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

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

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

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

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

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

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

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 ?

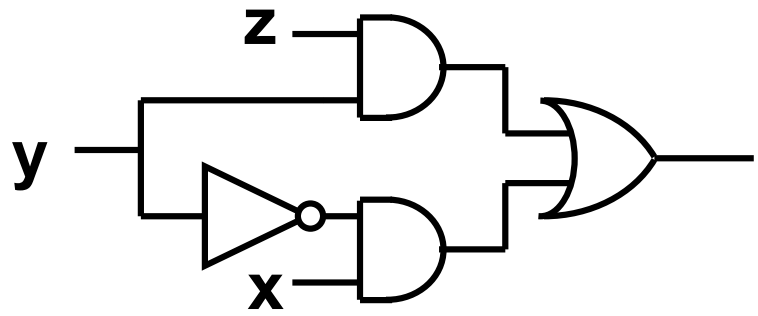
Cell library

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

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



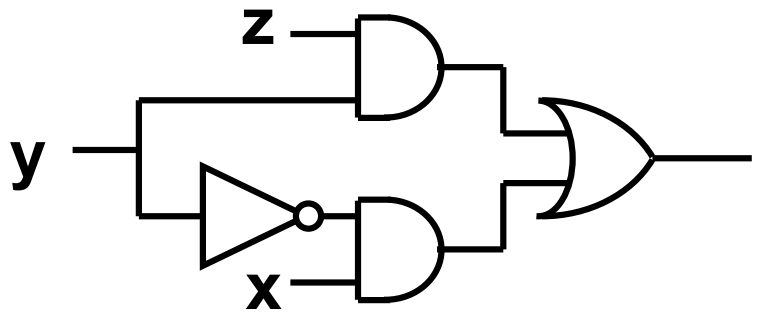
Non-inverting gates do NOT exist

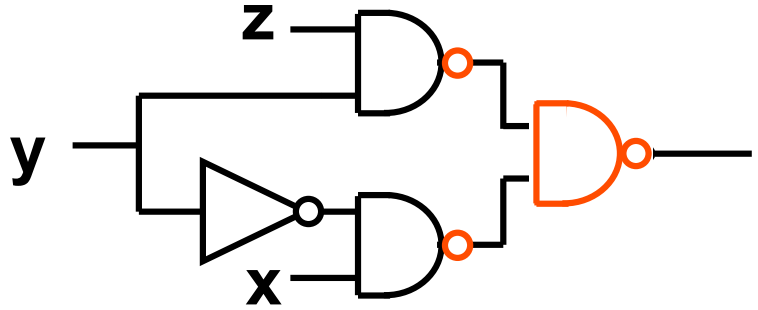
$f = x.\bar{y} + y.z$

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





CMOS Circuits

Example :

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

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

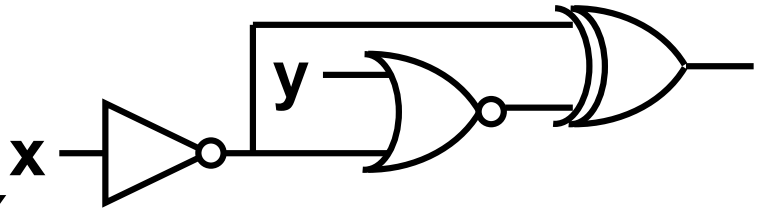
$$f = \overline{(x + y)}\,.\,\overline{x} + \overline{\overline{(x + y)}\,.\,x}$$

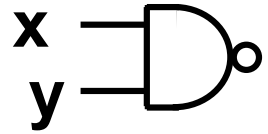
$$f = \overline{x} + \overline{y}.x + \overline{x}.y$$

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

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

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





CMOS Circuits

How to implement Boolean functions with a gate network ?

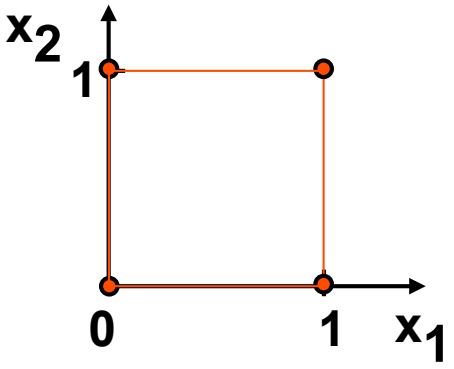
- Which expression ?

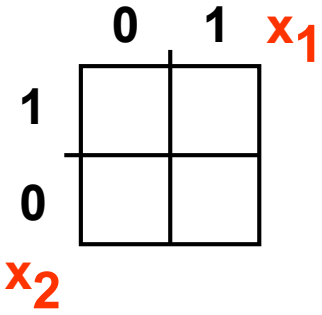
CMOS Circuits

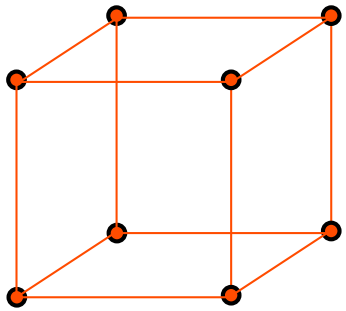
A function can be defined by its Truth table

- Karnaugh representation gives a minimal expression

Representation of the function in a space of dimension **n**



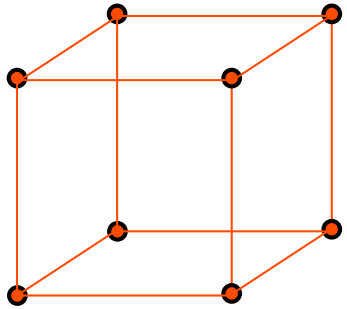


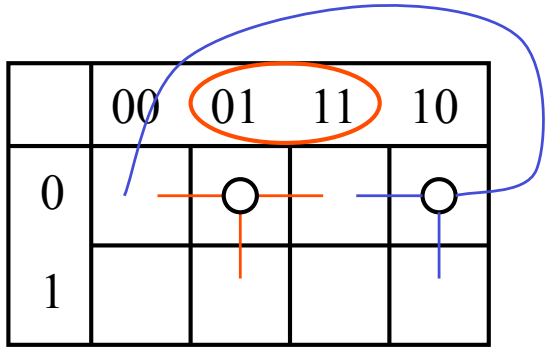


Representation of vectors' adjacency

CMOS Circuits

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





vectors' adjacency

CMOS Circuits

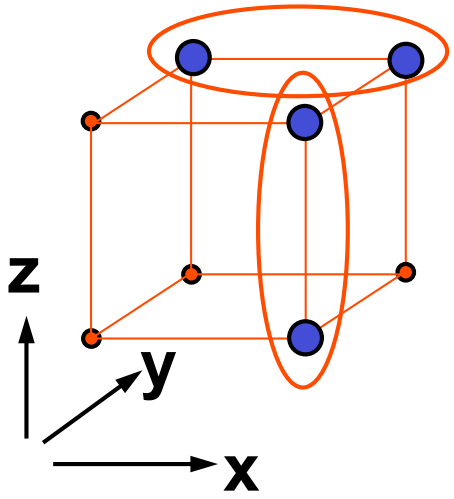
Example :

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
0	0	0	0	1
1	0	1	1	1

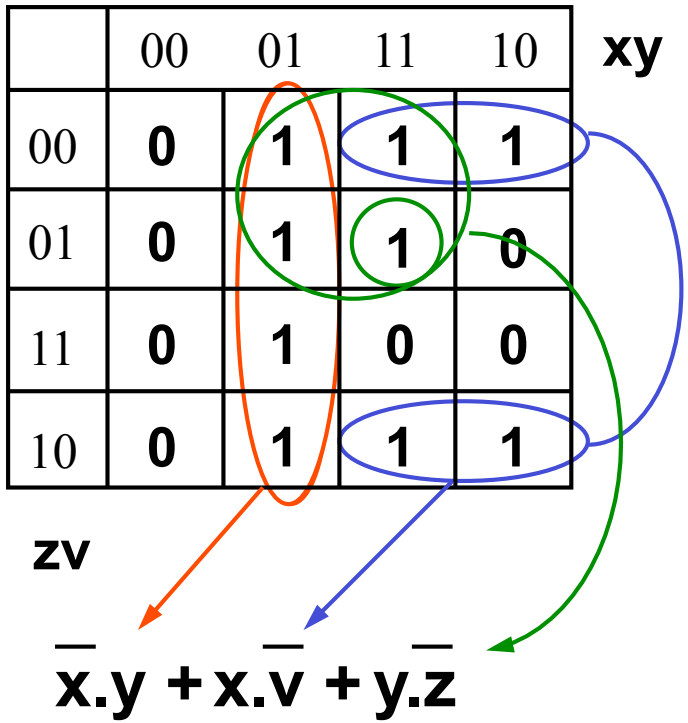
xy

z

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

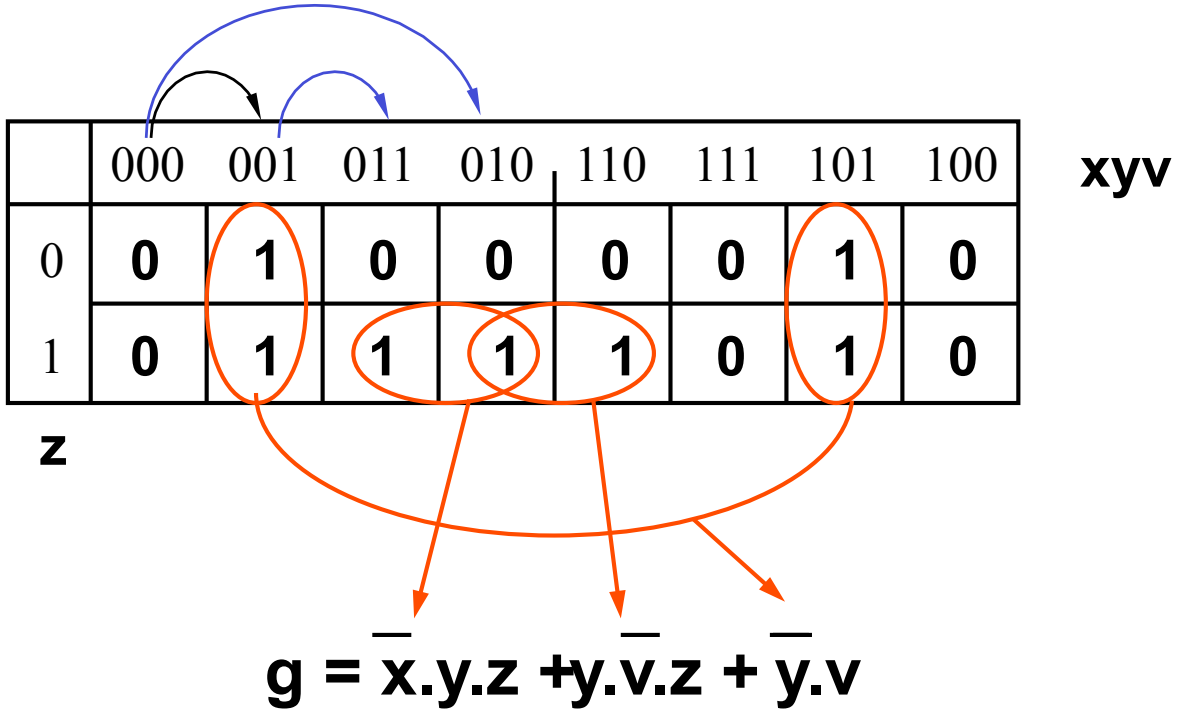
CMOS Circuits

Example :



CMOS Circuits

Example :



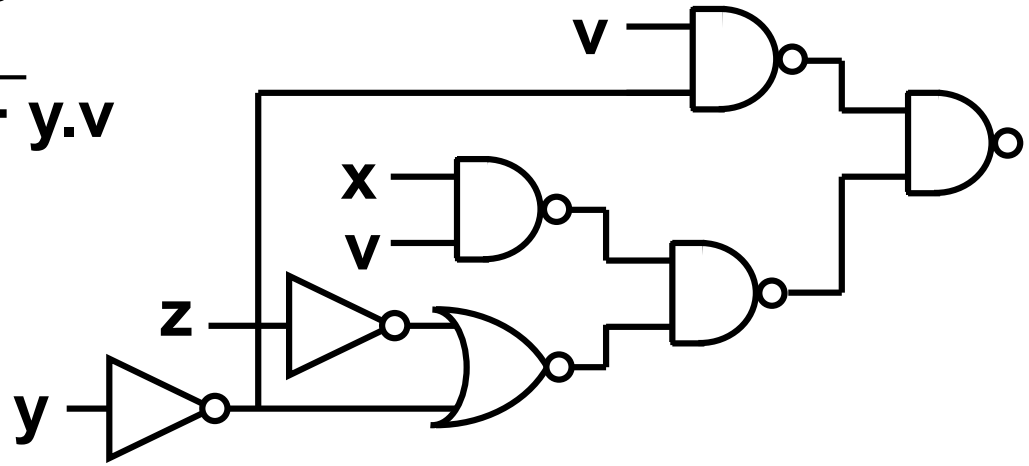
CMOS Circuits

Example :

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

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

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



CMOS Circuits

How to implement Boolean functions with a gate network ?

- Local optimization using Karnaugh tables

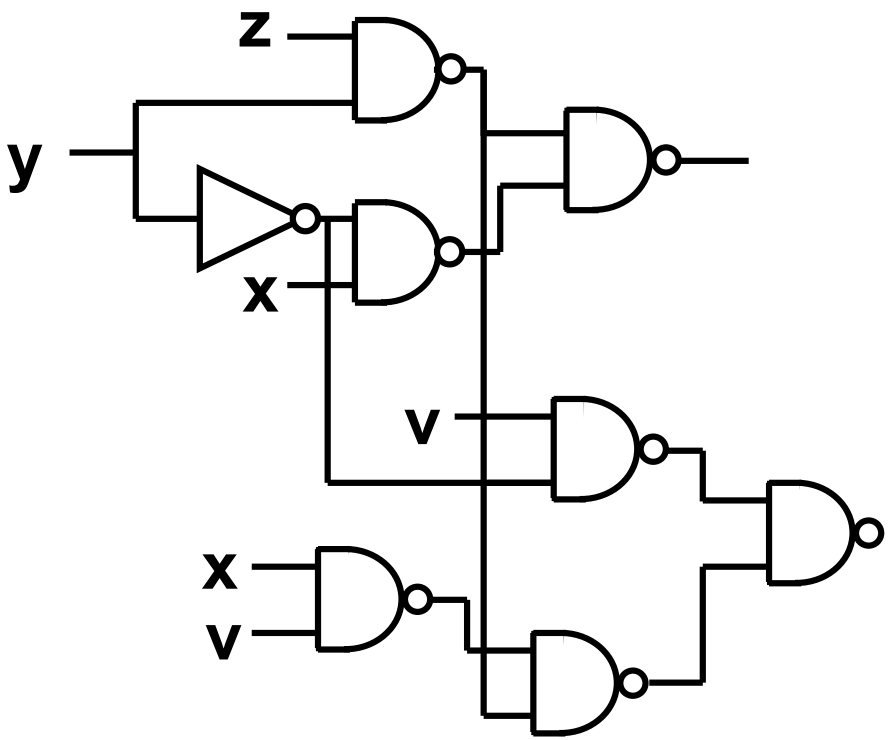
A design includes several Boolean functions

CMOS Circuits

Example :

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

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



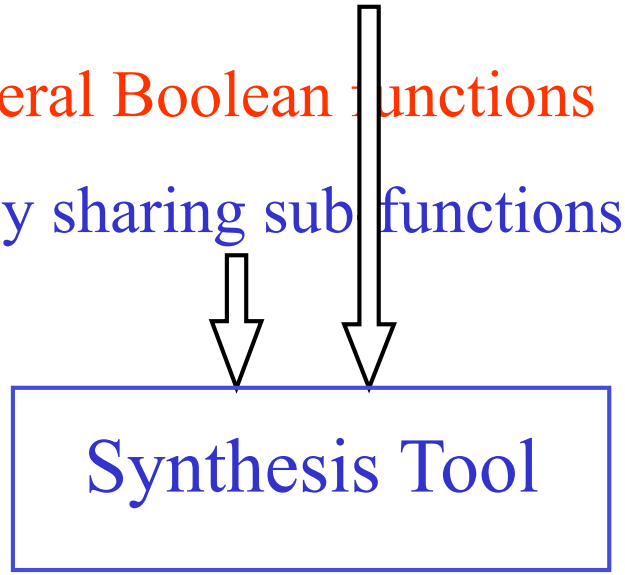
CMOS Circuits

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



CMOS Circuits

- Combinational logic
- Sequential logic

CMOS Circuits

- Combinational logic

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

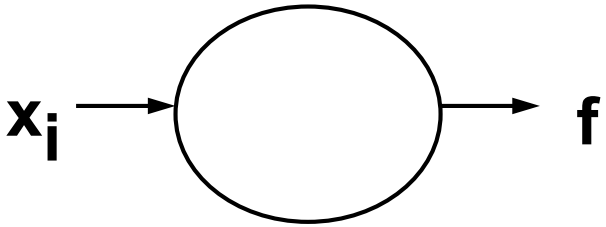
- Sequential logic

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

Notion of memory

CMOS Circuits

- Sequential logic

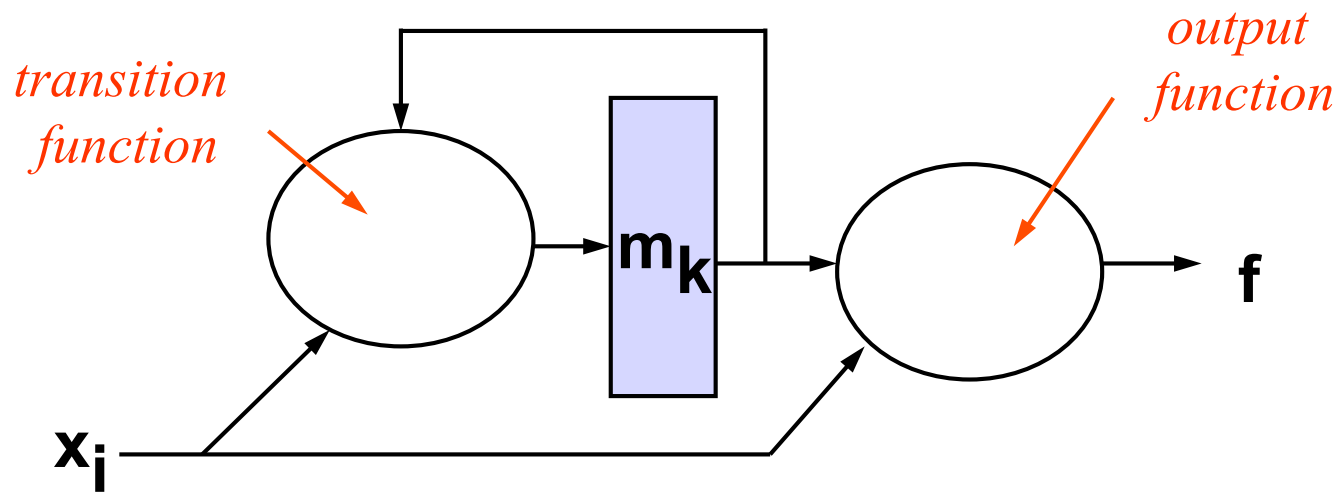


$$f\,(x_1, \ldots, x_i, \ldots, x_n, m_1, \ldots, m_k, \ldots, m_p)$$

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

CMOS Circuits

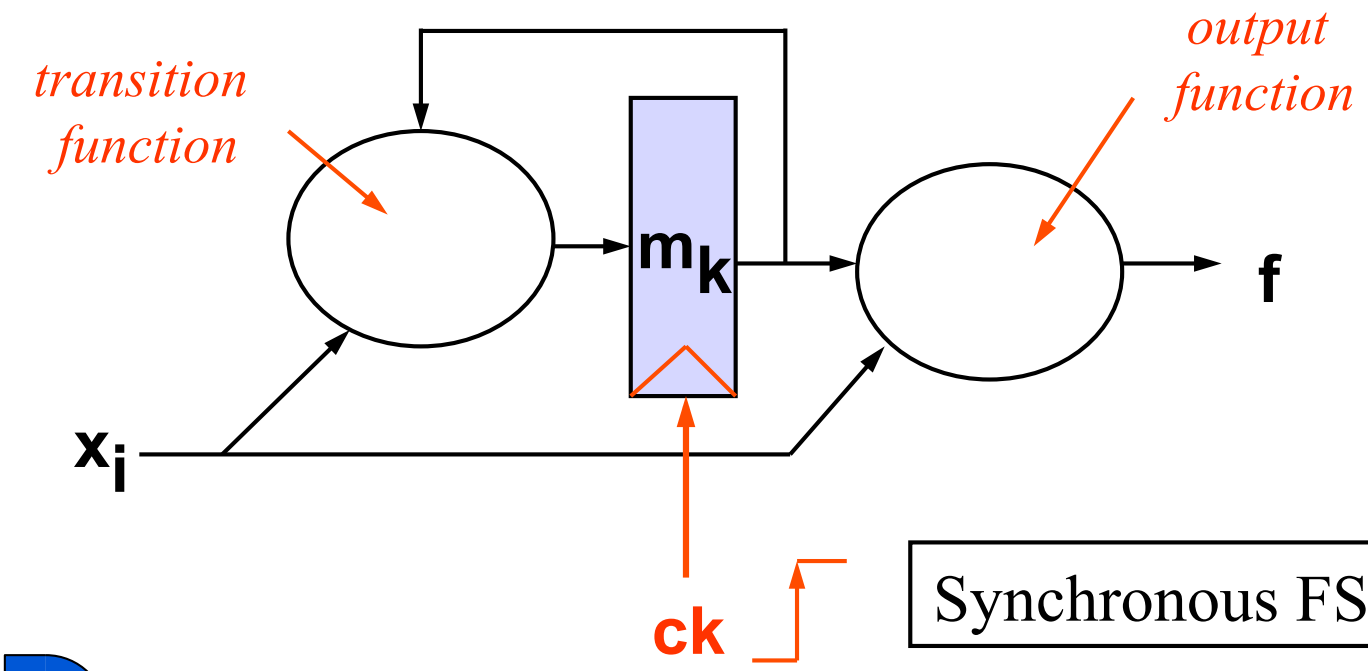
- Sequential logic



Finite State Machine (FSM)

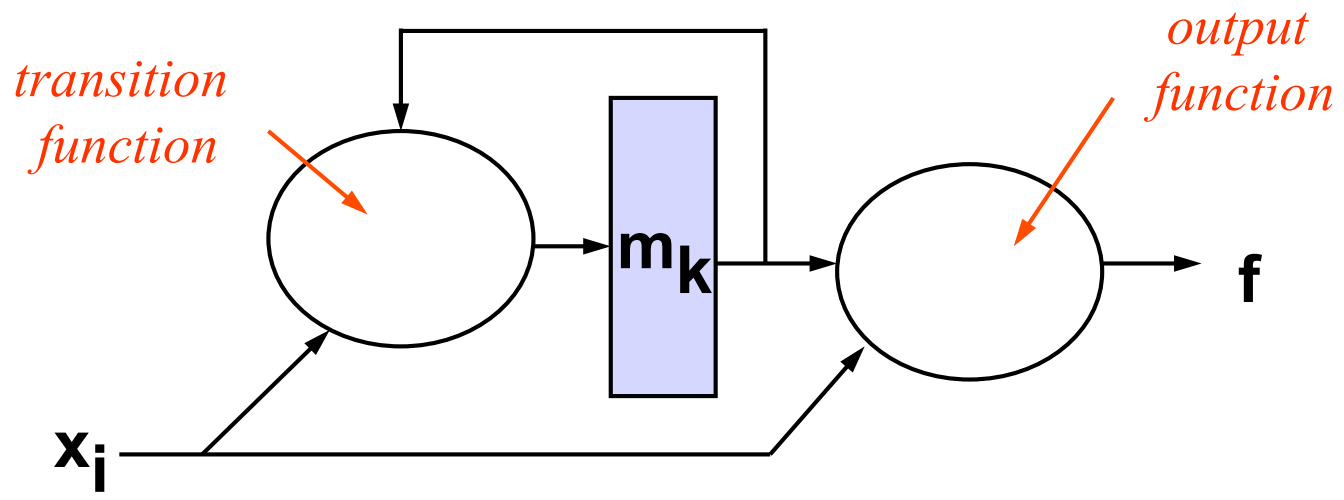
CMOS Circuits

- Finite State Machine



CMOS Circuits

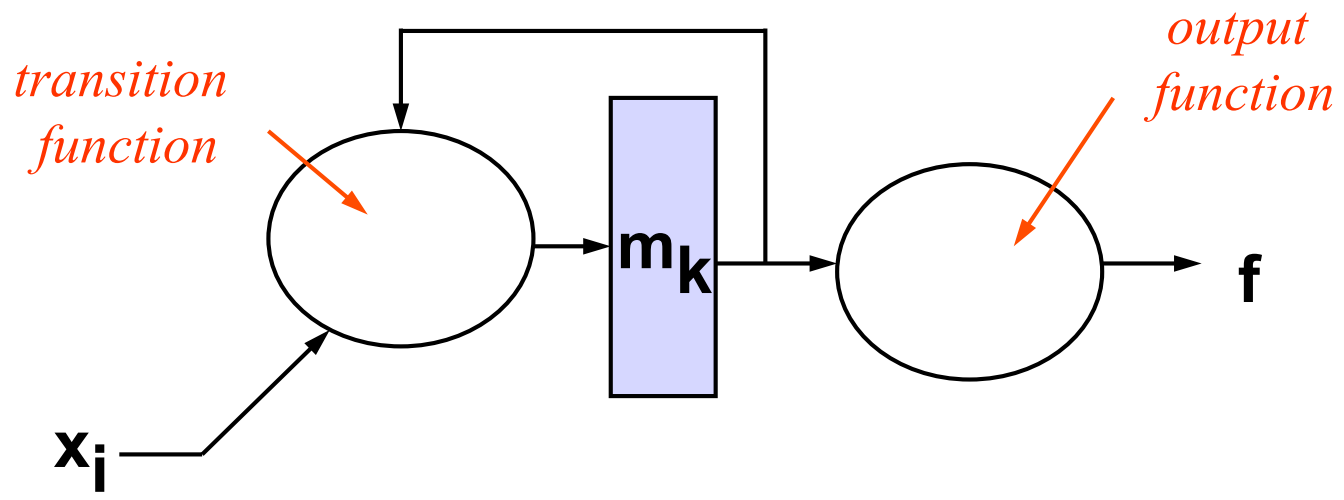
- Finite State Machine



Mealy FSM

CMOS Circuits

- Finite State Machine



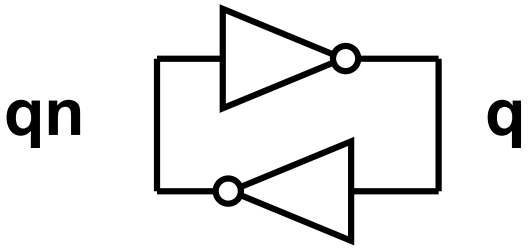
Moore FSM

CMOS Circuits

Memory :

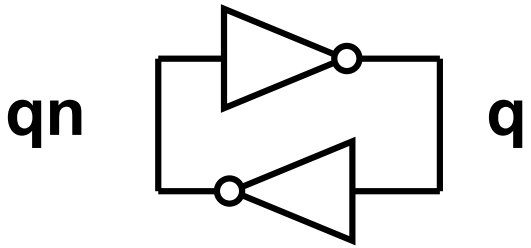
Hold a data (0 or 1)

Write a data (0 or 1)



CMOS Circuits

Memory :



$q = \overline{qn}$ $\quad qn = \overline{q}$

$f = \overline{\overline{q}} = q$ $\quad f(q) = q$

$$\frac{\partial f^{+}}{\partial q} = f_{q1} \cdot \overline{f_{q0}} = 1$$

$$\frac{\partial f^{-}}{\partial q} = \overline{f_{q1}} \cdot f_{q0} = 0$$

q depends always on itself in a positive way

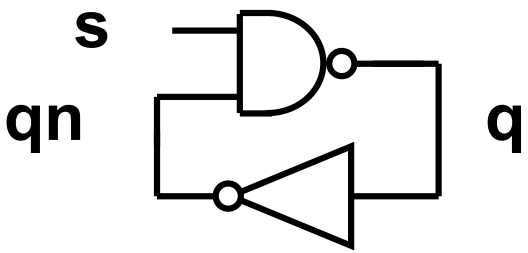
this dependency cannot be disabled

CMOS Circuits

Memory :

Hold a data (0 or 1)

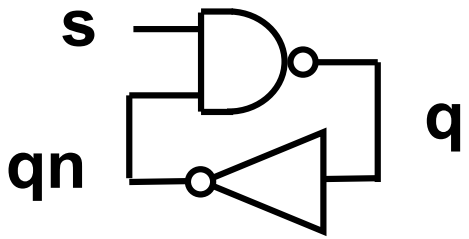
Write a data (0 or 1)



s	q	qn
0	1	0
1	q	qn

CMOS Circuits

Memory :



$q = \overline{s.qn}$ $\quad qn = \overline{q}$

$f = q = \overline{s . \overline{(q)}}$

$f = \overline{s} + q$ $\quad f = q + \overline{q} . s$

$$\frac{\partial f^{+}}{\partial q} = 1 . s = s \qquad \frac{\partial f^{-}}{\partial q} = \overline{s} . 0 = 0$$

if **s = 1** **q** is a positive function of itself

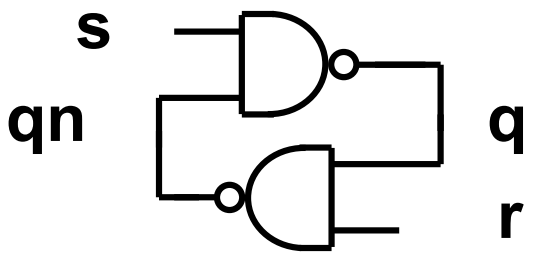
if **s = 0** **q** is a combinational function

CMOS Circuits

Memory :

Hold a data (0 or 1)

Write a data (0 or 1)

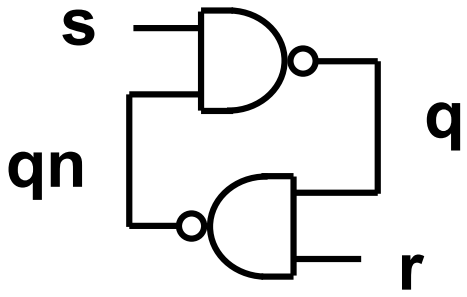


s	r	q	qn
0	1	1	0
1	0	0	1
1	1	q	qn
0	0	1	1

RS flip flop

CMOS Circuits

Memory :



$$q = \overline{s.qn} \qquad qn = \overline{r.q}$$

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

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

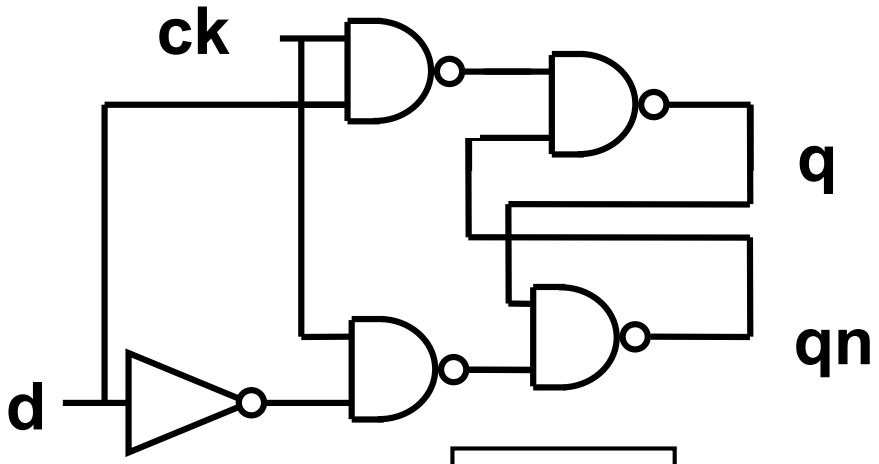
$$f = q \cdot (\overline{r+s}) + \overline{q} \cdot s$$

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

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

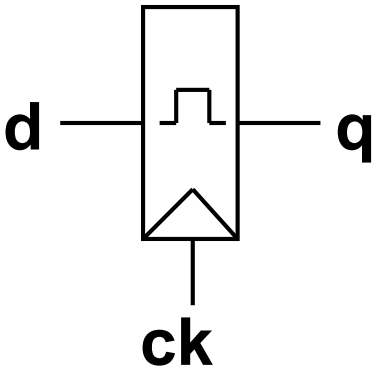
if **ck.d = 1** **s = 0**

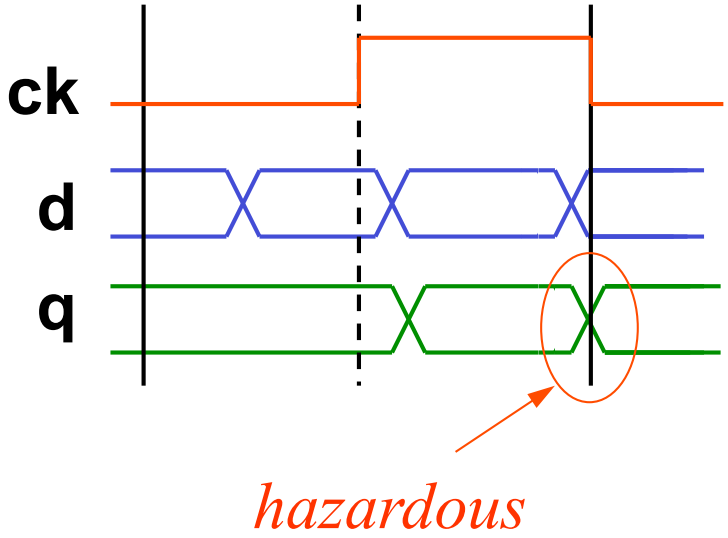
if **ck.$\overline{d}$ = 1** **r = 0**

CMOS Circuits

Synchronous Memory :

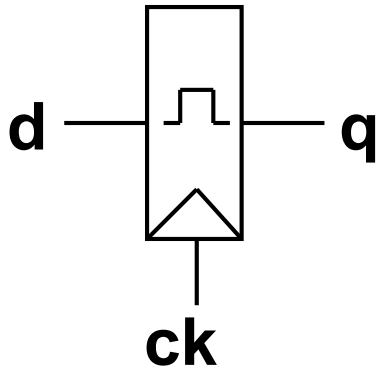
Write a data **d** when the clock **ck = 1**

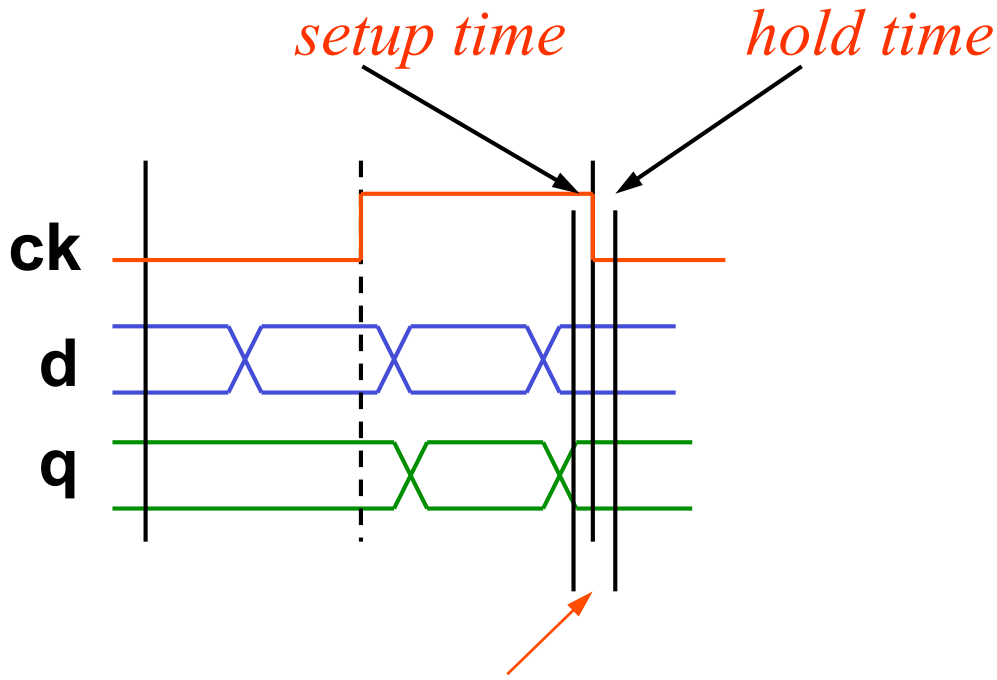




CMOS Circuits

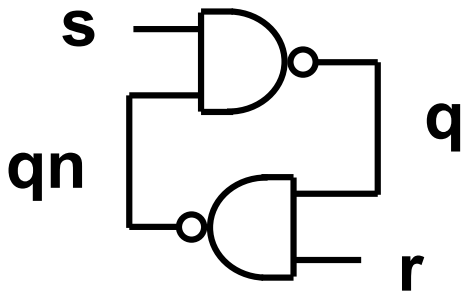
Synchronous Memory :

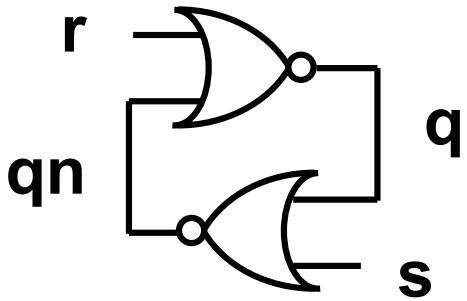




CMOS Circuits

Memory :





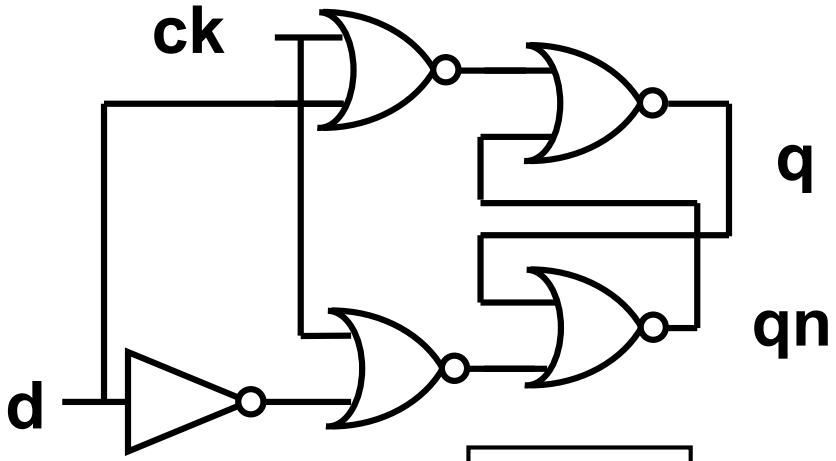
s	r	q	qn
0	1	0	1
1	0	1	0
1	1	0	0
0	0	q	qn

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

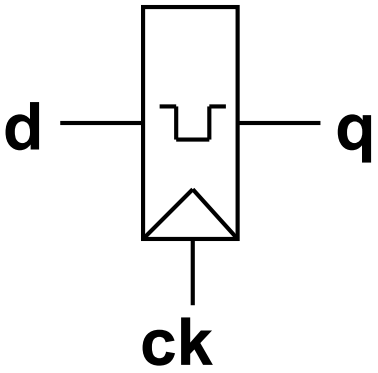
if $\overline{ck}.d = 1$ s = 1

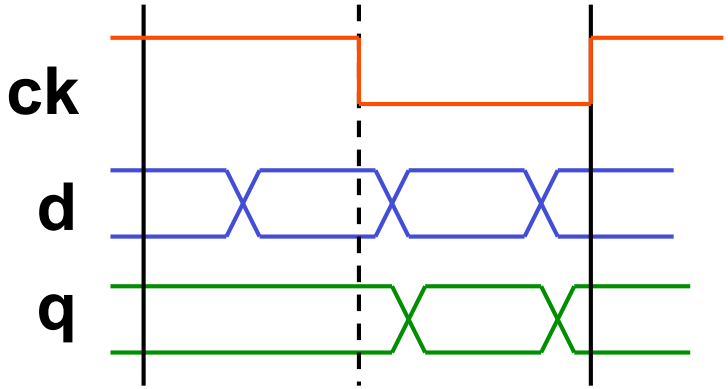
if $\overline{ck}.\overline{d} = 1$ r = 1

CMOS Circuits

Synchronous Memory :

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

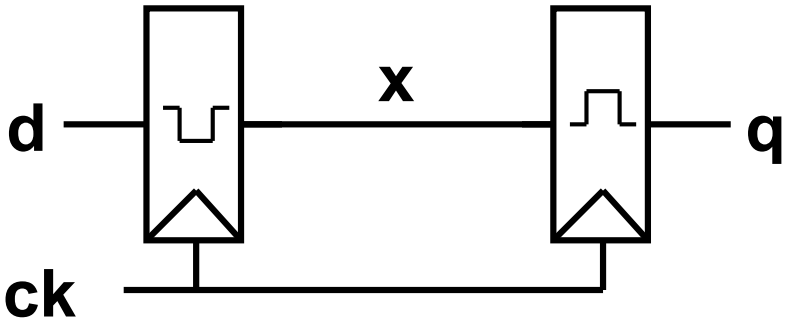


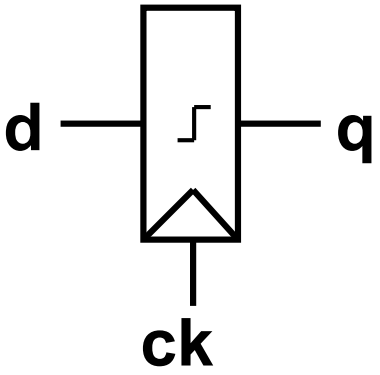


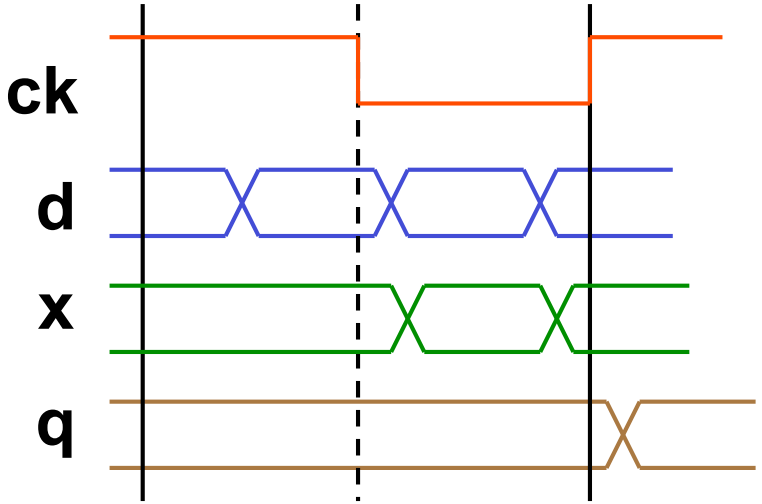
CMOS Circuits

Synchronous Memory :

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



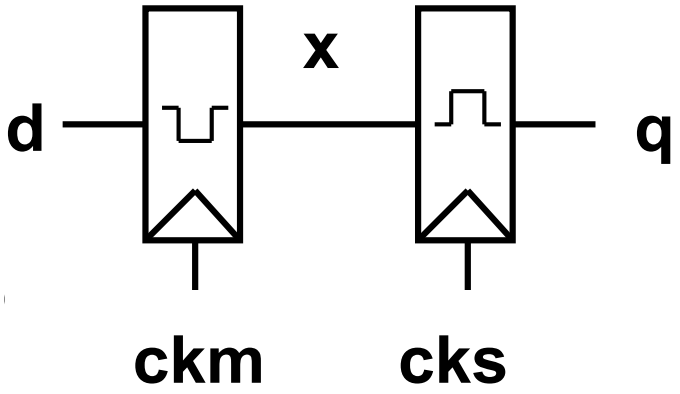


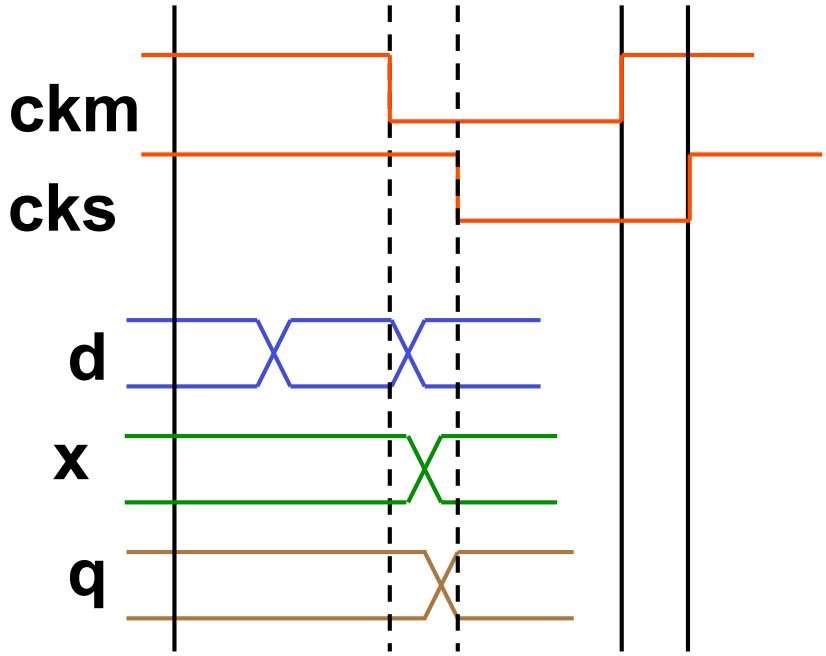


CMOS Circuits

Synchronous Memory :

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

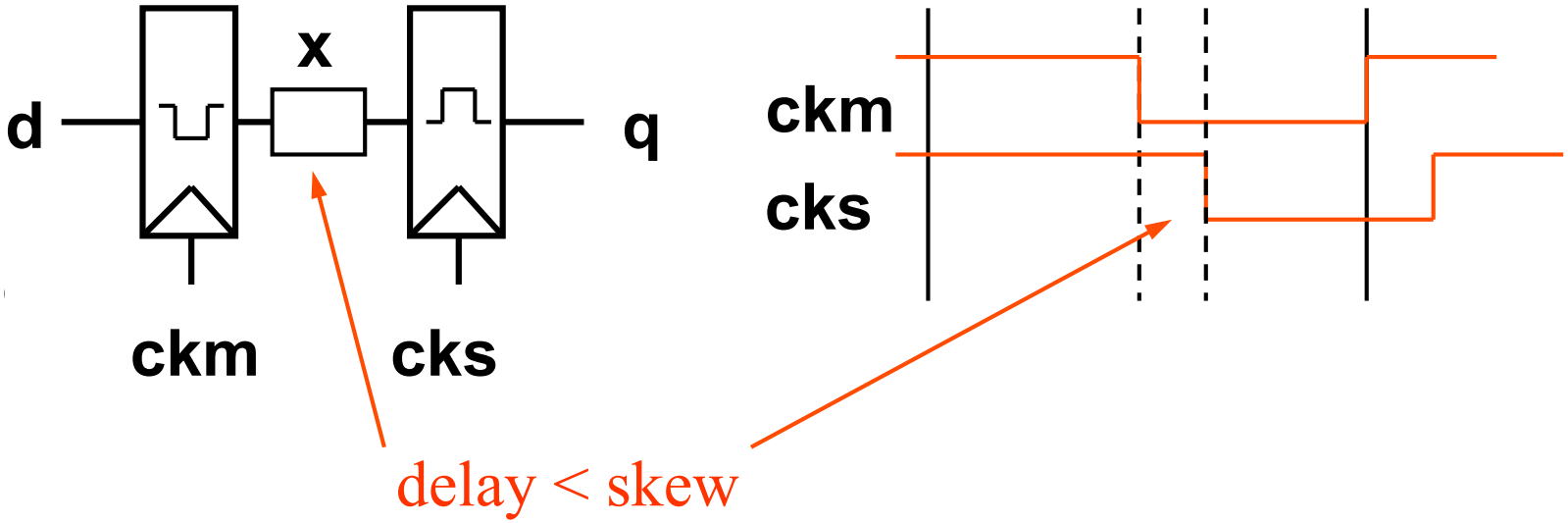




CMOS Circuits

Synchronous Memory :

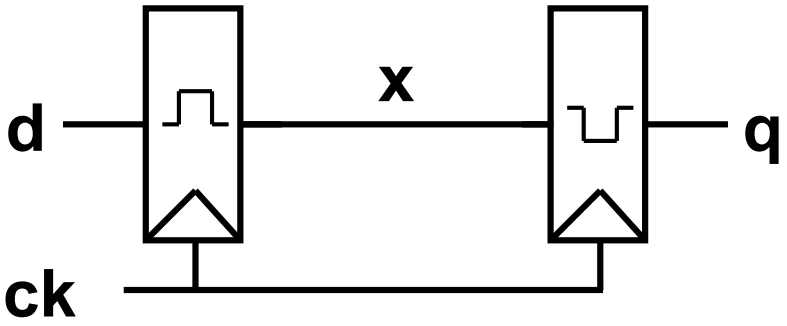
Write a data **d** on the rising edge of the clock **ck**

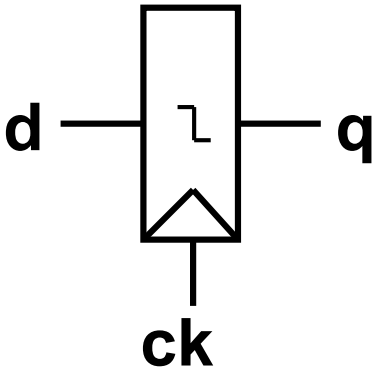


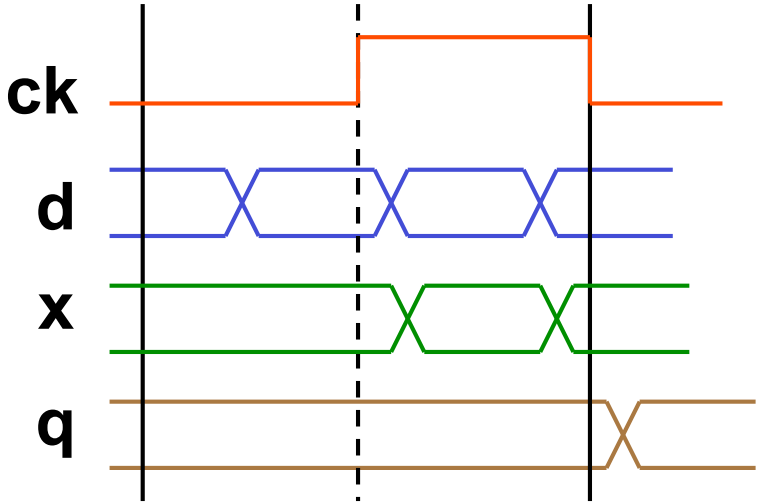
CMOS Circuits

Synchronous Memory :

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



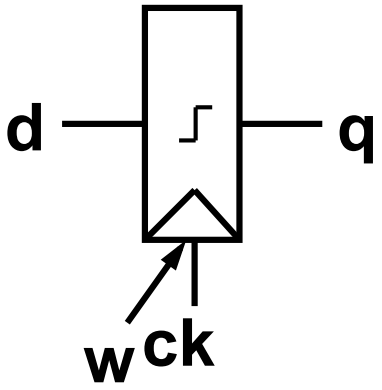


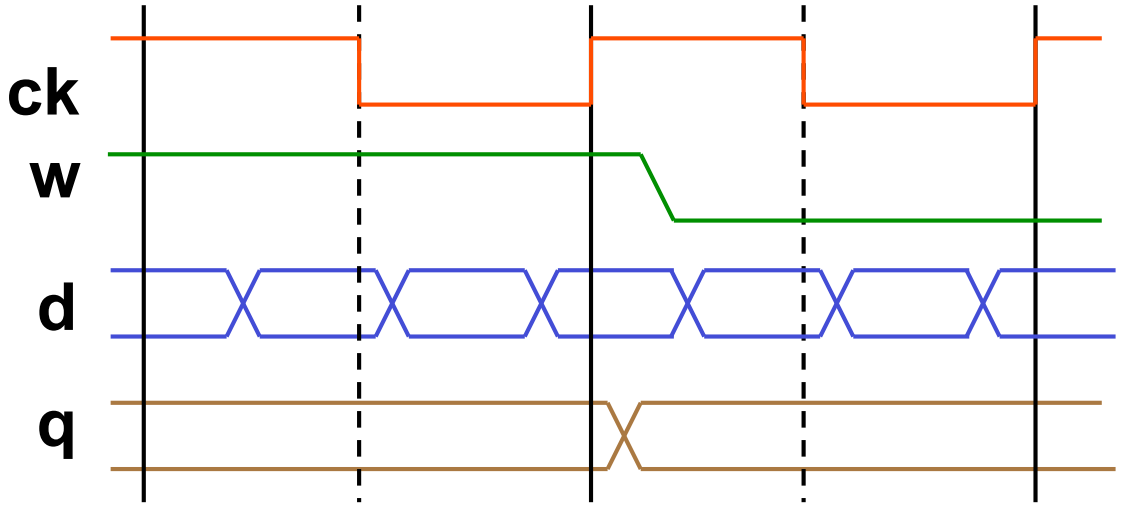


CMOS Circuits

Synchronous Memory :

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

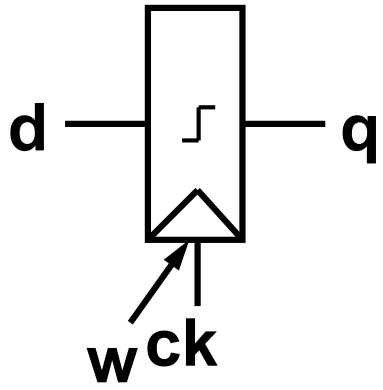


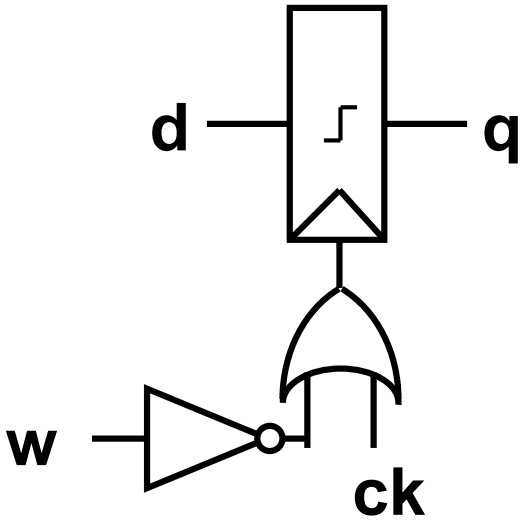


CMOS Circuits

Synchronous Memory :

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

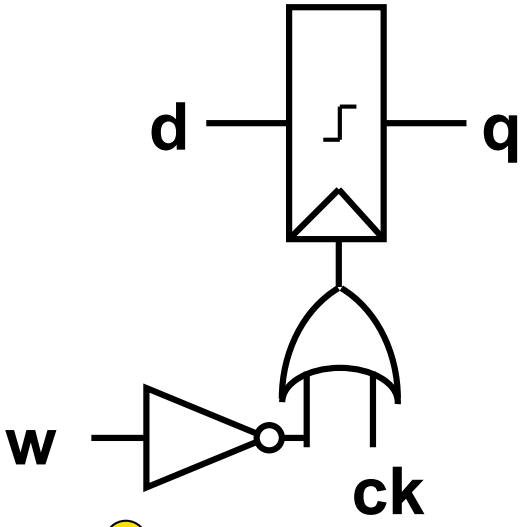


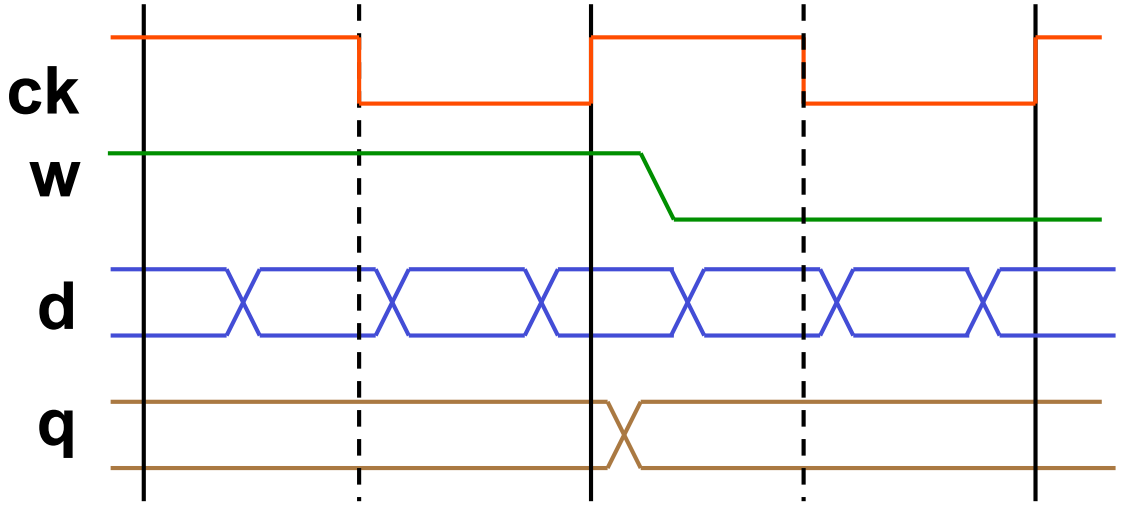


CMOS Circuits

Synchronous Memory :

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

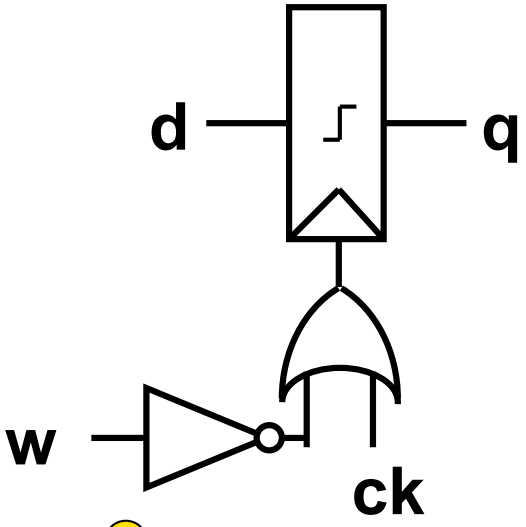


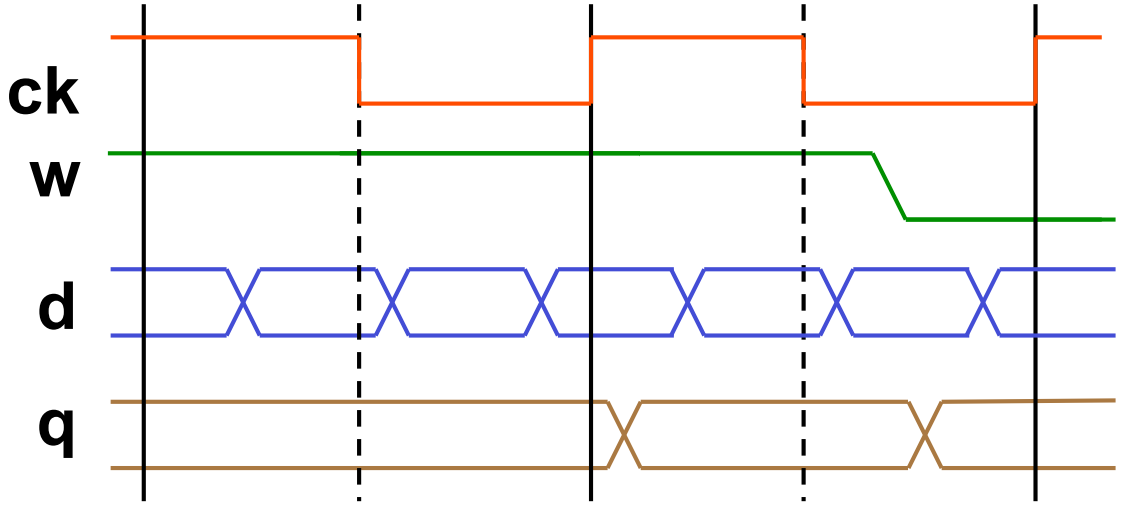


CMOS Circuits

Synchronous Memory :

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

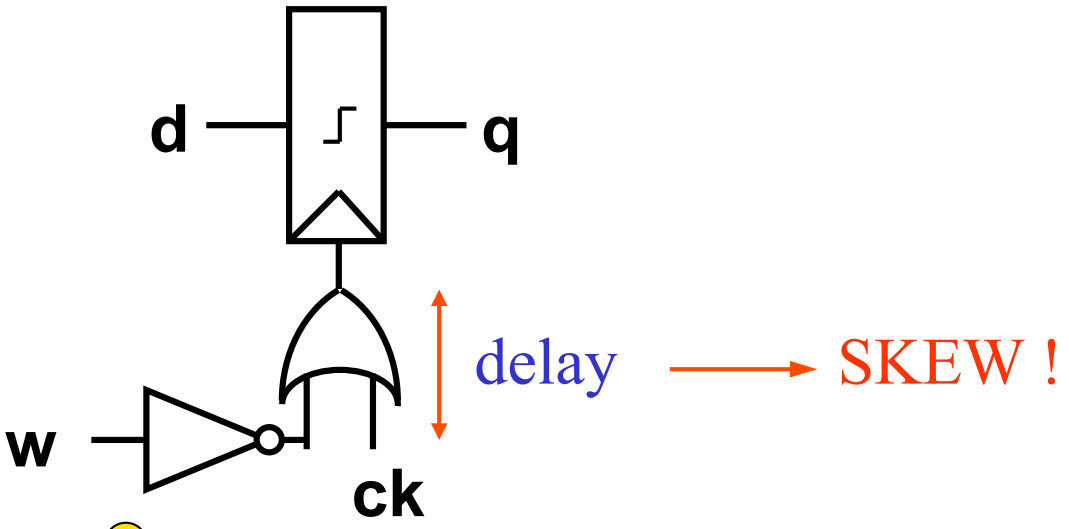




CMOS Circuits

Synchronous Memory :

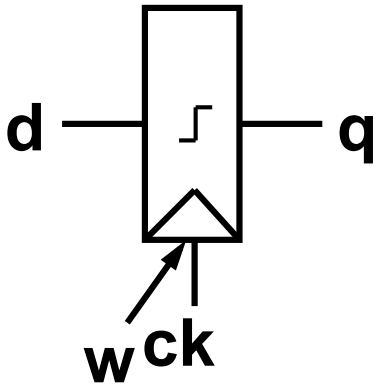
Write a data **d** on the rising edge of the clock **ck**
when a condition is true (write enable)

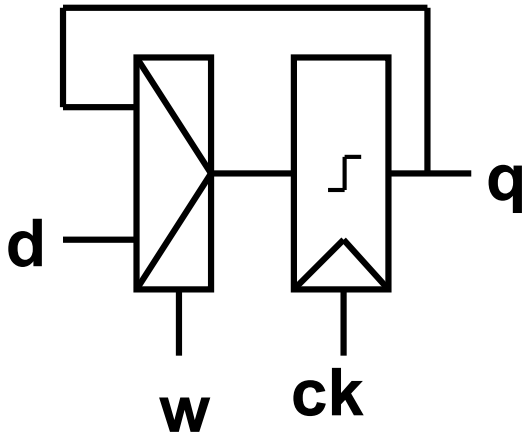


CMOS Circuits

Synchronous Memory :

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

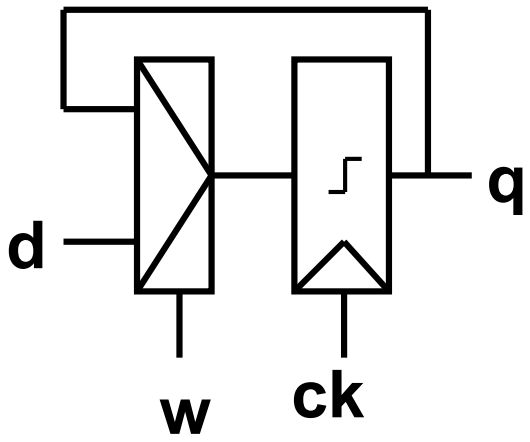




CMOS Circuits

Synchronous Memory :

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



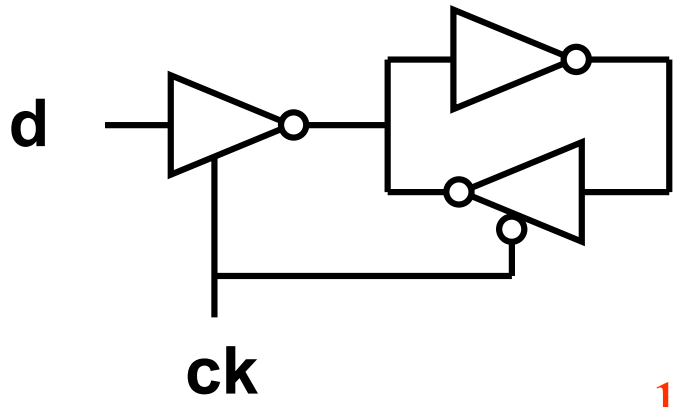
44 tr per register !!

CMOS Circuits

Memory : Latch

Hold a data (0 or 1)

Write a data (0 or 1)



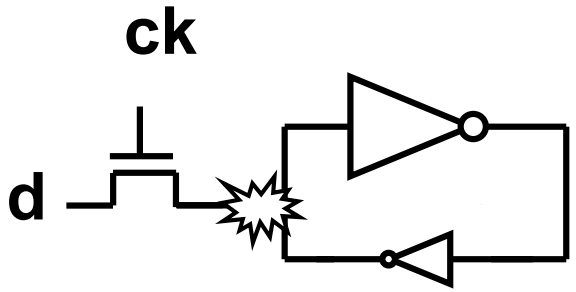
10 tr per latch

CMOS Circuits

Memory : Latch

Hold a data (0 or 1)

Write a data (0 or 1)



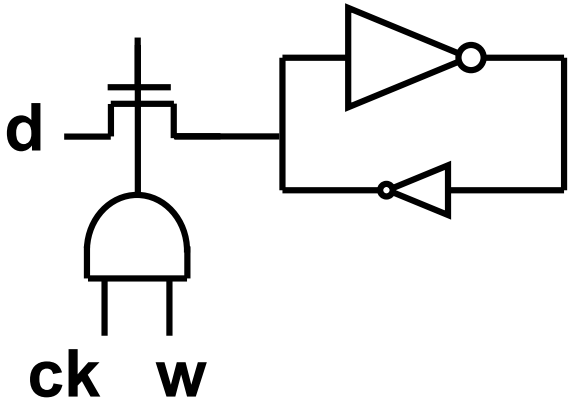
short circuit !

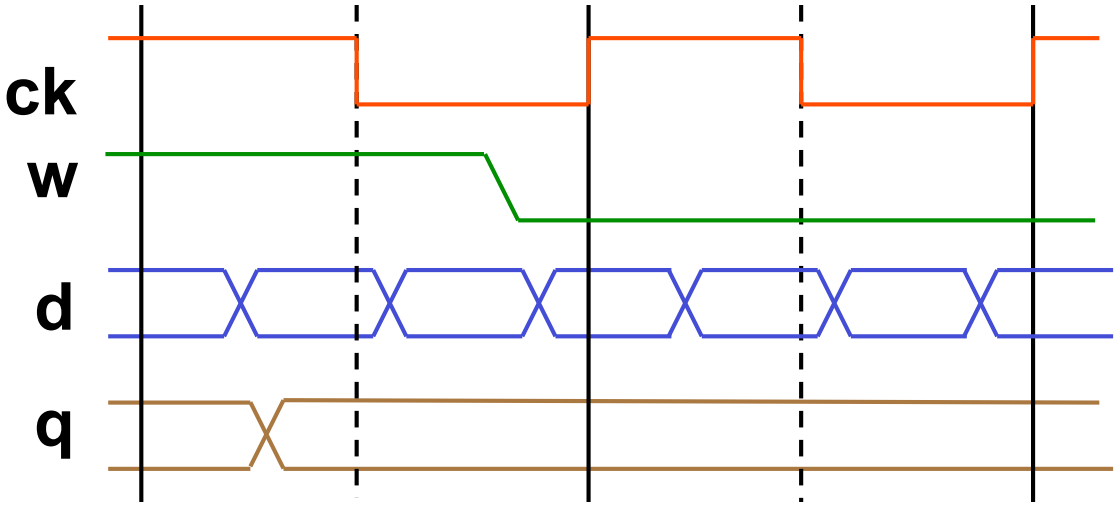
5 tr per latch

CMOS Circuits

Synchronous Memory :

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

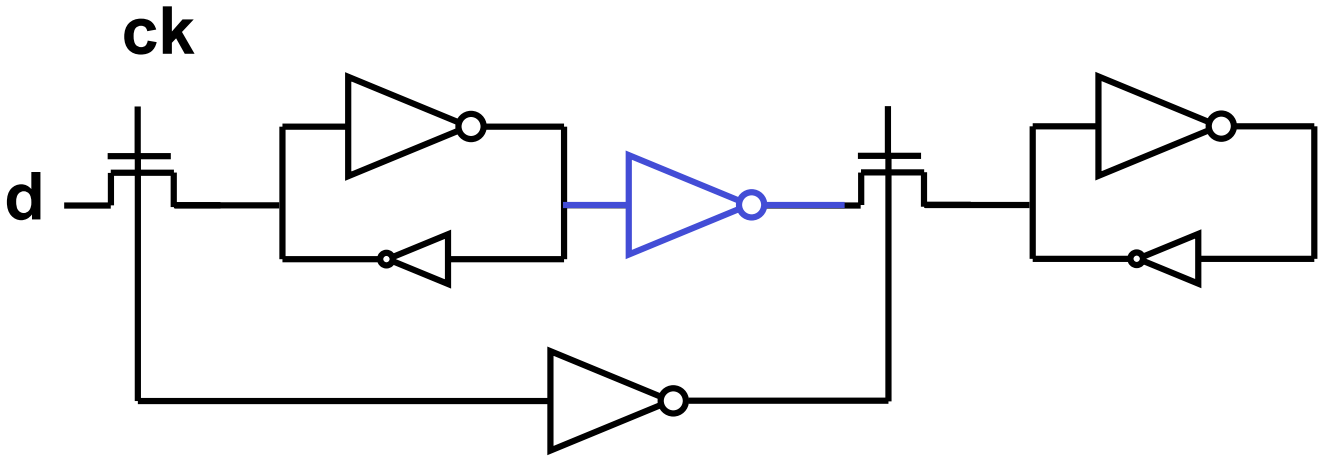




CMOS Circuits

Synchronous Memory :

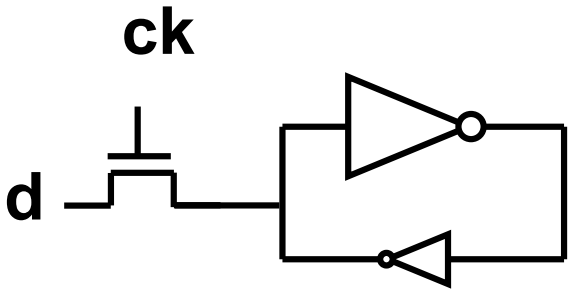
Write a data **d** on the falling edge of the clock **ck**

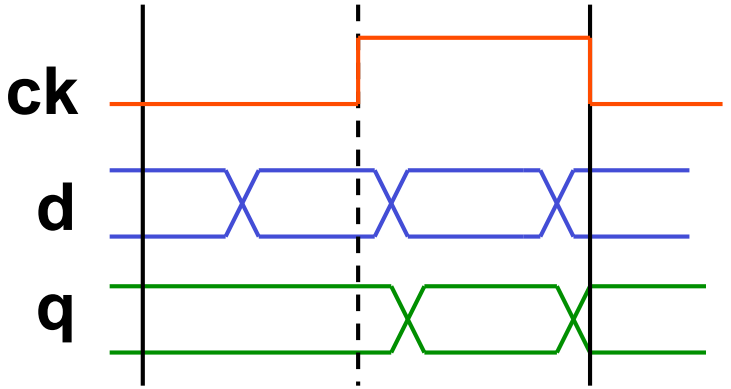


CMOS Circuits

Synchronous Memory :

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

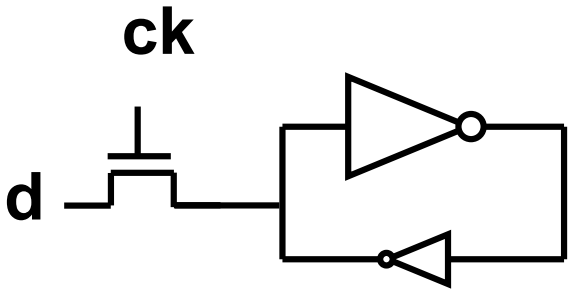


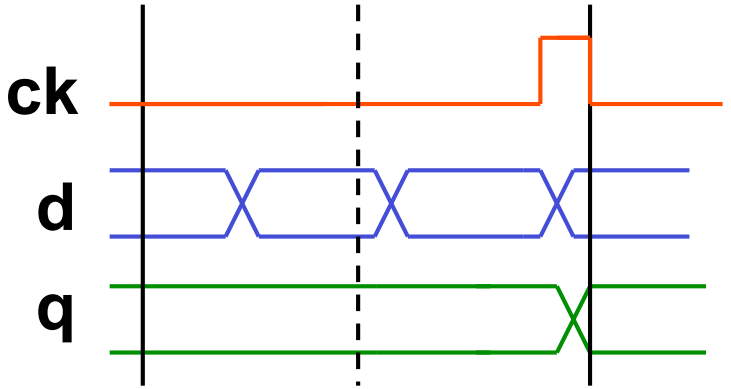


CMOS Circuits

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**