2499-12

International Training Workshop on FPGA Design for Scientific Instrumentation and Computing

11 - 22 November 2013

Digital CMOS Design
Combinational and sequential circuits

Pirouz Bazargan-Sabet

Department ASIM, LIP6
University Pierre & Marie Curie (VI), 4
place Jussieu, 75252 Paris Cedex 05
France

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

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

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

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

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

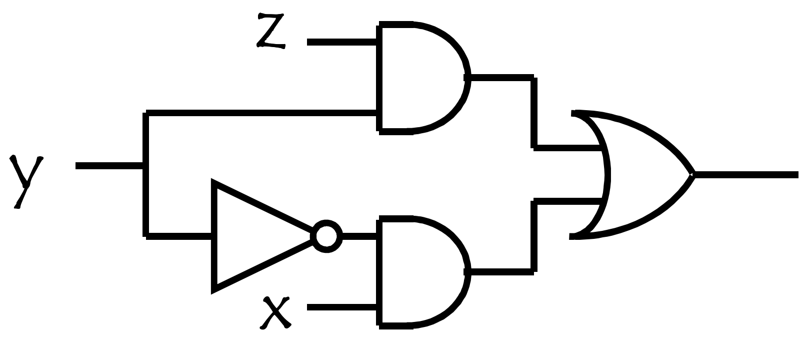
$$f = x.\overline{y} + y.z$$

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



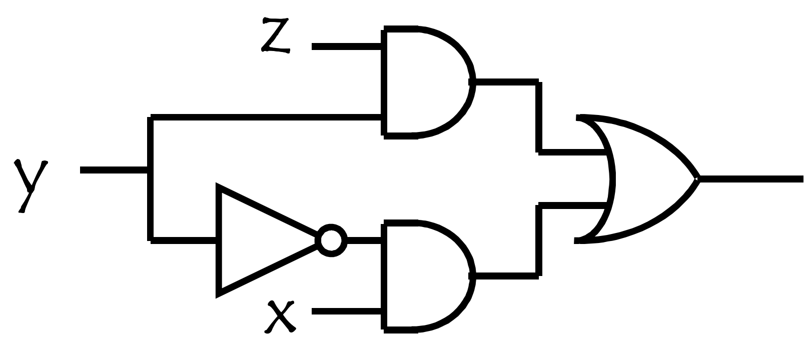
Non-inverting gates do NOT exist

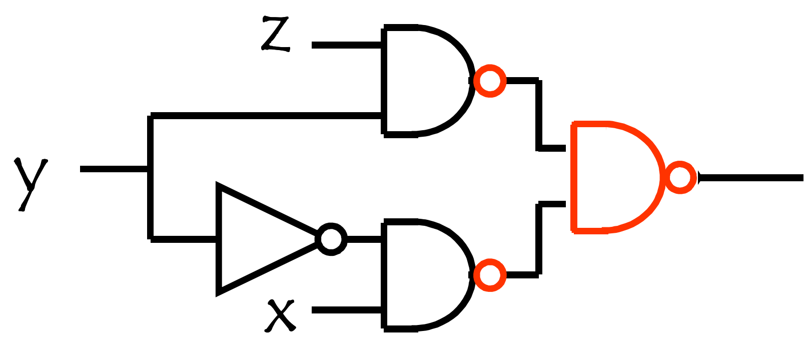
$f = x.\overline{y} + y.z$

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





CMOS Circuits

Example :

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

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

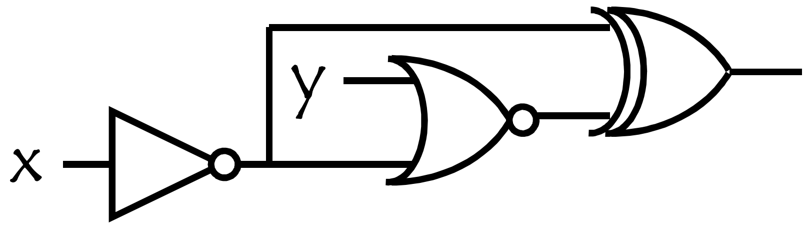
$f = (\overline{x} + y) . \overline{x} + \overline{(\overline{x} + y)} . x$

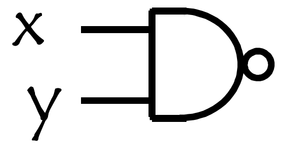
$f = \overline{x} + y.\overline{x} + x.\overline{y}$

$f = \overline{x} + x.\overline{y}$

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

$f = \overline{x . y}$





CMOS Circuits

How to implement Boolean functions with a gate network ?

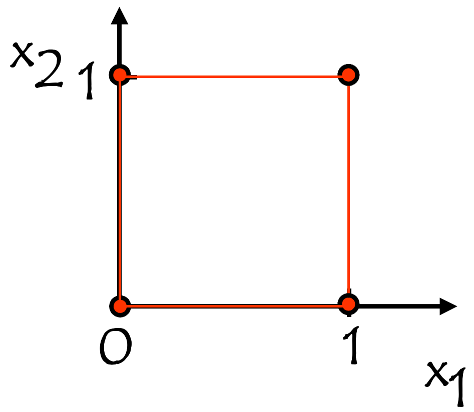
- Which expression ?

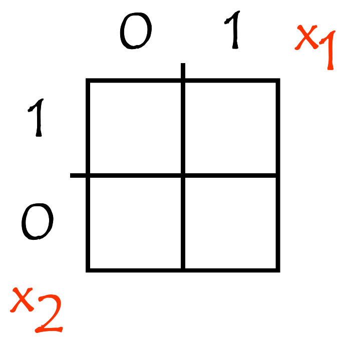
CMOS Circuits

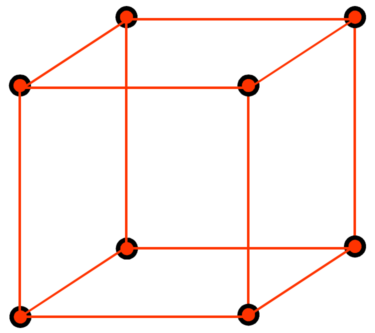
A function can be defined by its Truth table

- Karnaugh representation gives a minimal expression

Representation of the function in a space of dimension n







Representation of vectors' adjacency

CMOS Circuits

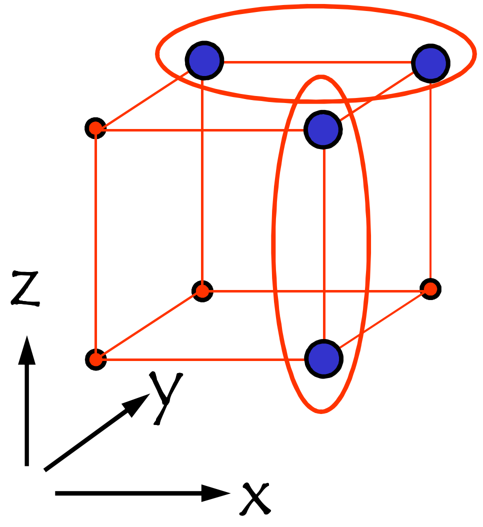
Example :

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

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

CMOS Circuits

Example :

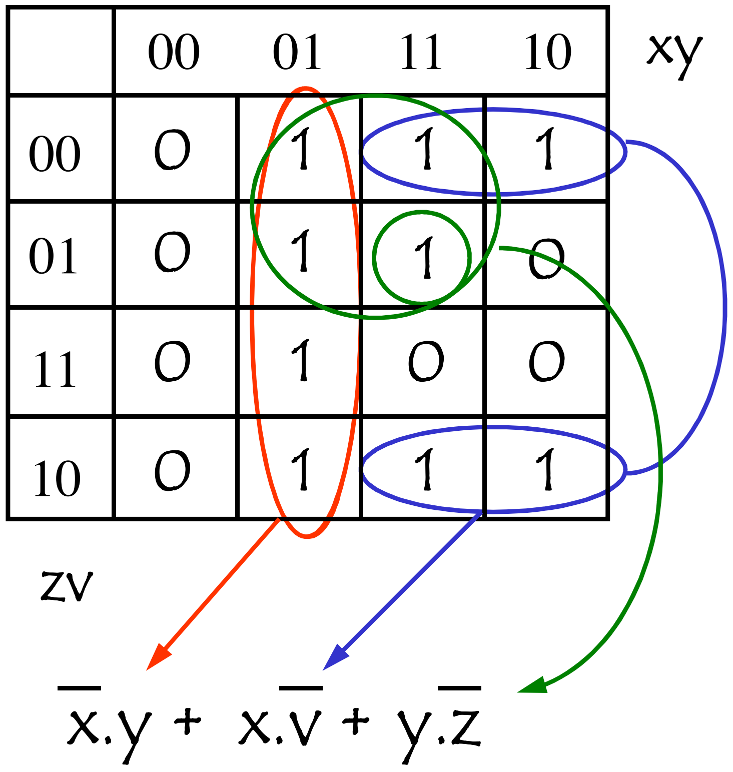


z \ xy	00	01	11	10
0	0	0	0	1
1	0	1	1	1

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

CMOS Circuits

Example :



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

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

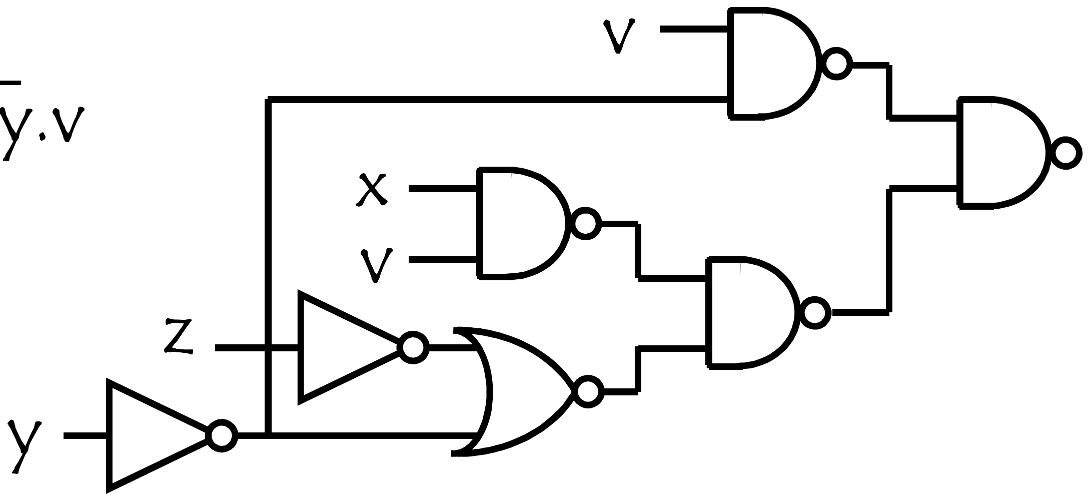
CMOS Circuits

Example :

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

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

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



CMOS Circuits

How to implement Boolean functions with a gate network ?

- Local optimization using Karnaugh tables

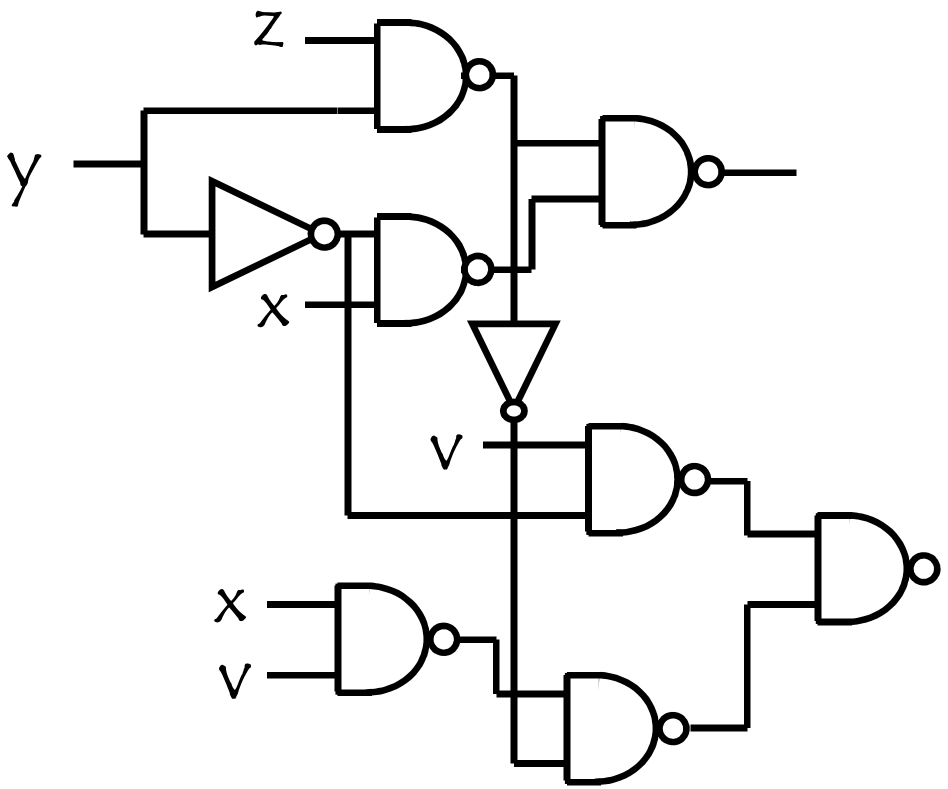
A design includes several Boolean functions

CMOS Circuits

Example :

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

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



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

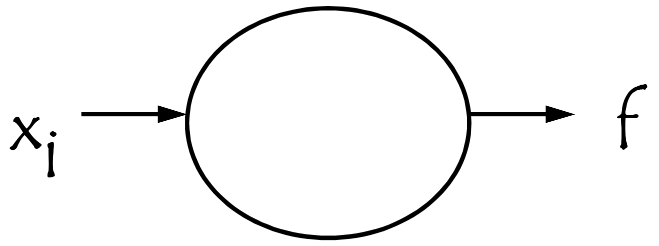
- Sequential logic

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

Notion of memory

CMOS Circuits

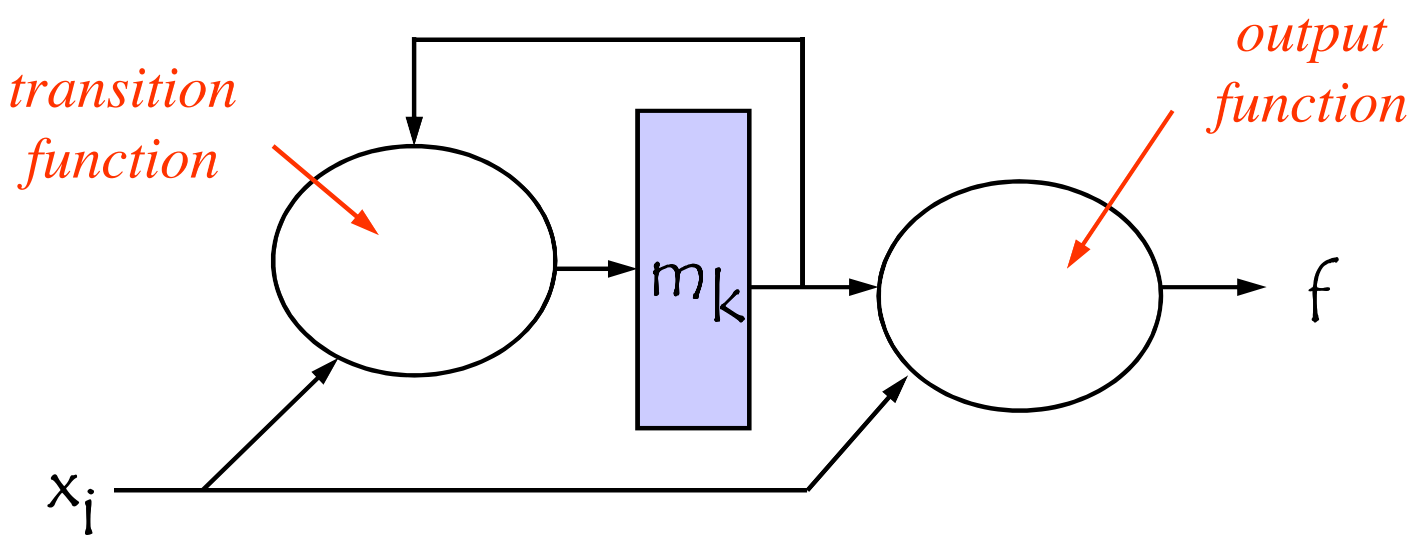
Sequential logic



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

CMOS Circuits

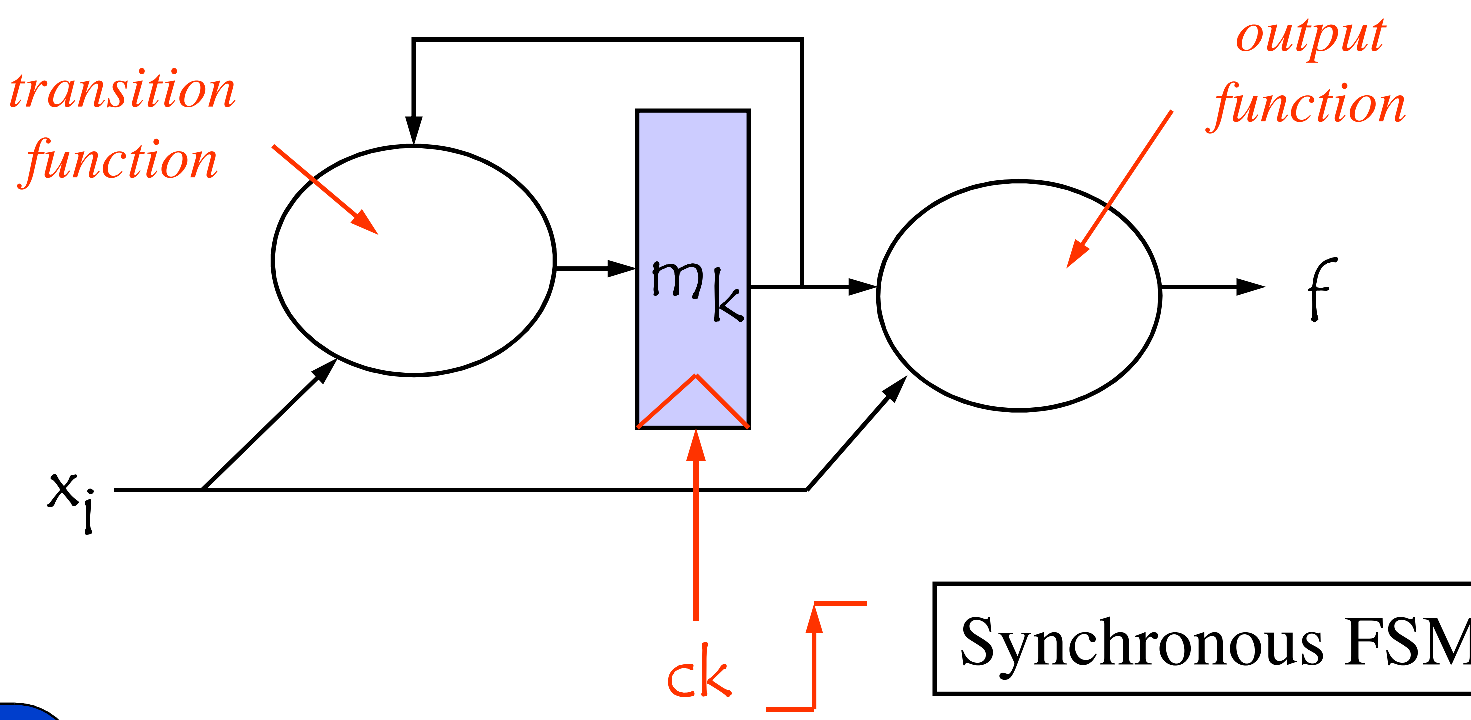
- Sequential logic



Finite State Machine (FSM)

CMOS Circuits

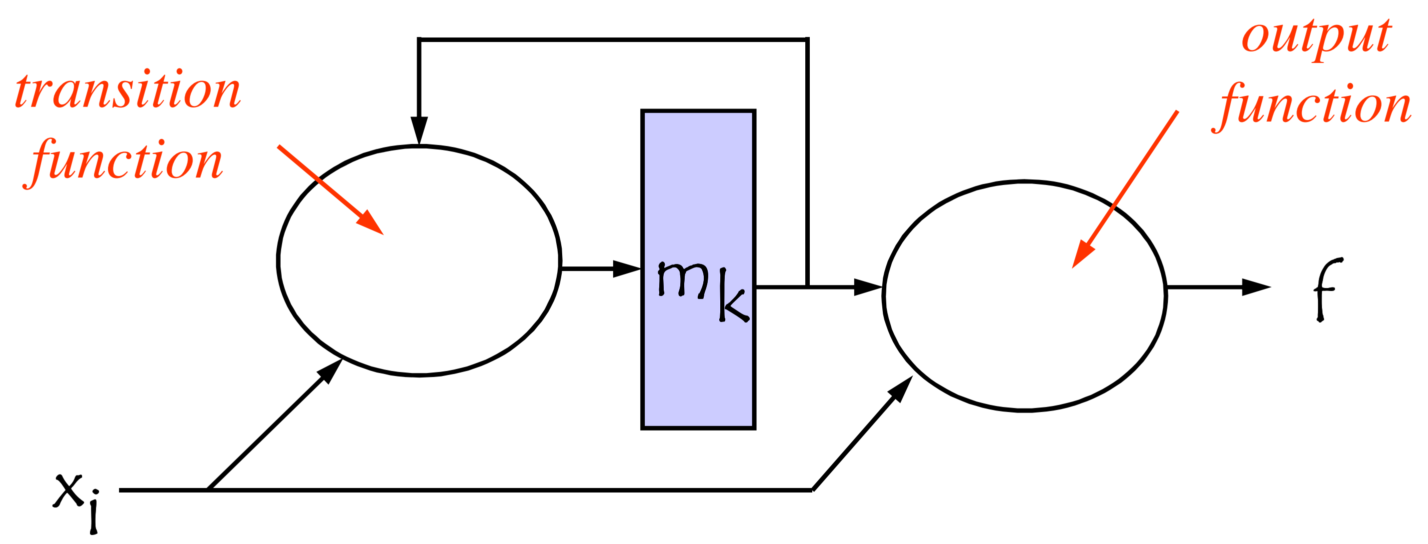
Finite State Machine



Synchronous FSM

CMOS Circuits

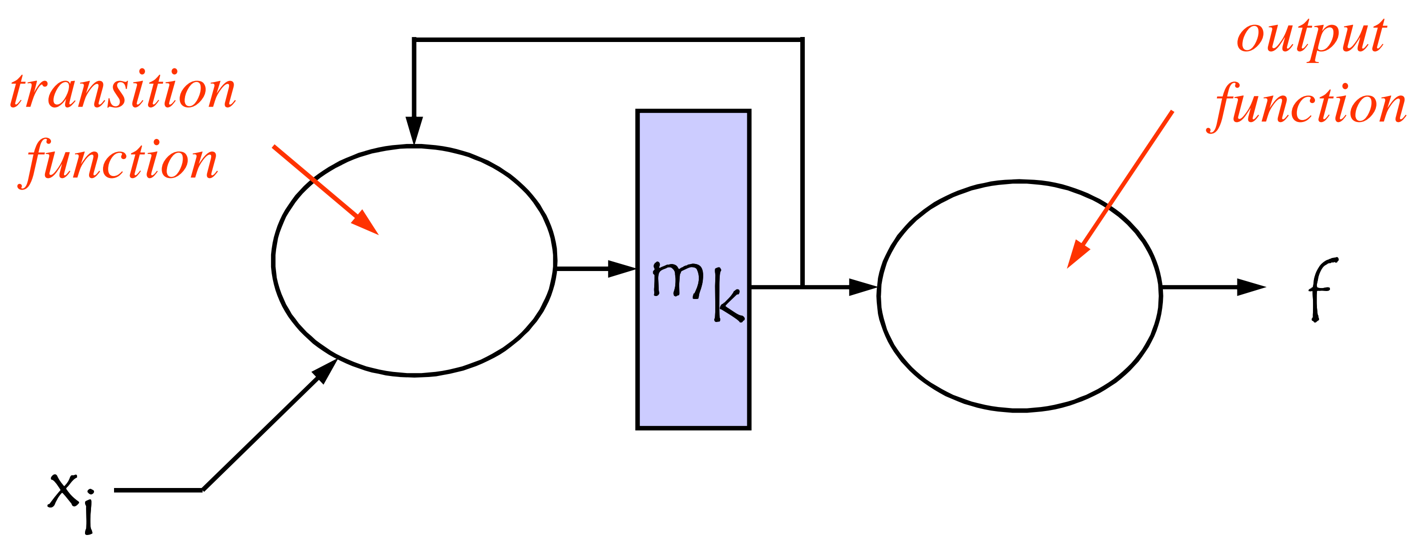
- Finite State Machine



Mealy FSM

CMOS Circuits

- Finite State Machine



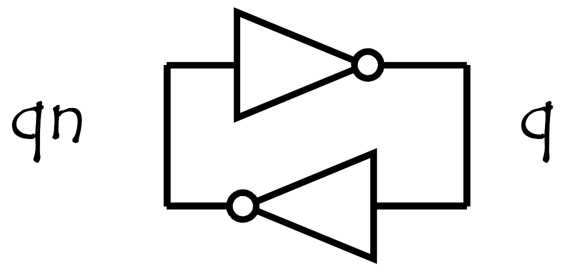
Moore FSM

CMOS Circuits

Memory :

Hold a data (0 or 1)

Write a data (0 or 1)

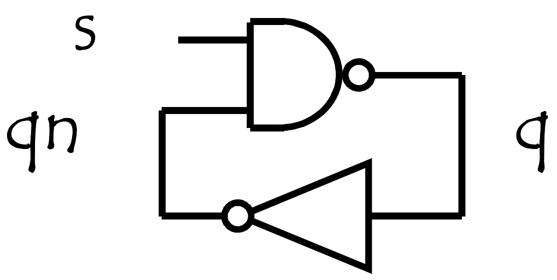


CMOS Circuits

Memory :

Hold a data (0 or 1)

Write a data (0 or 1)



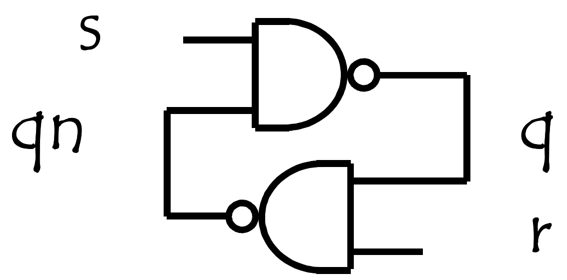
s	q	qn
0	1	0
1	q	qn

CMOS Circuits

Memory :

Hold a data (0 or 1)

Write a data (0 or 1)



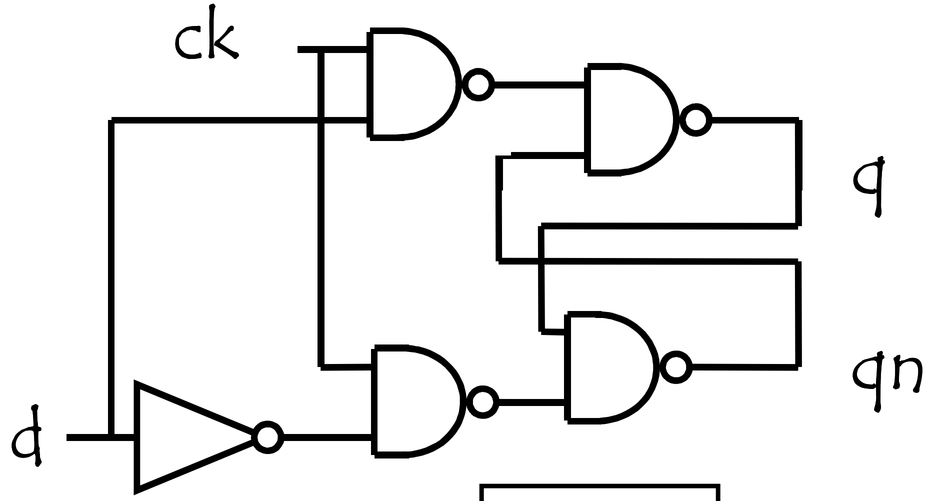
s	r	q	qn
0	1	1	0
1	0	0	1
1	1	q	qn
0	0	1	1

RS flip flop

CMOS Circuits

Synchronous Memory :

Write a data d when the clock ck = 1



d latch

s	r	q	qn
0	1	1	0
1	0	0	1
1	1	q	qn
0	0	1	1

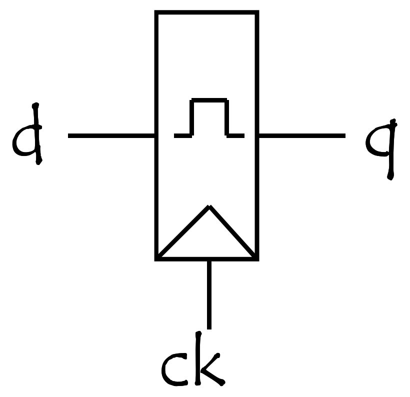
if $ck.d = 1$ $s = 0$

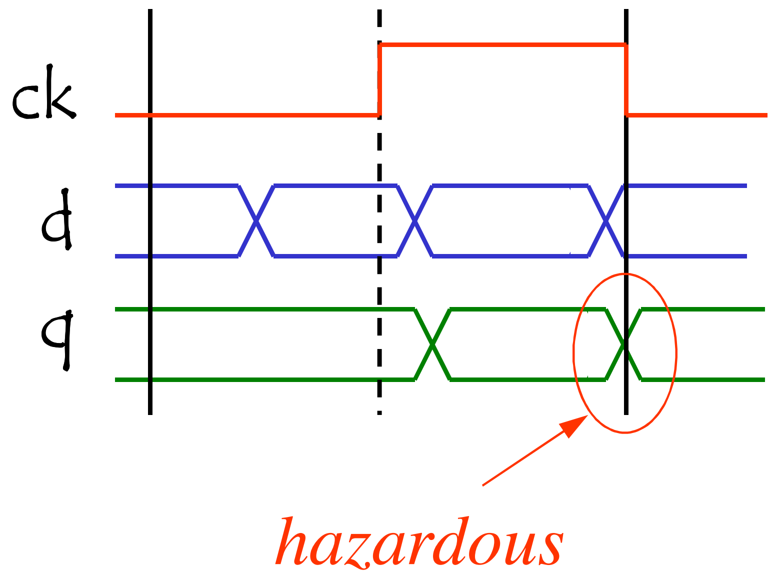
if $ck.\overline{d} = 1$ $r = 0$

CMOS Circuits

Synchronous Memory :

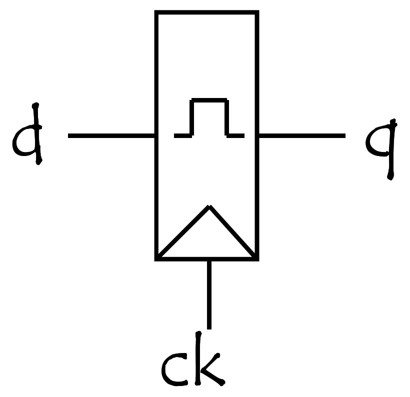
Write a data d when the clock ck = 1

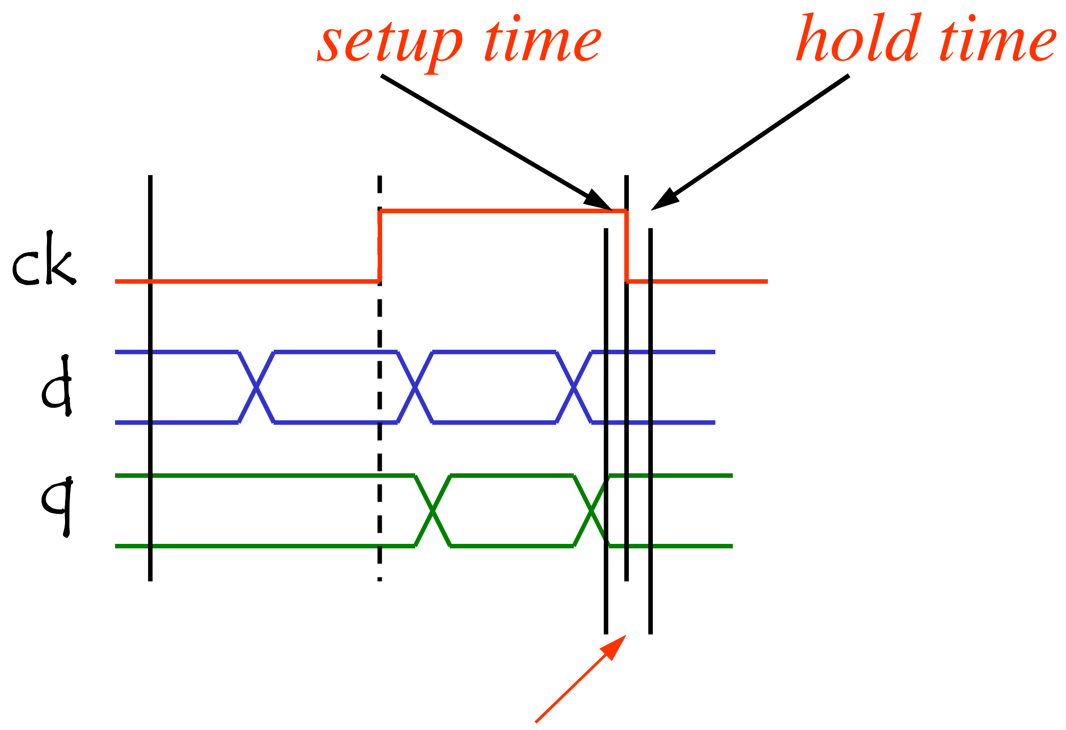




CMOS Circuits

Synchronous Memory :

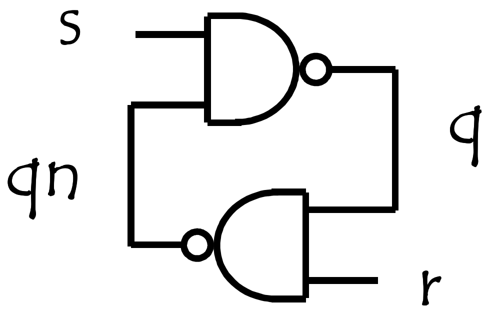


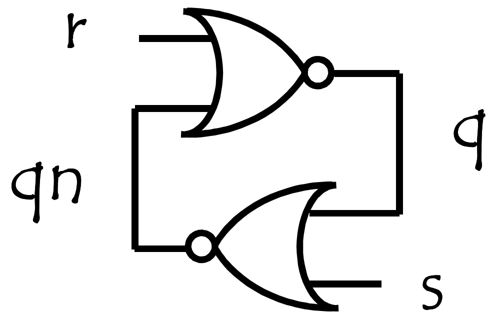


data should not change in this period

CMOS Circuits

Memory :





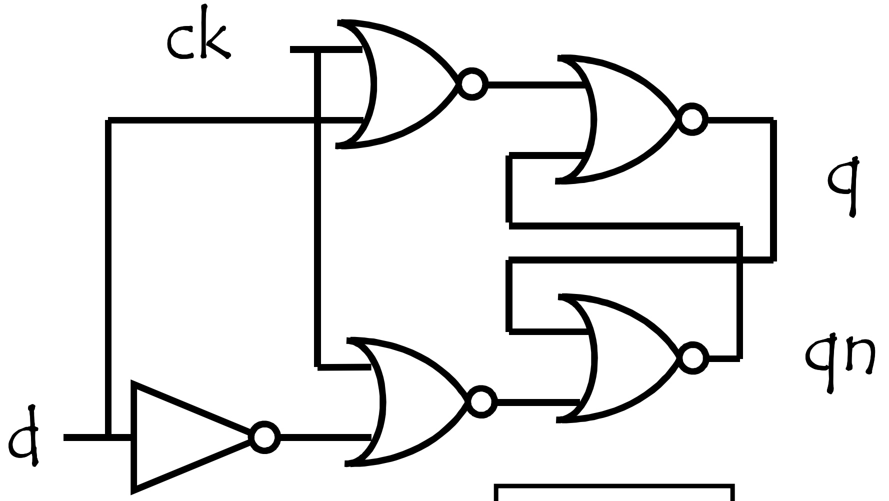
s	r	q	qn
0	1	0	1
1	0	1	0
1	1	0	0
0	0	q	qn

RS flip flop

CMOS Circuits

Synchronous Memory :

Write a data d when the clock ck = 0



d latch

s	r	q	qn
0	1	0	1
1	0	1	0
1	1	0	0
0	0	q	qn

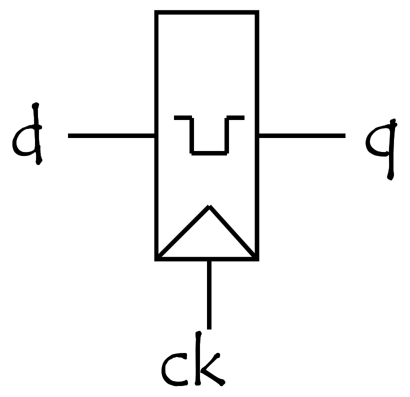
if $\overline{ck}.d = 1$ s = 1

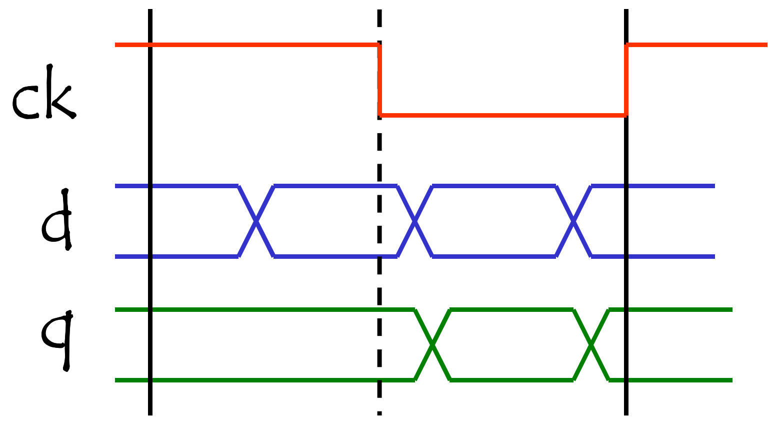
if $\overline{ck}.\overline{d} = 1$ r = 1

CMOS Circuits

Synchronous Memory :

Write a data d when the clock ck = 0

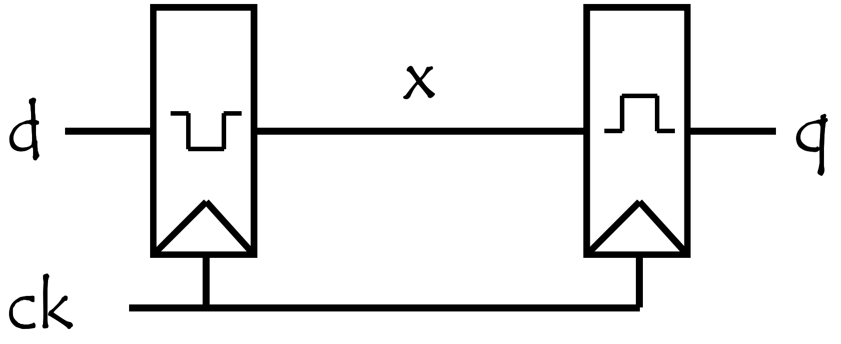


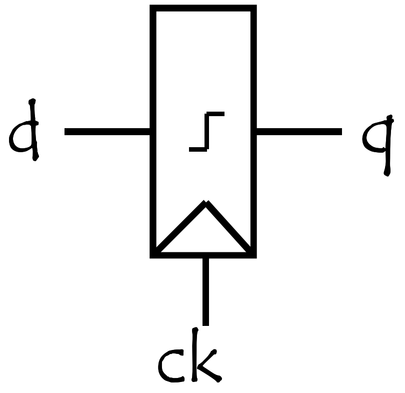


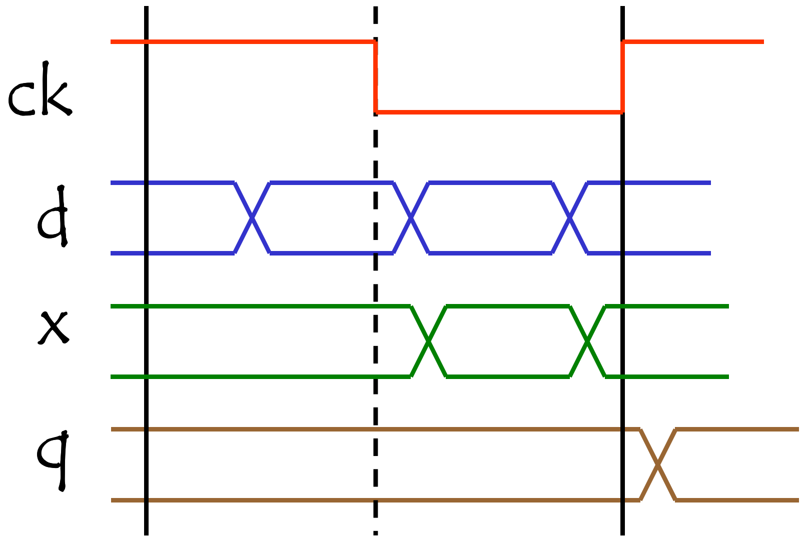
CMOS Circuits

Synchronous Memory :

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



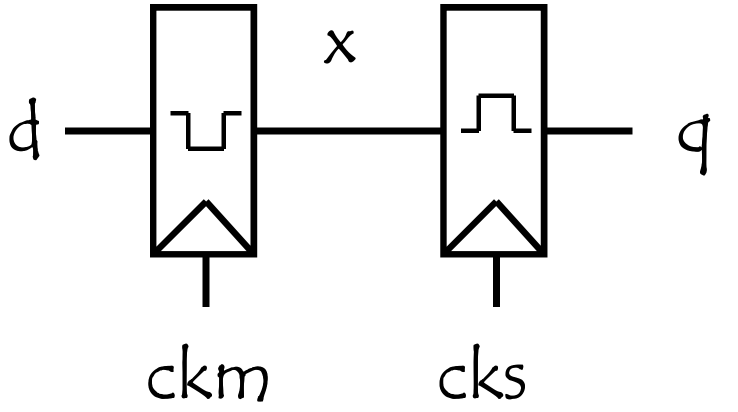


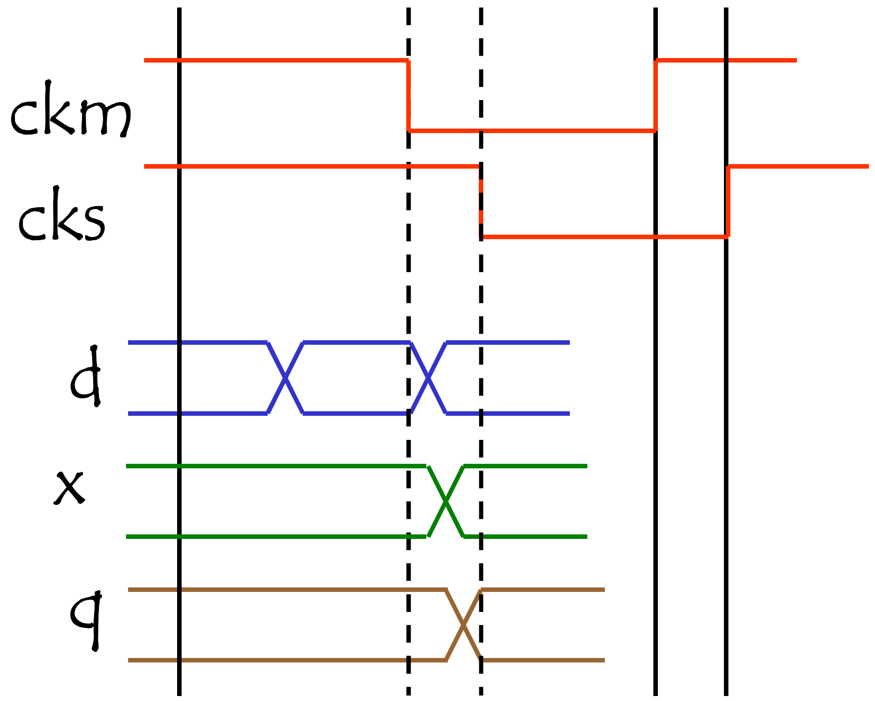


CMOS Circuits

Synchronous Memory :

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

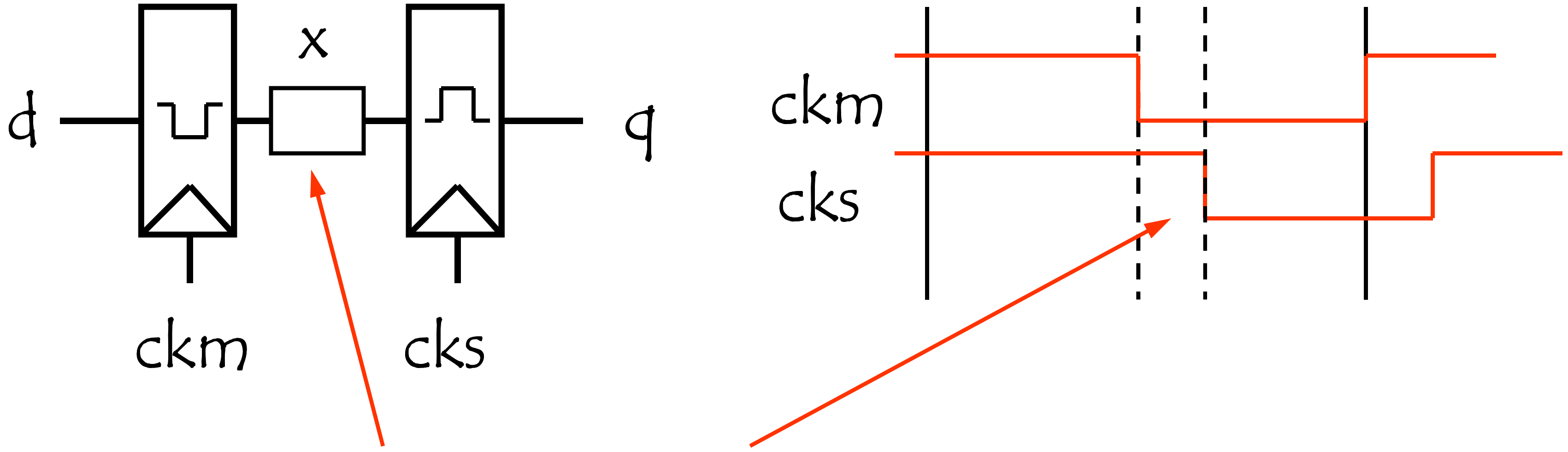




CMOS Circuits

Synchronous Memory :

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

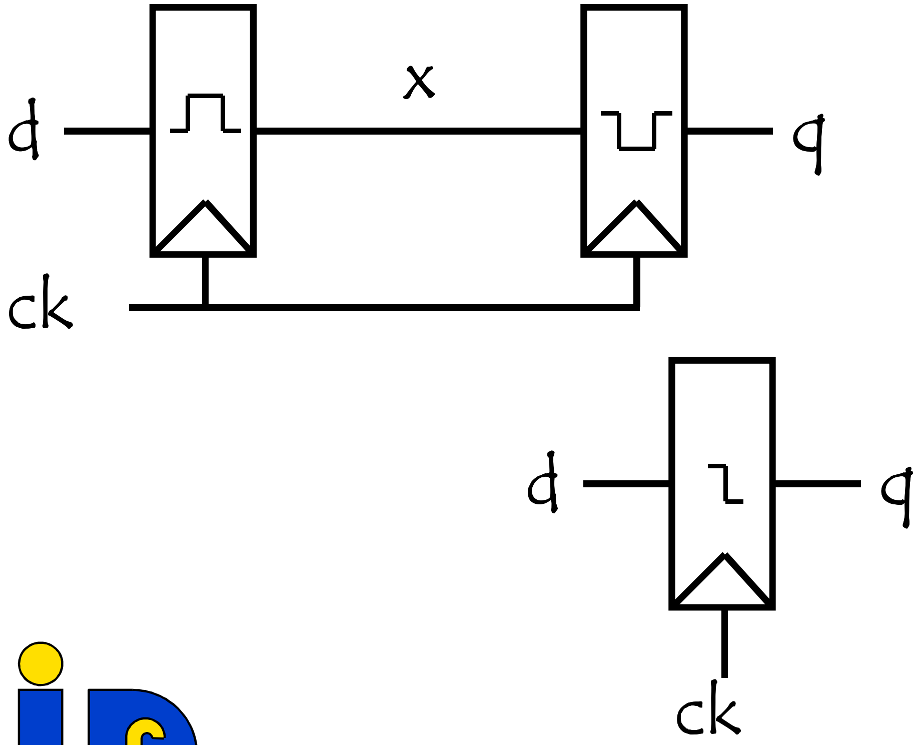


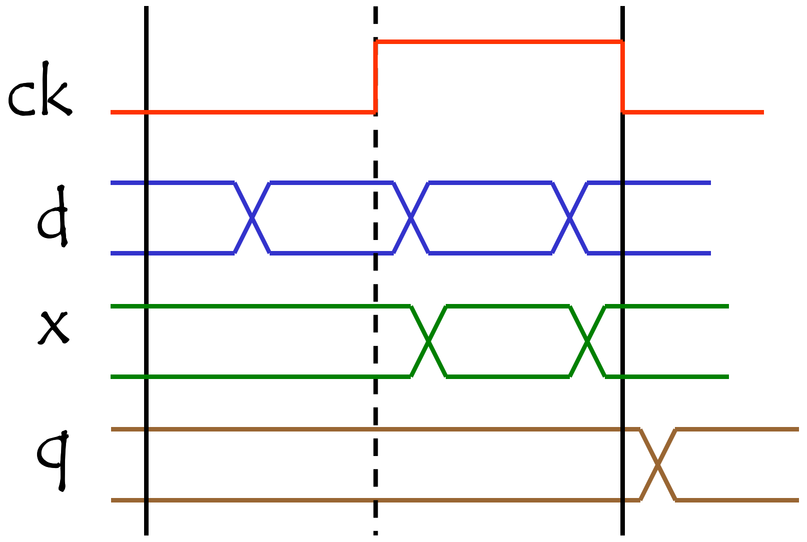
delay < skew

CMOS Circuits

Synchronous Memory :

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

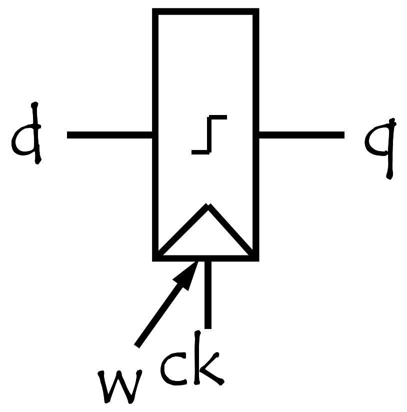


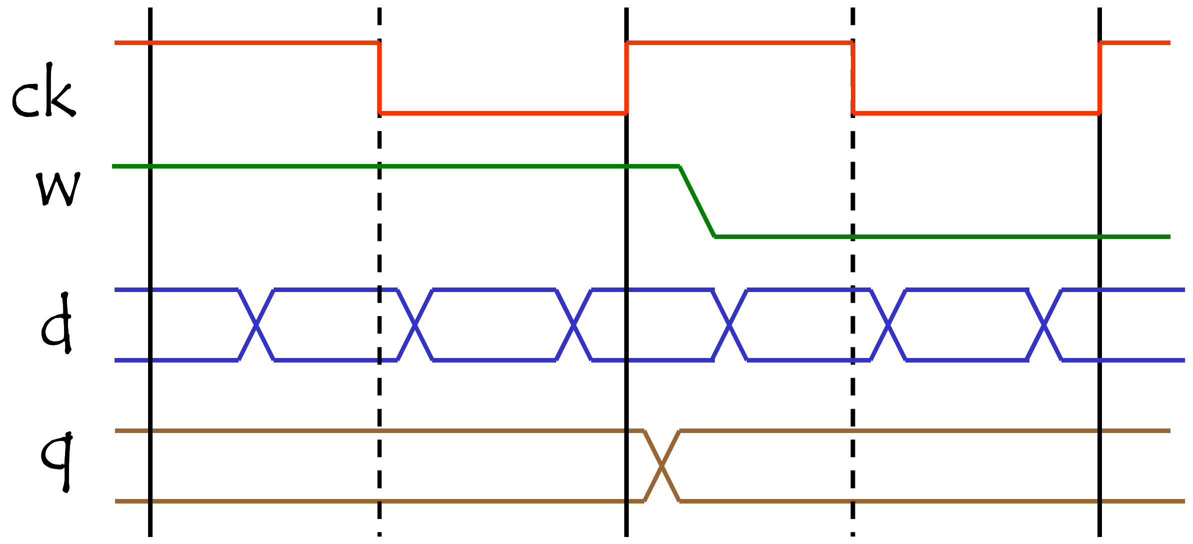


CMOS Circuits

Synchronous Memory :

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

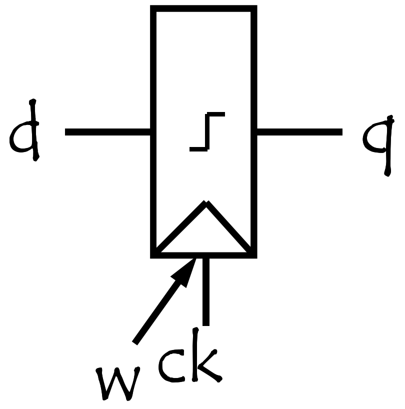


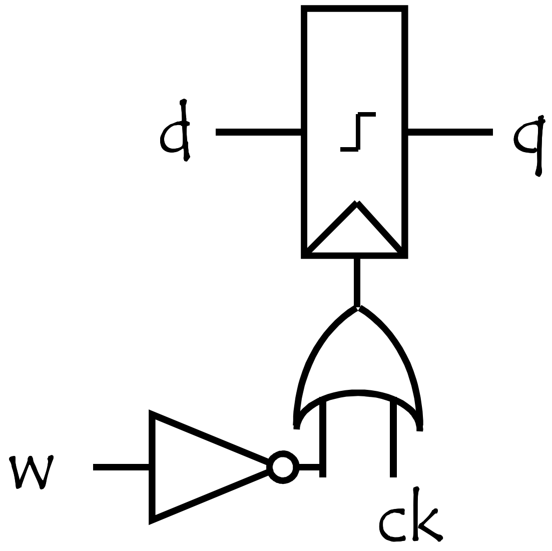


CMOS Circuits

Synchronous Memory :

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

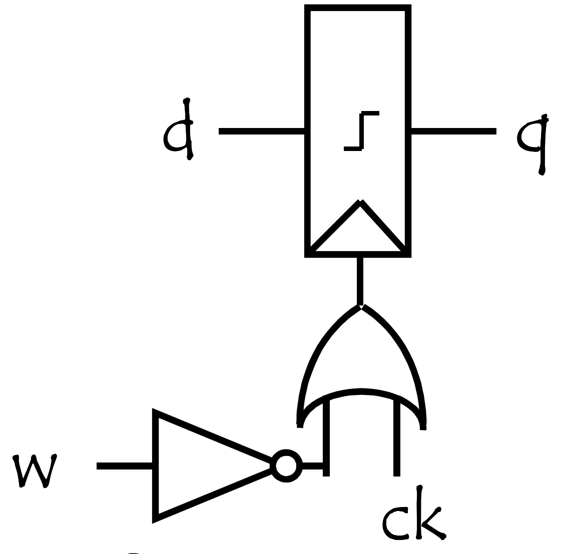


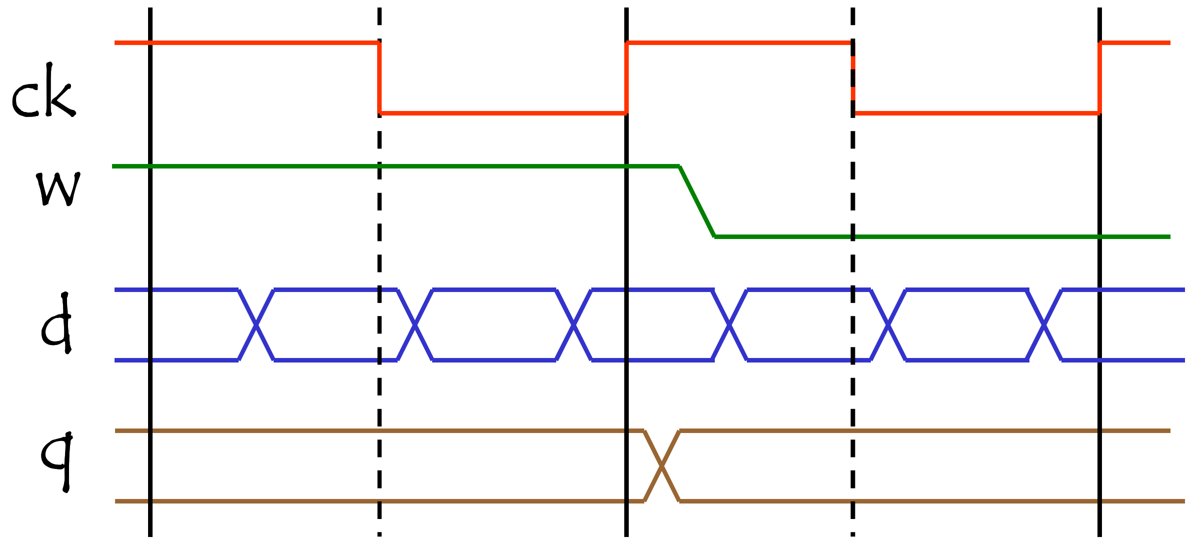


CMOS Circuits

Synchronous Memory :

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

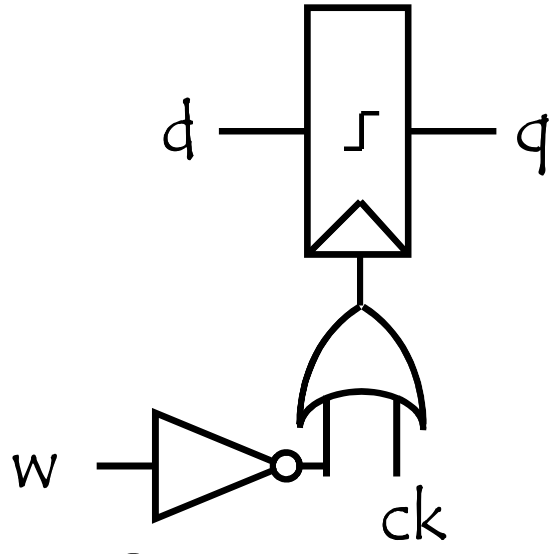


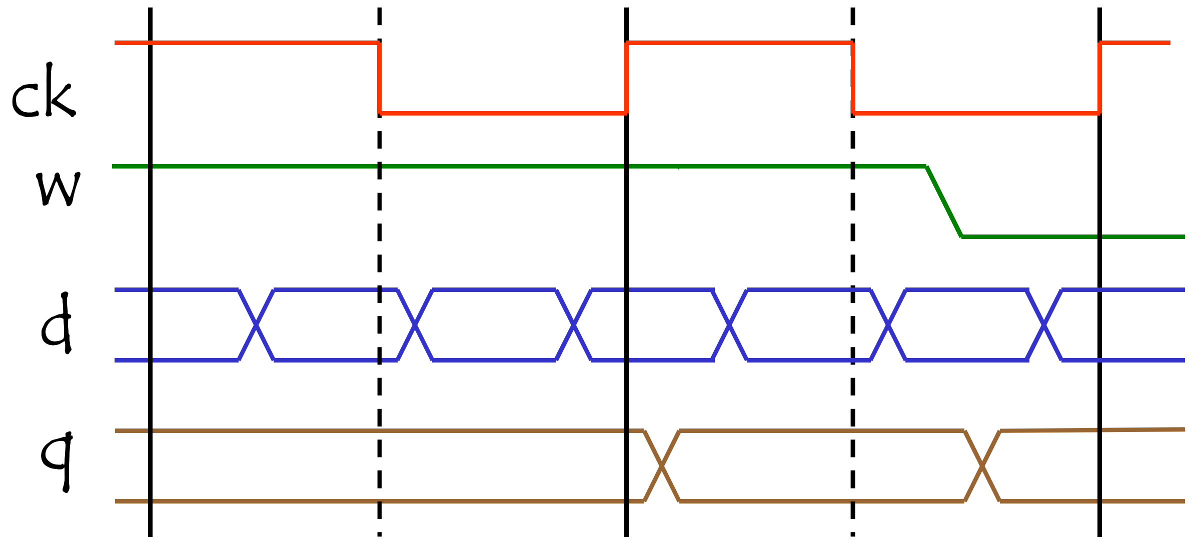


CMOS Circuits

Synchronous Memory :

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

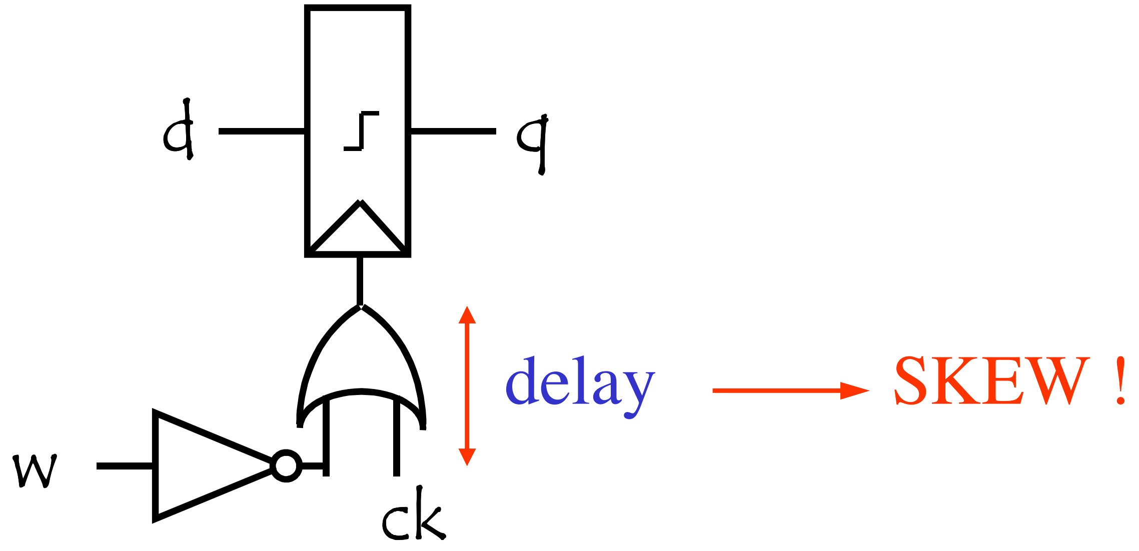




CMOS Circuits

Synchronous Memory :

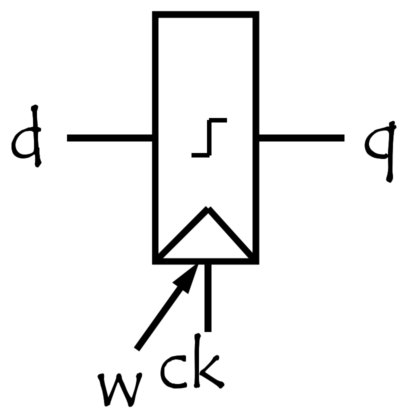
Write a data d on the rising edge of the clock ck
when a condition is true (write enable)

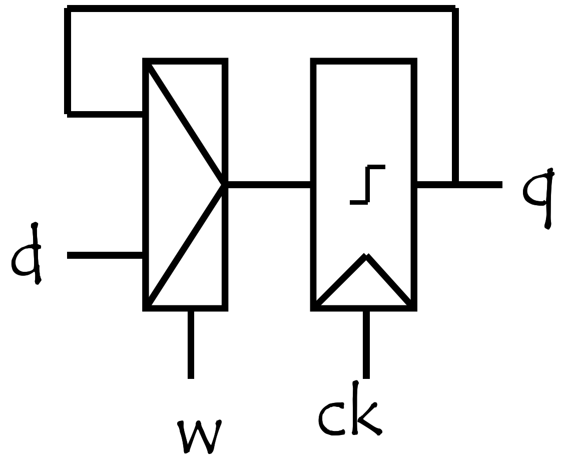


CMOS Circuits

Synchronous Memory :

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

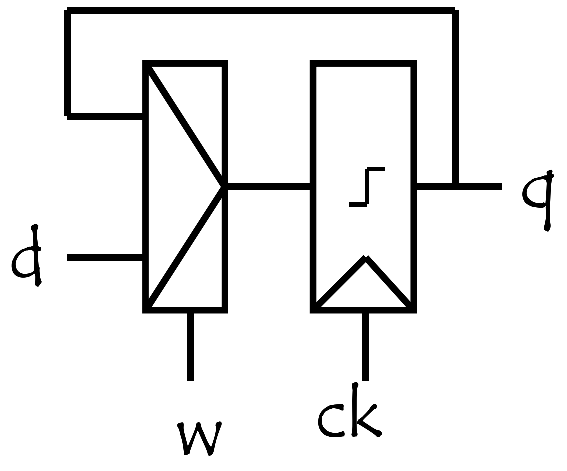




CMOS Circuits

Synchronous Memory :

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



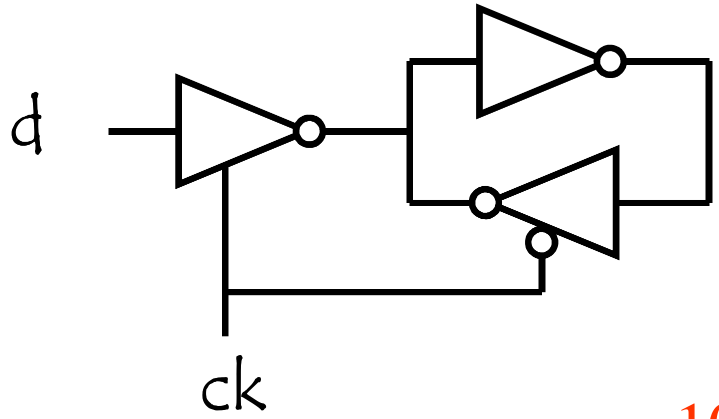
44 trs per register !!

CMOS Circuits

Memory : Latch

Hold a data (0 or 1)

Write a data (0 or 1)



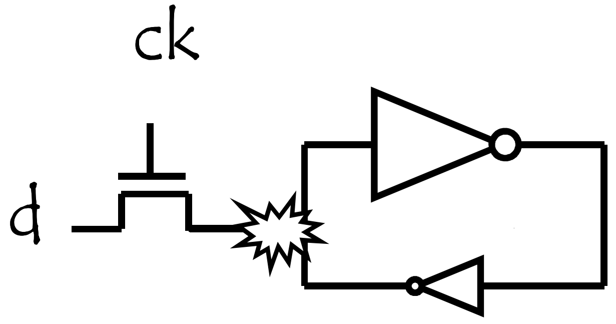
10 trs per latch

CMOS Circuits

Memory : Latch

Hold a data (0 or 1)

Write a data (0 or 1)



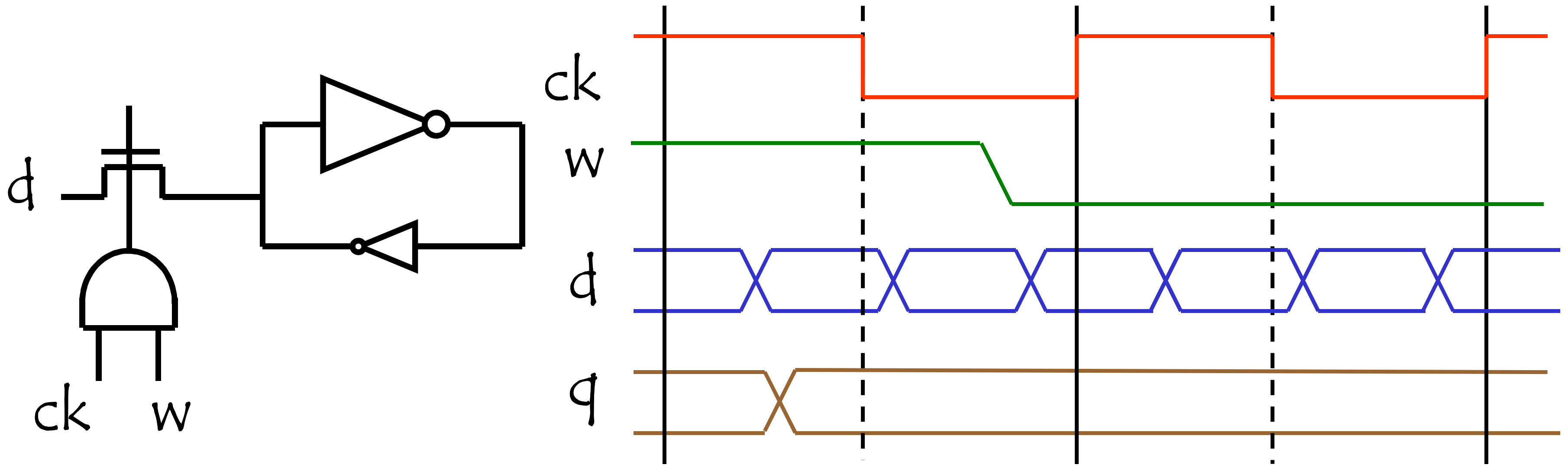
short circuit !

5 trs per latch

CMOS Circuits

Synchronous Memory :

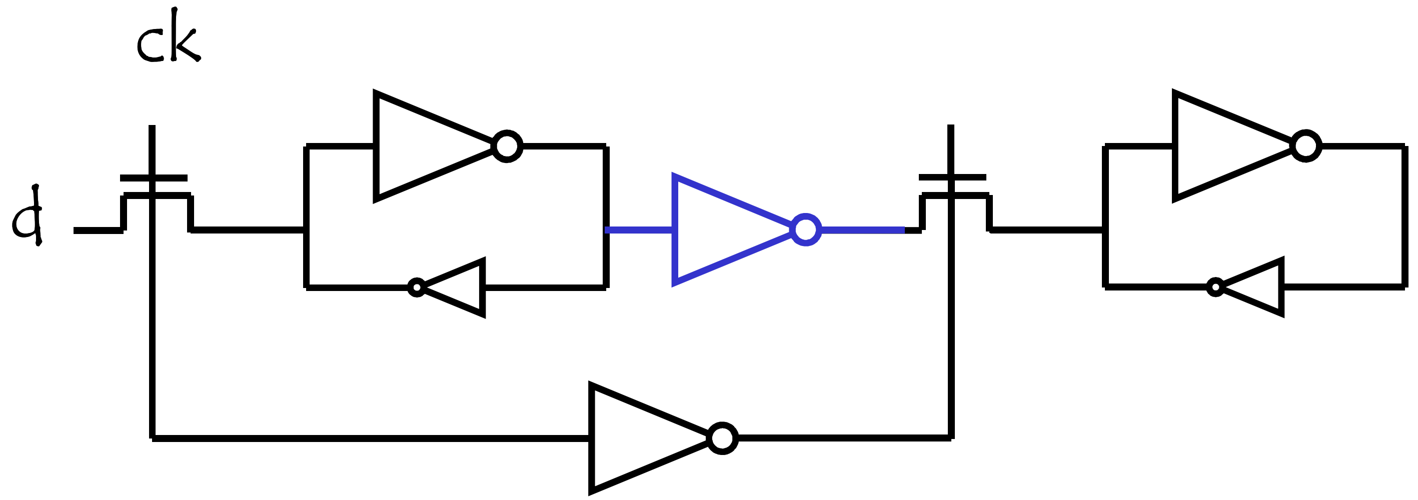
Write a data d when the clock ck = 1
when a condition is true (write enable)



CMOS Circuits

Synchronous Memory :

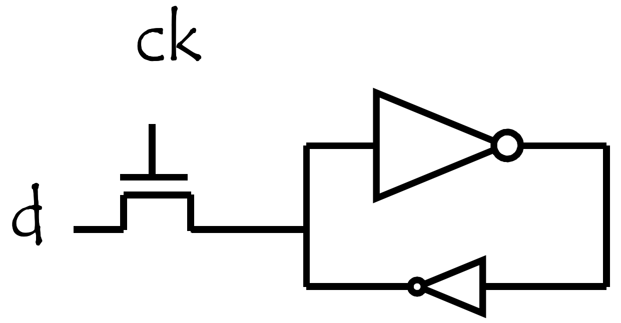
Write a data d on the falling edge of the clock ck

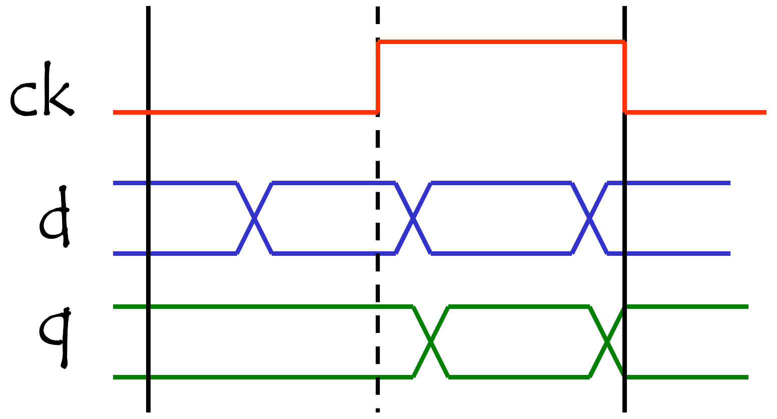


CMOS Circuits

Synchronous Memory :

Write a data d when the clock ck = 1

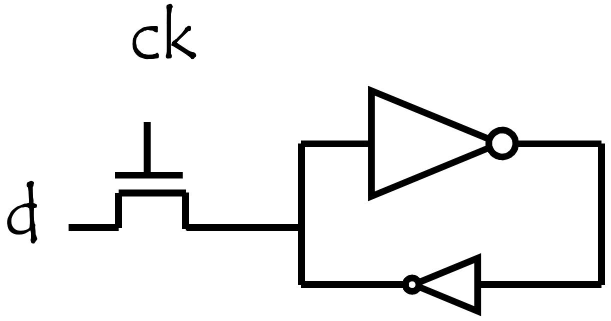


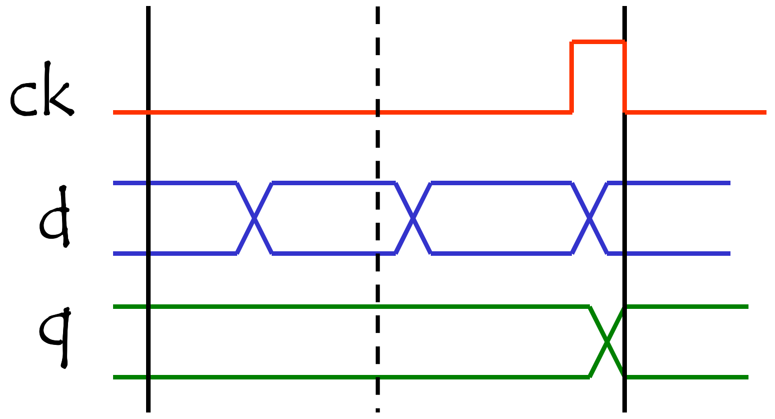


CMOS Circuits

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