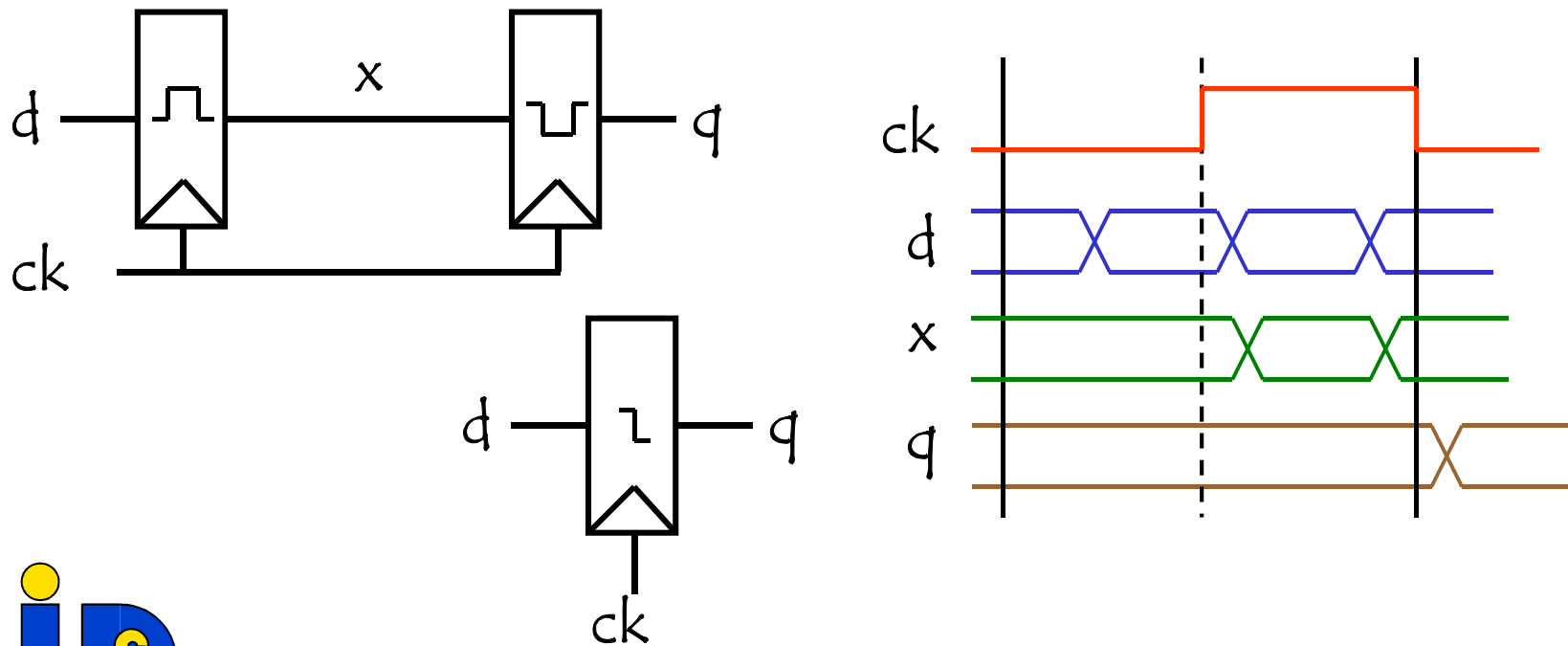
Write a data d on the rising **edge** of the clock ck



CMOS Circuits

Synchronous Memory :

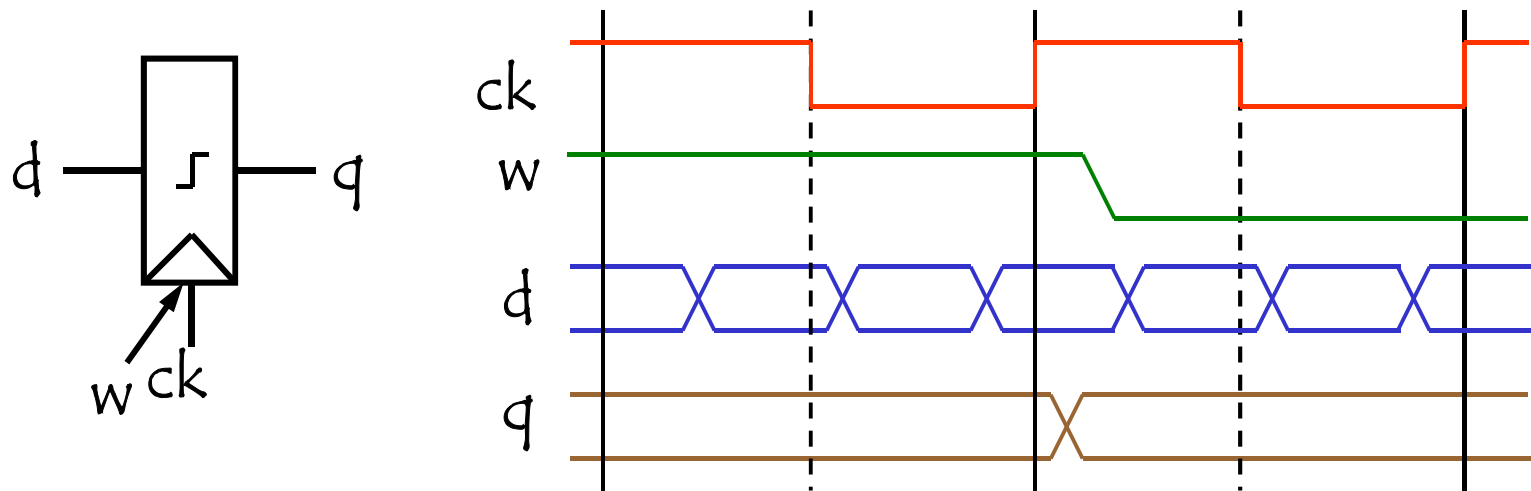
Write a data d on the falling edge of the clock ck



CMOS Circuits

Synchronous Memory :

Write a data d on the rising **edge** of the clock ck when some condition is true (write enable)



CMOS Circuits

Synchronous Memory :

Write a data d on the rising **edge** of the clock ck when some condition is true (write enable)

