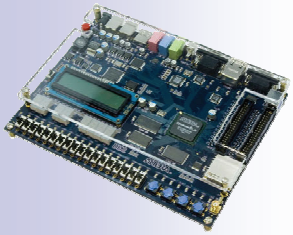
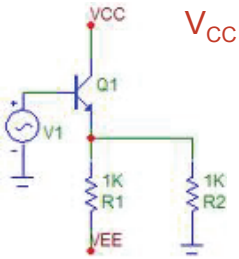
Struktury počítačových systémů



Přednáší: Richard Šusta

Přednáška 29. září

Voltage Symbols in Schematics



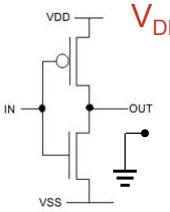
V_{CC} - Common Collector Voltage

GND – ground (0 V)

V_{EE} - Voltage Emitter-Emitter

V_{DD} - Voltage Drain-Drain

"0" - logic zero



"1" - logic "1"

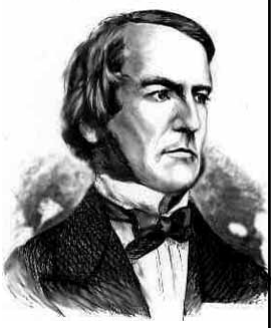
GND – ground (0 V)

V_{SS} - Voltage for Substrate & Sources

Booleova algebra

Mini-opakování znalostí
předmětu ***Logika a grafy***

George Boole (1815-1864)



Napsal řadu matematických prací

- kromě jiného novým způsobem prozkoumal logiku
  - redukoval ji na jednoduchou algebru
    - tím zavedl logiku do matematiky
      - později se jeho algebra začala nazývat Booleovská algebra

Booleovská algebra

Booleovská algebra je šestice (A, +, . , ', 0, 1) , kde

- A je neprázdná množina prvků, např. { 0, 1 },
- 0 je nejmenší prvek A
- 1 je největší prvek A
- ' je unární operace (komplement) a
- + . jsou dvě binární operace definované na A, které splňují následující axiomy:

Axiom	(a)	(b)
Komutativita	$a + b = b + a$	$a \cdot b = b \cdot a$
Neutralita	$a + 0 = a$	$a \cdot 1 = a$
Distributivita	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = a \cdot b + a \cdot c$
Komplementarita	$a + a' = 1$	$a \cdot a' = 0$
Agresivita	$a + 1 = 1$	$a \cdot 0 = 0$

(Pojmy: Booleovská logika je jeden případ abstraktní Booleovské algebry...)

Cz: Dualita axiomů a teorémů

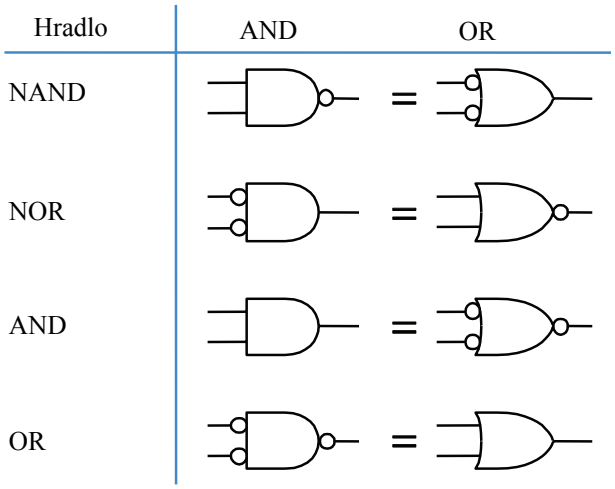
Vlastnost	AND OR 0 1	OR AND 1 0
Komutativita	$a + b = b + a$	$a \cdot b = b \cdot a$
Identita	$a + 0 = a$	$a \cdot 1 = a$
Distributivita	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = a \cdot b + a \cdot c$
Komplementarita	$a + a' = 1$	$a \cdot a' = 0$
Idempotence	$a + a = a$	$a \cdot a = a$
Agresivita	$a + 1 = 1$	$a \cdot 0 = 0$
Dvojí negace	$(a')' = a$	
Asociativita	$a + (b + c) = (a + b) + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c$
DeMorgan	$\mathbf{(a + b)' = a' \cdot b'}$	$\mathbf{(a \cdot b)' = a' + b'}$
Absorpce	$a + (a \cdot b) = a$	$a \cdot (a + b) = a$
Sloučení	$(x \cdot y) + (x \cdot y') = x$	$(x + y) \cdot (x + y') = x$

Eng: Manipulating Logic Expressions

Table 1.2 Laws (basic identities) of Boolean algebra.

Name of law	OR version	AND version
Identity	$x \mid 0 = x$	$x \& 1 = x$
One/Zero	$x \mid 1 = 1$	$x \& 0 = 0$
Idempotent	$x \mid x = x$	$x \& x = x$
Inverse	$x \mid x' = 1$	$x \& x' = 0$
Commutative	$x \mid y = y \mid x$	$x \& y = y \& x$
Associative	$(x \mid y) \mid z = x \mid (y \mid z)$	$(x \& y) \& z = x \& (y \& z)$
Distributive	$x \mid (y \& z) = (x \mid y) \& (x \mid z)$	$x \& (y \mid z) = (x \& y) \mid (x \& z)$
DeMorgan's	$(x \mid y)' = x' \& y'$	$(x \& y)' = x' \mid y'$

DeMorganovy ekvivalenty hradel



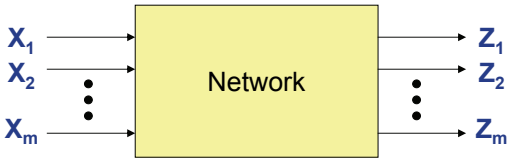
Richard S. Sandige, Digital Design Essentials, Prentice-Hall, 2002. p 141

1. **AND <-> OR**
2. **invertujeme znak negace**,
   tj. přidáme tam, kde dříve nebyl, a smažeme tam, kde dříve byl...

Logické funkce

Matematický zápis

Kombinační / Sekvenční obvod



Kombinační obvod:

bezestavový, výstup je jednoznačně určený vstupem

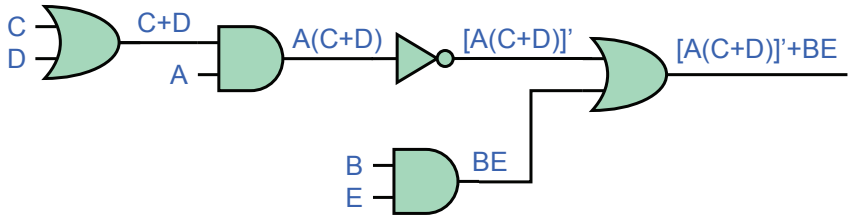
Sekvenční obvod:

Výstupní hodnota závisí na historii hodnot vstupního vektoru a vnitřních veličinách.

Logické výrazy a obvody

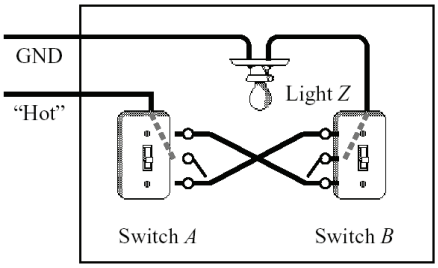
Jakýkoliv logický výraz lze implementovat jako logickou funkci

X = [A(C+D)]'+BE



Pravdivostní tabulka

- Vyvinul 1854 George Boole, později dopracoval Claude Shannon (Bell Labs)



Spínač světla

Vstupy		Výstupy
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

- **Jedinečný otisk logické funkce**
  - 1 funkci lze realizovat mnoha způsoby, které vedou na totožnou tabulku
  - Dovoluje přímo zapsat v kanonickém tvaru disjunktivním nebo konjunktivním
- **Nevýhoda – složitost zápisu dána $O(2^n)$**

Disjunktivní normální forma 1/2

- známa také jako
- **Sum-of-Products = S-o-P**
- **mintermy**

F = *001* *011* *101* *110* *111*

A'B'C+ A'BC+ AB'C + ABC' + ABC

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

F' = A'B'C' + A'BC' + AB'C'

Disjunktivní normální forma 2/2

A	B	C	minterms	
0	0	0	A'B'C'	m0
0	0	1	A'B'C	m1
0	1	0	A'BC'	m2
0	1	1	A'BC	m3
1	0	0	AB'C'	m4
1	0	1	AB'C	m5
1	1	0	ABC'	m6
1	1	1	ABC	m7

zkratka pro mintermy
3 proměnných

F v kanonickém tvaru:

F(A, B, C) = Σm(1,3,5,6,7)
= m1 + m3 + m5 + m6 + m7
= A'B'C + A'BC + AB'C + ABC' + ABC

Kanonická forma ≠ minimalní formy

F(A, B, C) = A'B'C + A'BC + AB'C + ABC + ABC'
= (A'B' + A'B + AB' + AB)C + ABC'
= ((A' + A)(B' + B))C + ABC'
= C + ABC'
= ABC' + C
= AB + C

S-o-P a de Morganův teorém

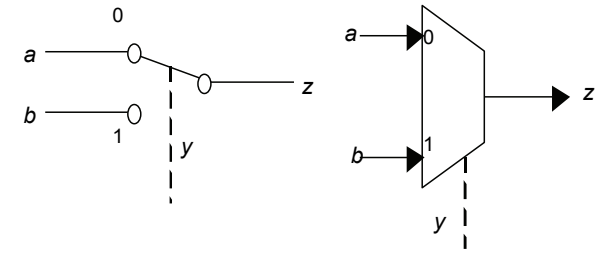
- Sum-of-products
  - F' = A'B'C' + A'BC' + AB'C'
- Aplikujeme de Morganův teorém
  - (F')' = (A'B'C' + A'BC' + AB'C')'
  - F = (A + B + C) (A + B' + C) (A' + B + C)

Examples of S-o-P

Simple Combinational Logic Circuits

2:1 Multiplexer

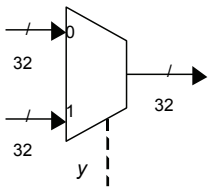
A multiplexer (a.k.a. data selector) has
- n control inputs
- 2^n data inputs
- a single data output



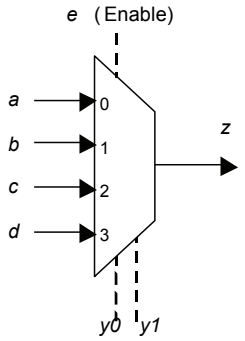
(a) Switch view

(b) Mux symbol

Multiplexers



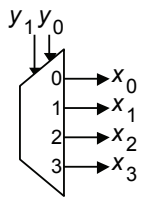
Mux array



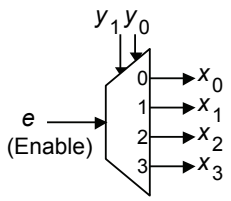
4:1 mux with enable

Decoder

Decoders allow selecting one of 2^N inputs by N as a bit address



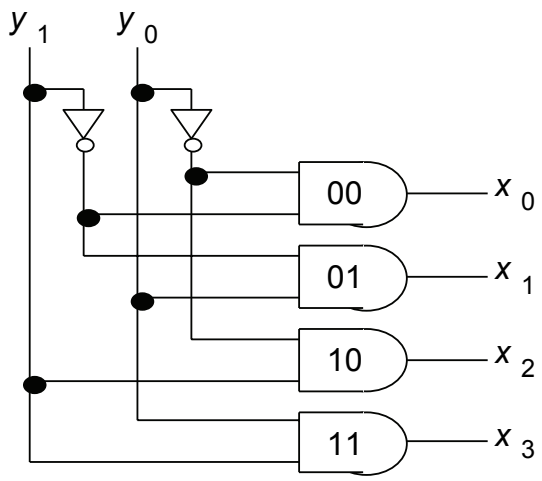
Decoder 2 x 4

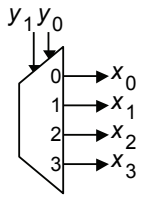


Demultiplexer
= decoder with “enable”

2:1 decoder

Design by direct application of S-o-P

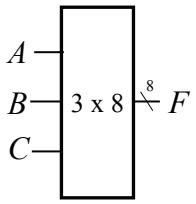




2 x 4 decoder

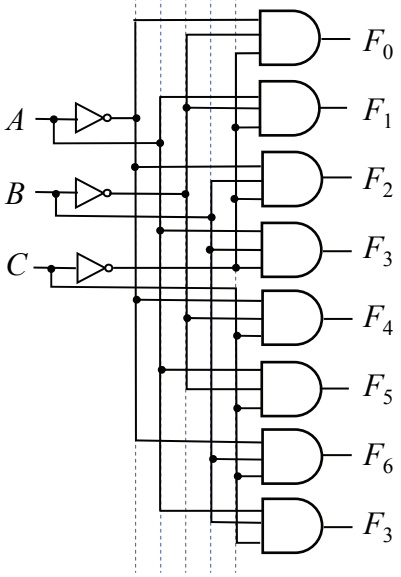
Design decoder 3 x 8

A	B	C	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

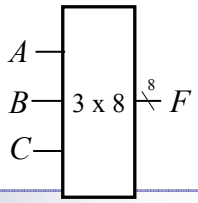


Solution - decoder 3 x 8

Design by direct application of S-o-P



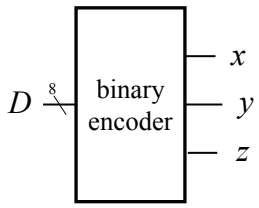
A	B	C	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Binární enkodér inverze dekodéru

Design by direct application of S-o-P

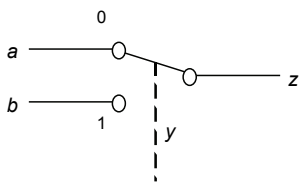
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



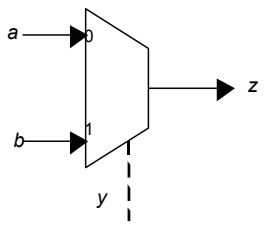
$$x = D_4 + D_5 + D_6 + D_7$$
$$y = D_2 + D_3 + D_6 + D_7$$
$$z = D_1 + D_3 + D_5 + D_7$$

Pokud může nastat možnost nastavení více vstupů současně do 1, pak musíme použít složitější prioritní enkodér, bude na některé další přednášce

Design 2:1 Multiplexer

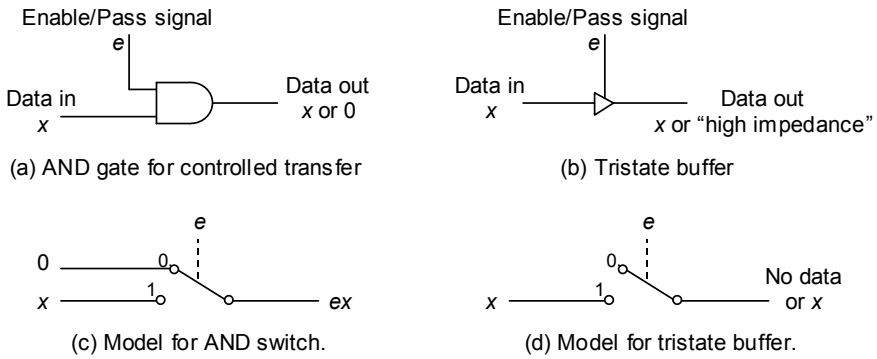


(a) Switch view



(b) Mux symbol

Model of Gate as a Control Element of Switch



(a) AND gate for controlled transfer

(b) Tristate buffer

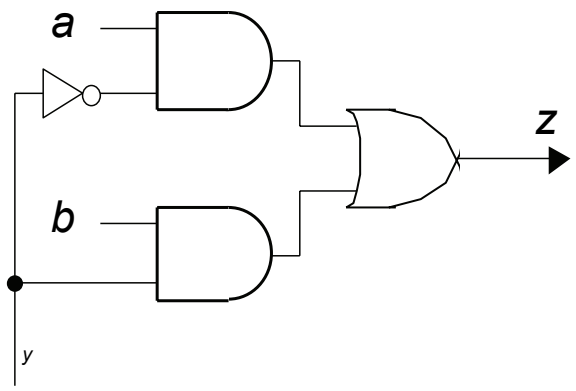
(c) Model for AND switch.

(d) Model for tristate buffer.

An AND gate and a tristate buffer act as controlled switches or valves. An inverting buffer is logically the same as a NOT gate.

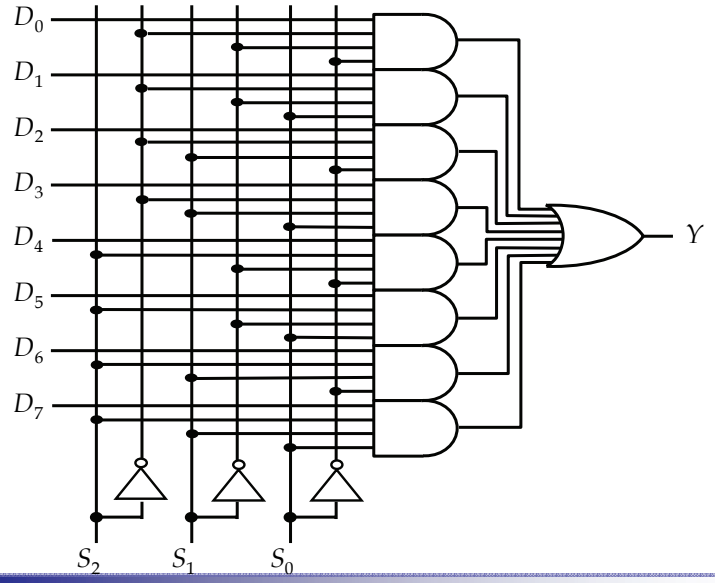
2:1 multiplexer

Design by application of switching model of gate



8:1 Multiplexer

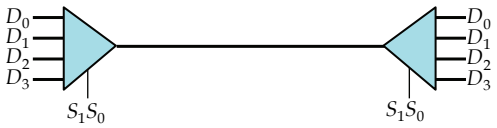
Design by application of switching model of gate

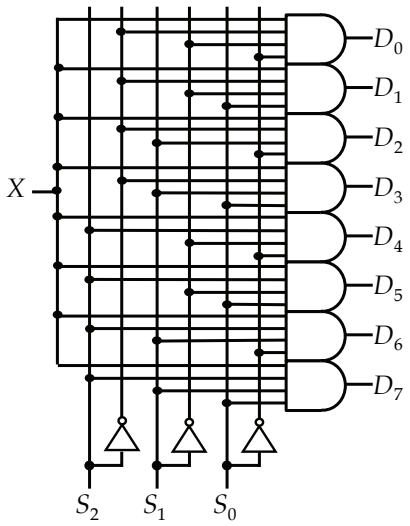


Demultiplexers

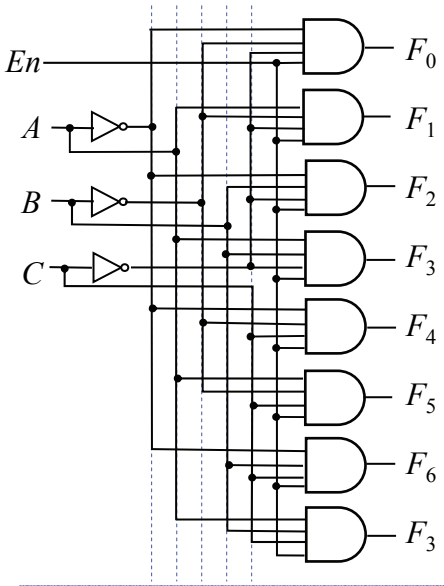
Design by application of switching model of gate

A demultiplexer (demux) is a decoder that only selects an output if its enable signal is asserted.



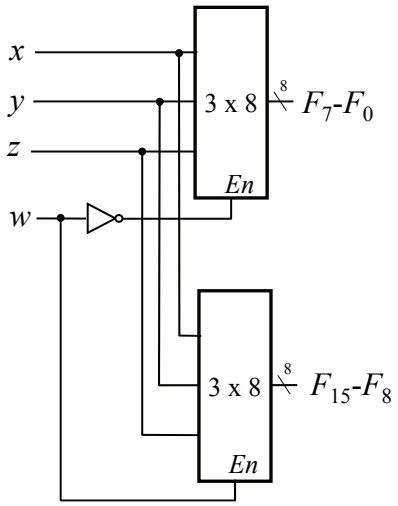


Decoder/demultiplexer 3 z 8 s Enable



- Vstup En - enable
  - uvolnění výstupu,
  - Enable velmi časté v logických obvodech
  - Zpravidla bývá negované
  - Někdy se dělá jako AND několika vstupů, pak se občas značí jako G (gate enable), např. demultiplexor 74138

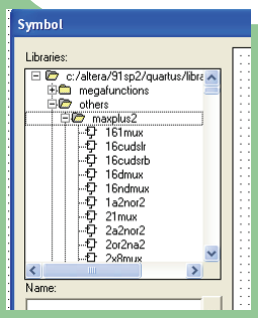
Využití enable 4 x 16 dekodér z 3 x 8



Z jednoduchých bloků vytváříme složitější funkce

přidán vstup enable pro uvolnění výstupu

7400 series



in maxplus2 library

Code of Datasheet (cz: katalogový list)

- Each IC contains a code identifying the package

N74LS00

Manufacturers Code

N = National Semiconductors
SN = Signetics

Specification

Family
L
LS
H

Member code
00 = Quad 2 input NAND
02 = Quad 2 input Nor
04 = Hex Invertors
20 = Dual 4 Input NAND

current, voltage and time properties
- they are different in FPGA

functionality

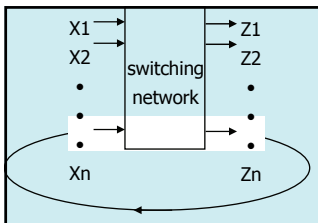
Max-warning of Maxplus2

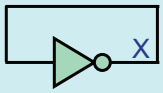
*The Maxplus2 library circuits offer functionality very similar to 74 series, but **not in all cases**!*

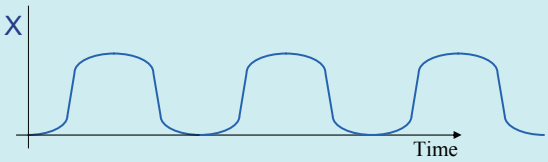
Sequential Logic Circuits

Genesis

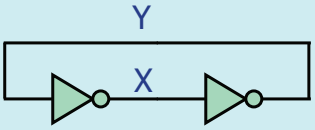
Circuits with Feedback

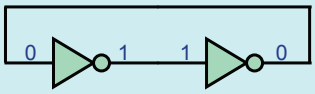


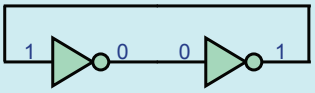




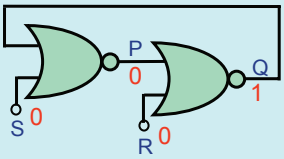
What are the values of X and Y in this circuit?







The Set-Reset Latch



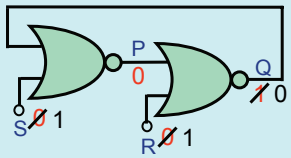
(i) Assume P=0;

P=0 and Q=1 is stable

Některé anglické texty (hlavně z průmyslové automatizace) používají místo SET slovo LATCH a místo RESET slovo UNLATCH

The Set-Reset Latch

What happens if R and S are 1 at the same time?

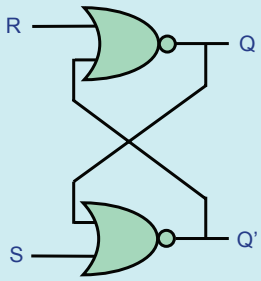


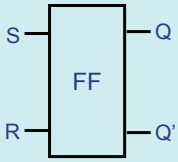
P is no longer a complement of Q!

S=1 and R=1 is not allowed for a usage as a latches.

The Set-Reset Latch

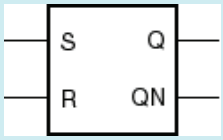
using NOR loop

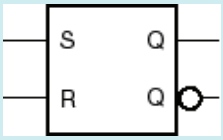


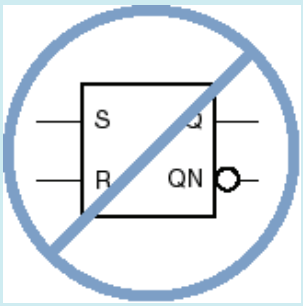


Characteristic equation $Q(t+\varepsilon) = R'(t).(S(t)+Q(t))$

S-R latch symbols

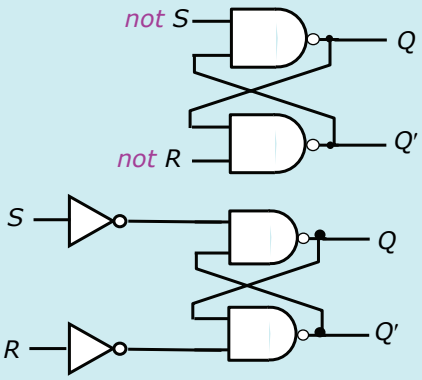


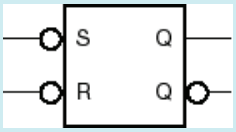


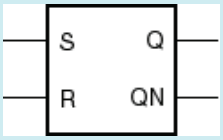


S-R latch using NAND gates

using NAND loop







Characteristic equation $Q(t+\varepsilon) = S(t)+R'(t).Q(t)$

NOR – NAND SR

NOR equation $Q(t+\varepsilon) = R'(t).(S(t)+Q(t)) = R'(t).S(t) + R'(t).\ Q(t)$

has reset priority

NAND equation $Q(t+\varepsilon) = S(t)+R'(t).Q(t)$

has set priority

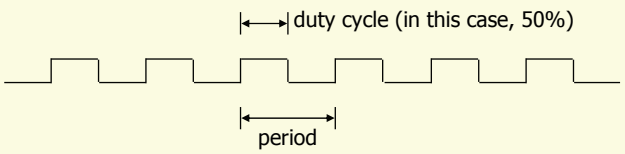
The behaviours are equal after adding the forbidden state constrain!

S	R	Q	QN
0	0	last Q	last QN
0	1	0	1
1	0	1	0
1	1	0	0

S_L	R_L	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	last Q	last QN

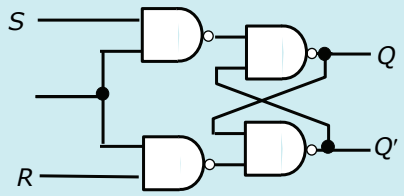
Clocks

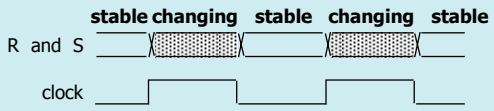
- Used to keep time
  - Wait long enough for inputs (R' and S') to settle
  - Then allow to have effect on value stored
- Clocks are regular periodic signals
  - Period (time between ticks)
  - Duty-cycle (time clock is high between ticks - expressed as % of period)



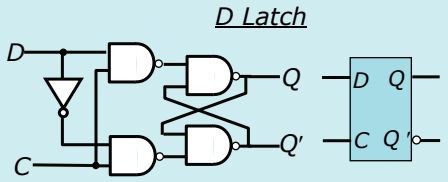
Clocks

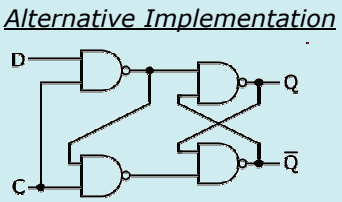
- Controlling an R-S latch with a clock





D Latch (7475) - Quartus latch

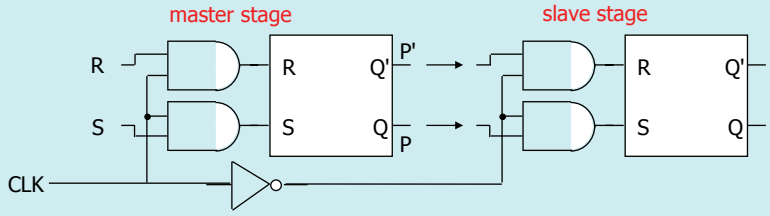




- The *D* latch stores the value on the *D* input when the enable input is asserted.

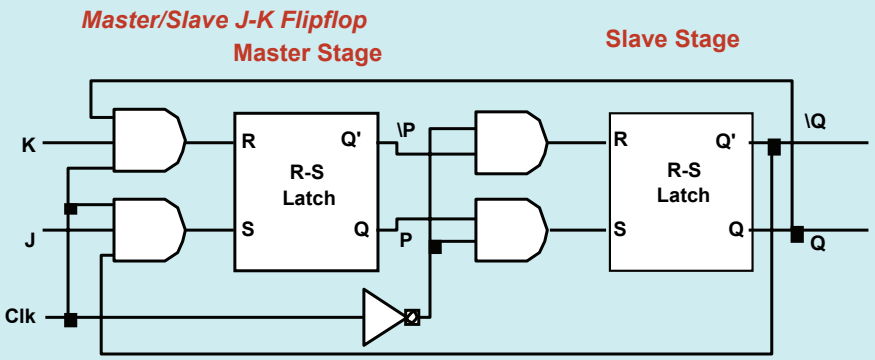
Master-Slave Structure

- Break flow by alternating clocks (like an air-lock)



JK flip-flop – Quartus JKFF

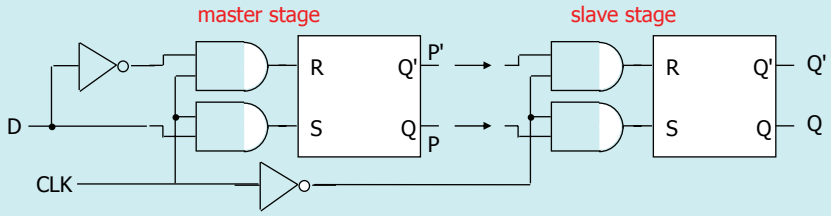
7470-73, 74110-114



J ~ S, K ~ R - forbidden inputs S=1, R=1 are modified. For J=1, K=1, JK goes to its opposite state.

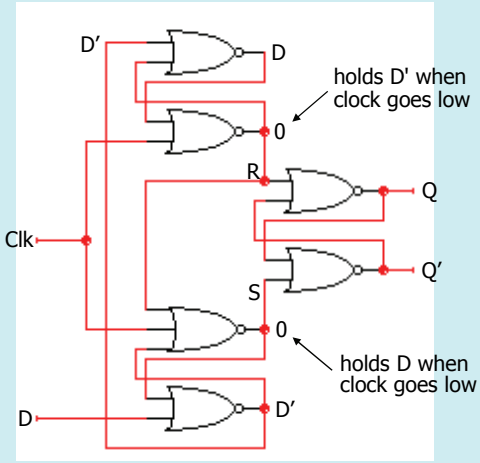
D Flip-Flop – Quartus DFF master-slave

- Make S and R complements of each other



Edge-Triggered Flip-Flops (7474)

- More efficient solution: only 6 gates



characteristic equation
Q(t+1) = D

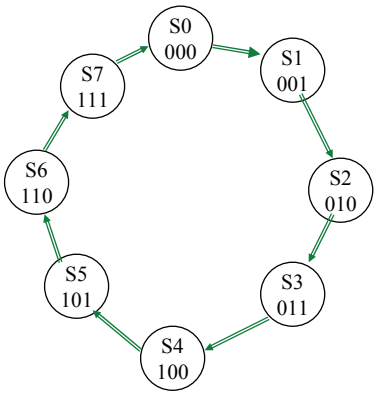
Counters

in hardware
ripple/synchronous

Counters

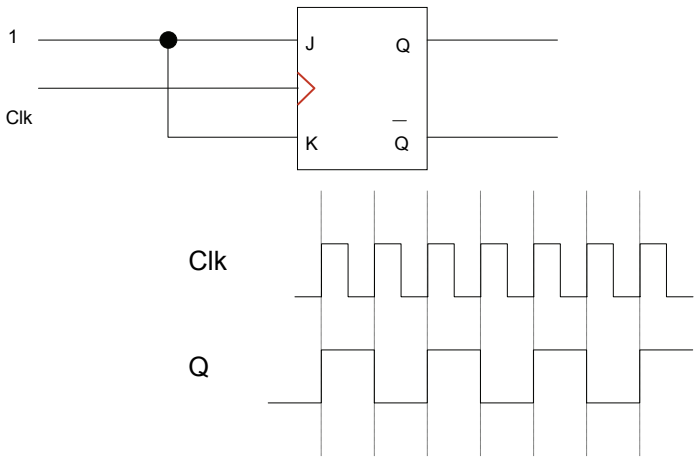
- Used for the control of sequence and program execution.
- Two categories of counters: *asynchronous* and *synchronous.*
- The asynchronous counters produce the outputs in sequence
- The outputs of the synchronous counters are available at the same time.

3bit Binary Counter State Diagram



Counters (Cont'd)

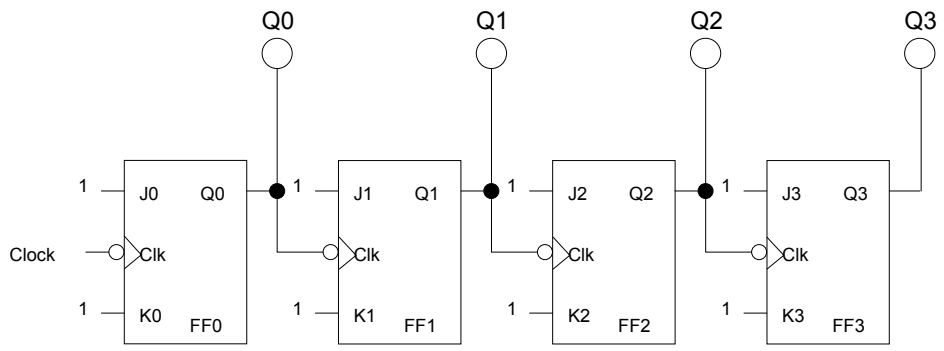
- Counters are made with either J-K or T-type flip-flops.



Counters

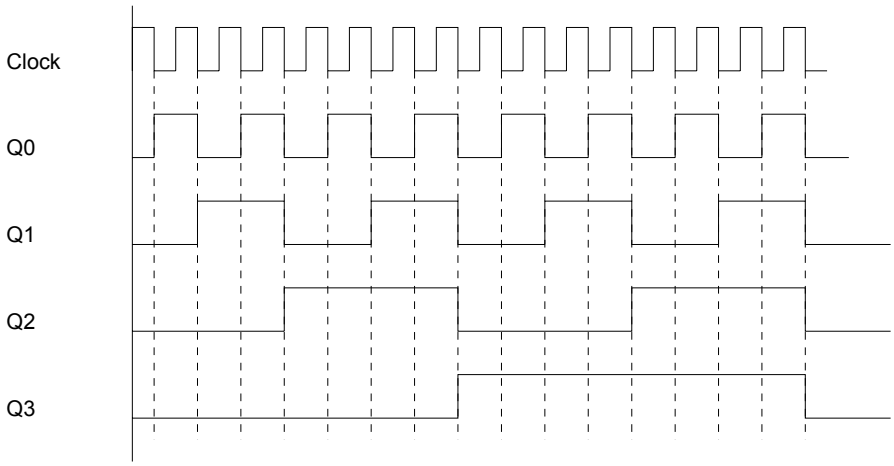
- Q produces one pulse for every two clock pulses input.
- The counter counts once for every two clock pulses.
- The frequency at Q is half of that at the clock.
- Sometimes called a divider.
- A J-K flip-flop can be regarded as a divide-by-2 counter.

Divide-by-16 Ripple (Asynchronous) Counter (cz: asynchronní čítač)



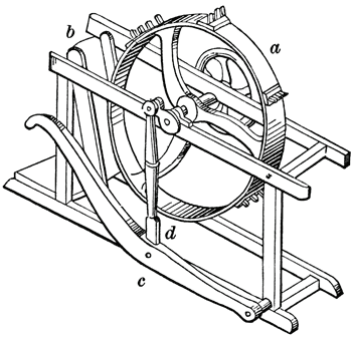
Divide-by-16 Ripple Counter

Ripple counter counts up



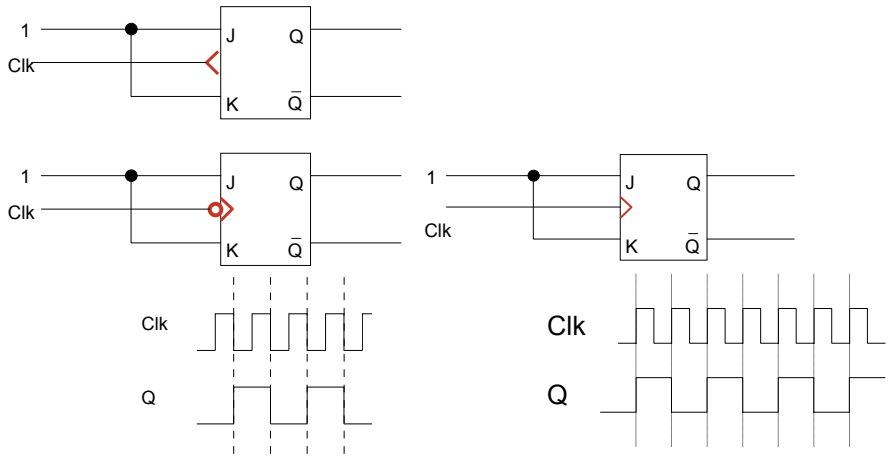
Ripple

- **ripple** (*cz:drhlice*)
  Etymology: Middle English ***repylle*** *from old high German* ***riffila****=saw:*
  a large instrument like a comb (cz:hřebínek) for removing seeds and other matter from flax or hemp (cz:len nebo konopí)
- ***ripple*** *(cz: zvlnit, zčeřit) - become covered with or form in small waves or undulations (cz:vlnění), to flow in small waves, to fall in soft undulating folds or wavy lines*
- *Ripple Effect = Domino Effect/Chain Reaction*



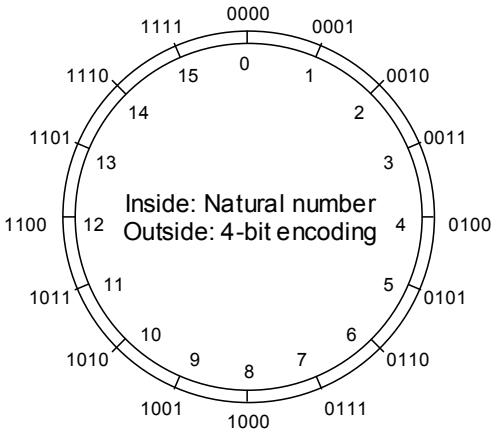
Counters (Cont'd)

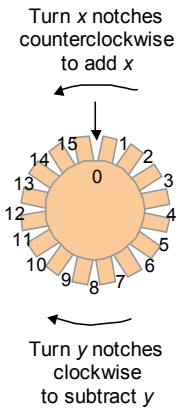
- Falling edge JK (*cz:náběžná hrana*) Rising edge JK (*cz: spádová, sestupná hrana*)



Counter Carry/Borrow

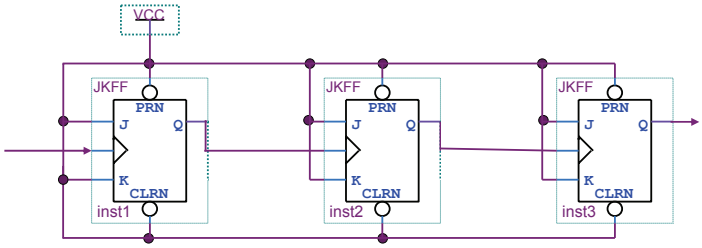
carry <-> borrow



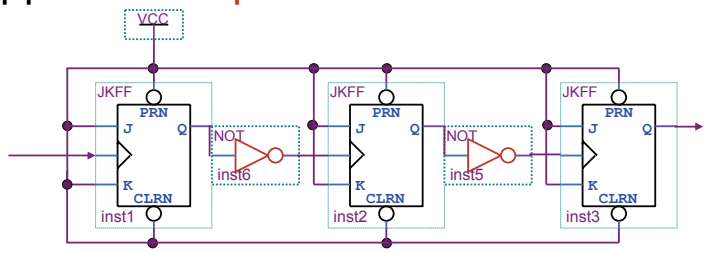


7 bit ripple counter

- Ripple count down



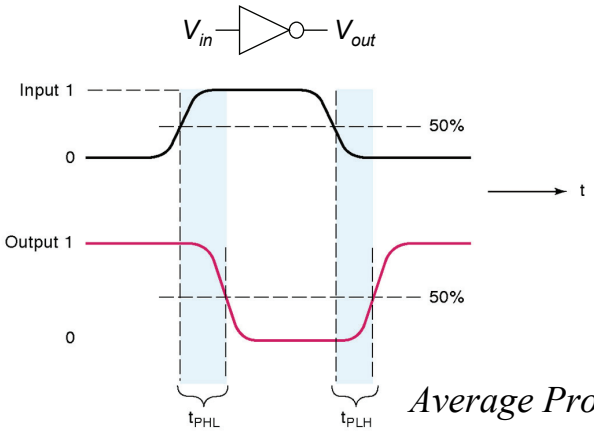
- Ripple count up



Sequential Logic Circuits

Delay

Propagation Delay Time (cz:zpoždění



t_{PHL} - to go from a high to a low

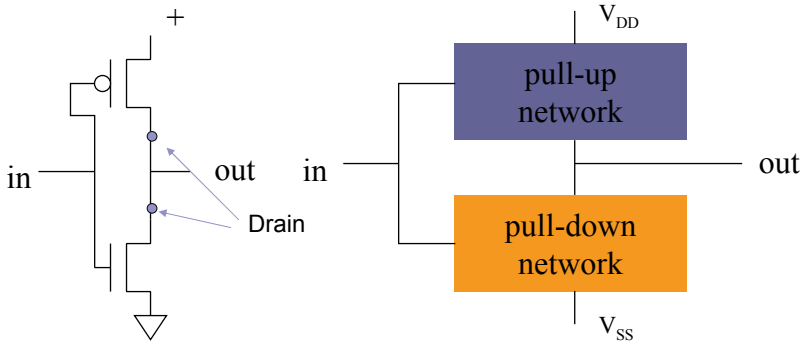
t_{PLH} - to go from a low to a high

Ideally, t_p should be as short as possible

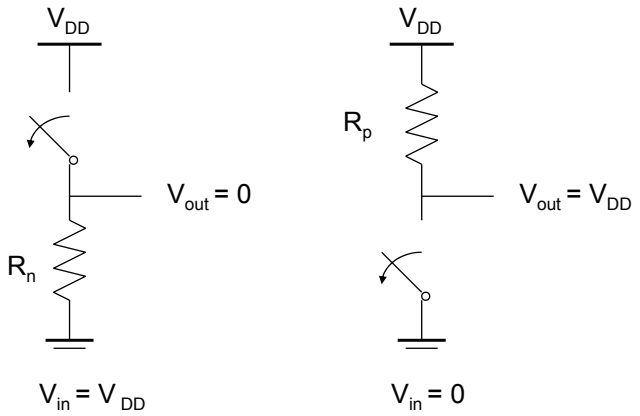
Average Propagation Delay Time

$$t_p = \frac{t_{pHL} + t_{pLH}}{2}$$

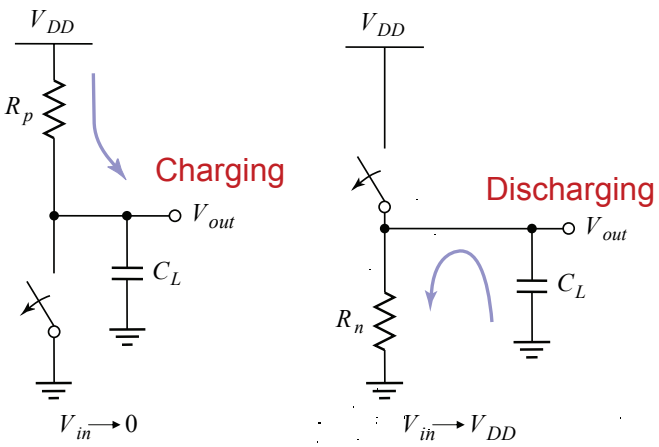
CMOS Inverter



Switch Models of a CMOS Inverter

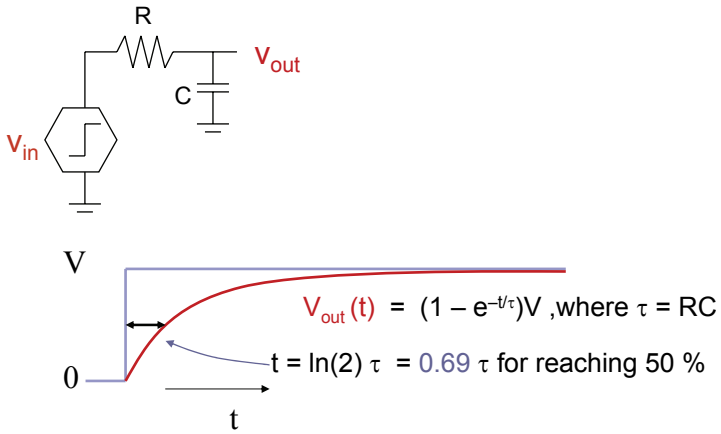


CMOS Inverter: Transient Response



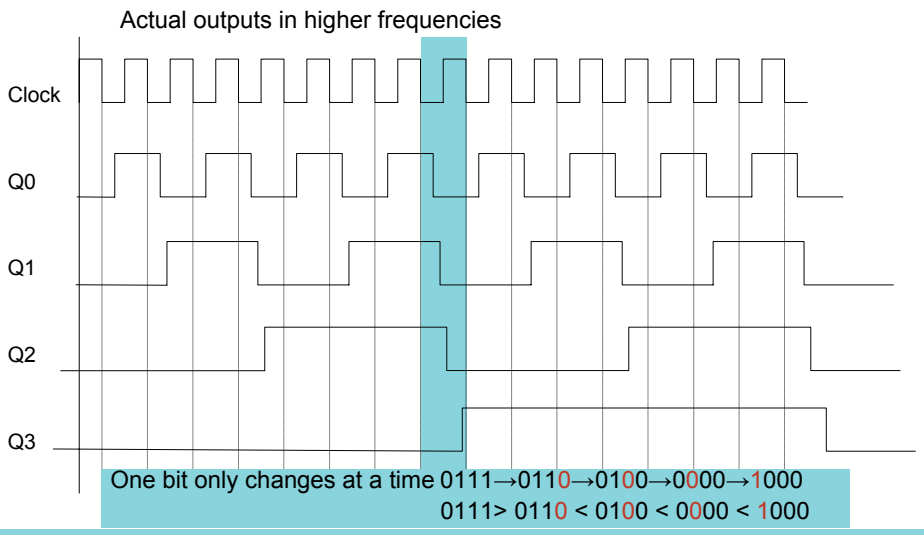
(a) Low-to-high output (b) High-to-low output

Model of Modeling Propagation Delay



$$t_p = 0.69\ C\ R$$

Divide-by-16 Ripple Counter (Cont'd)



Intermediate states are always less than next state

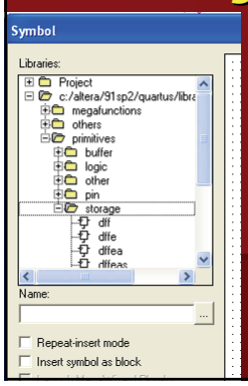
Divide-by-16 Ripple Counter (Cont'd)

- Propagation delay happens in operations of flip-flops.
- Time delay for all output clocks compared with their input clocks.
- Outputs are not available at the same time, it is an *asynchronous* counter.

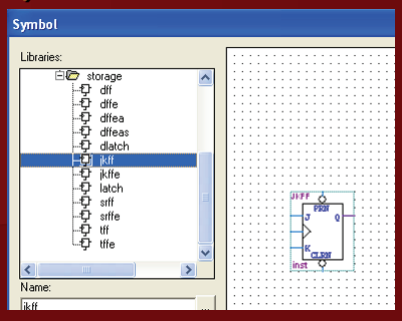
Max-warning of Maxplus2

Avoid using D and JK flip-flops circuits from maxplus2 library!

Prefer storage primitives of Quartus, e.g. latch, dff, jkff, and tff !

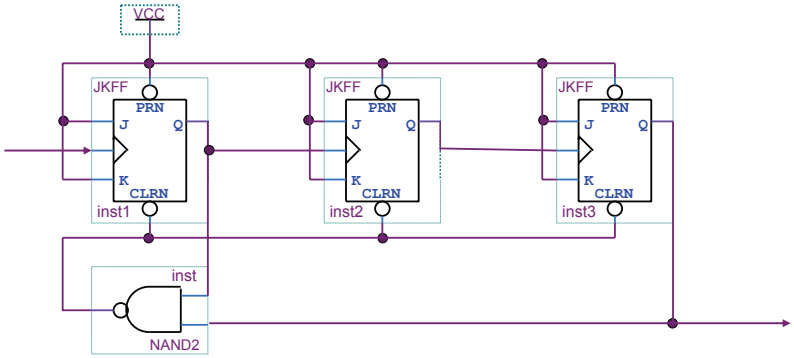


Mind your head



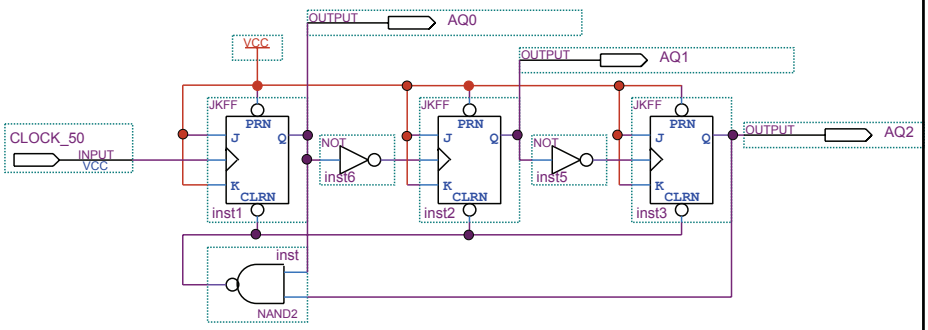
Divider modulo 5

- Is it correct?

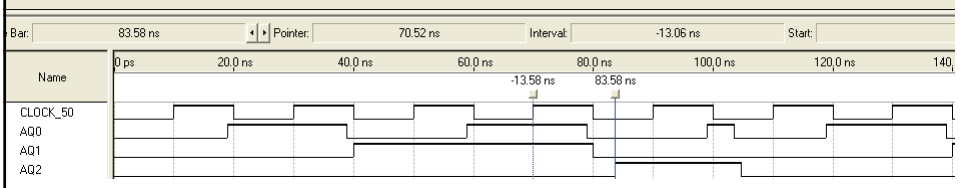


Correct divider

- We must use counter up !!!



Simulation



Synchronous counter

Do not use the previous schematic of asynchronous dividers with synchronous counters!

It does not work!