Základy číslicové elektroniky

Vyučující:

Zdeněk Plíva

e-mail: zdenek.pliva@tul.cz
3536

Miroslav Holada
Leoš Petržílka
…

Přehled témat

- Číselné soustavy
- Číslicová elektronika
- Logické stavy
- Základní logické funkce
- Booleova algebra
- Kombinační obvody
- Sekvenční obvodu

Proč číslicové obvody ??

- Velký rozptyl výrobních parametrů bez změny funkce
- Spolehlivější přenos a zpracování
- Menší náchylnost na rušení šumem
- Díky kompresním kódům možnost přenosu vyššího množství dat
- Obvody jsou univerzálnější – velké série / nízké náklady
- Vyšší stupeň integrace
- Možnost komprese přenášených dat

Číslicové obvody - pojmy

Technologie výroby

- TTL
- CMOS

Typy logických obvodů

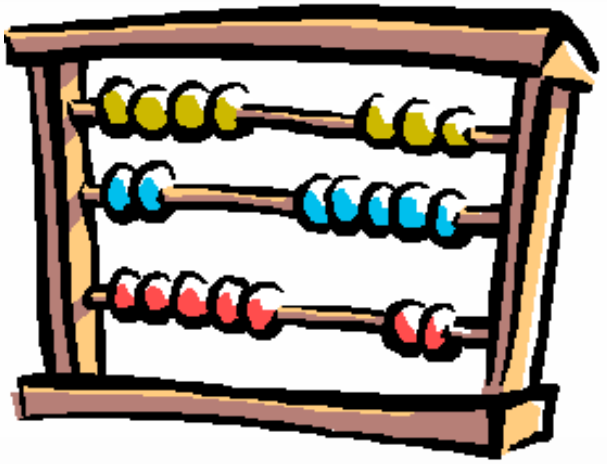
- Kombinační
- Sekvenční

Stupeň integrace

- SSI – malá integrace (Small Scale Integration) – 1-12 hradel
- MSI – střední integrace (Middle SI) – 13-99 hradel
- LSI – vysoká integrace (Large SI) – 100-9 999 hradel
- VLSI/XLSI – velmi vysoká int. (Very/eXtra LSI) – 10 000-99 999 hradel
- ULSI – hóódně vysoká integrace (Ultra LSI) – nad 100 000 hradel

Číselné soustavy

Polyadické číselné soustavy



z-adické číslo

v **z-adické** soustavě:

$$\mathbf{A = a_n z^n + \ldots + a_0 z^0 + a_{-1} z^{-1} + \ldots + a_{-m} z^{-m}}$$

z je základ soustavy $z \geq 2$ (2, 8, 10, 16)

$\mathbf{a_i}$ je z-adická číslice $0 \leq a_i < z$ („z“ různých a_i)

$a_{min} = 0$ $a_{max} = z - 1$

Číselné soustavy

1 001

$$1*10^3 + 0*10^2 + 0*10^1 + 1*10^0$$

Číselné soustavy

1001…

Je-li toto číslo psáno v soustavě:	pak je dekadická hodnota:
Binární	**9**
Oktalové	**513**
Dekadické	**1001**
Hexadecimální . .	**4097**

Číselné soustavy

$1001_{10/D}$

$1*10^3 + 0*10^2 + 0*10^1 + 1*10^0 = 1001_D$

$1001_{8/O}$

$1*8^3 + 0*8^2 + 0*8^1 + 1*8^0 = 513_D$

Číselné soustavy

$1001_{2/B}$

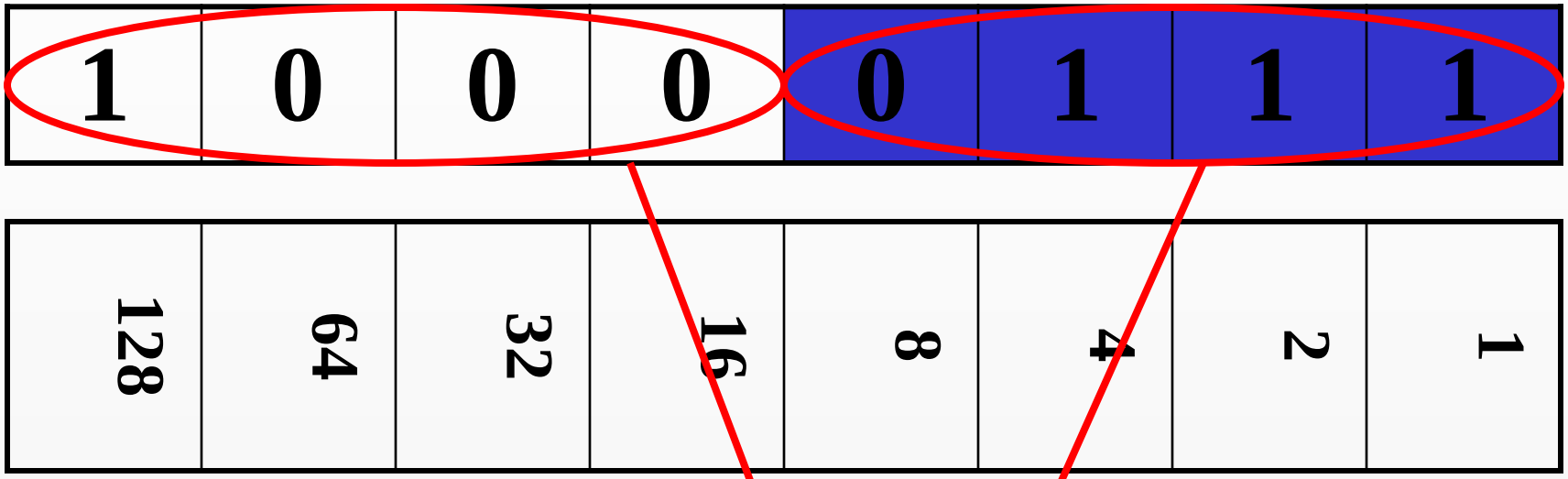
$1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 9_D$

$1001_{16/H}$

$1*16^3 + 0*16^2 + 0*16^1 + 1*16^0 =$

4097_D

Převod 2 → 10

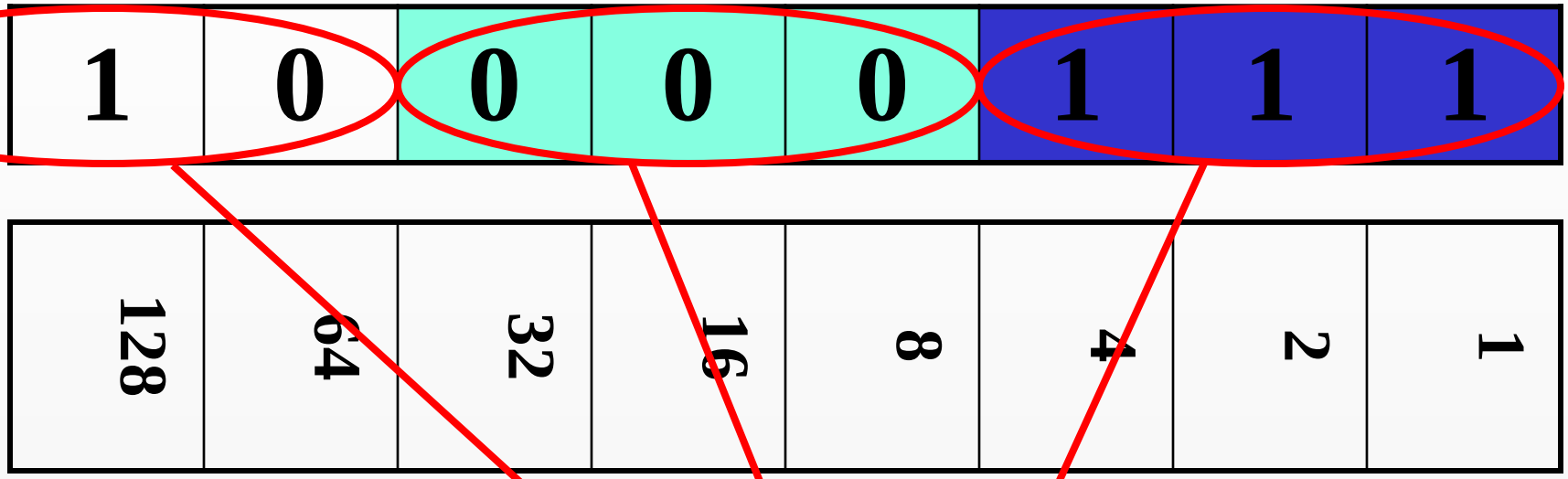


$$128 + 4 + 2 + 1 = 135_D$$

$$10000111_B = 87_H$$

$$8 * 16 + 7 = 135_D$$

Převod 2 → 10



$$128 + 4 + 2 + 1 = 135_D$$

$$010000111_B = 207_O$$

$$2*64 + 0*8 + 7 = 135_D$$

Soustavy

číslo	Z = 2	Z = 8	Z = 10	Z = 16
0	0000	00	00	00
1	0001	01	01	01
2	0010	02	02	02
3	0011	03	03	03
4	0100	04	04	04
5	0101	05	05	05
6	0110	06	06	06
7	0111	07	07	07
8	1000	10	08	08
9	1001	11	09	09
10	1010	12	10	0A
11	1011	13	11	0B
12	1100	14	12	0C
13	1101	15	13	0D
14	1110	16	14	0E
15	1111	17	15	0F

Převod 10 → 2

Číslo před desetinnou čárkou dělíme dvěma a zapisujeme zprava doleva zbytky při dělení:

364 : 2 = 182 zb. 0

Převod **desítkového** čísla do **dvojkového**

0_2

Převod 10 → 2

Číslo před desetinnou čárkou dělíme dvěma a zapisujeme zprava doleva zbytky při dělení:

364	**: 2 = 182**	**zb. 0**
182	**: 2 = 91**	**zb. 0**

Převod **desítkového** čísla do **dvojkového**

00_2

Převod 10 → 2

Číslo před desetinnou čárkou dělíme dvěma a zapisujeme zprava doleva zbytky při dělení:

364	**: 2 = 182**	**zb. 0**
182	**: 2 = 91**	**zb. 0**
91	**: 2 = 45**	**zb. 1**

Převod **desítkového** čísla do **dvojkového**

100_2

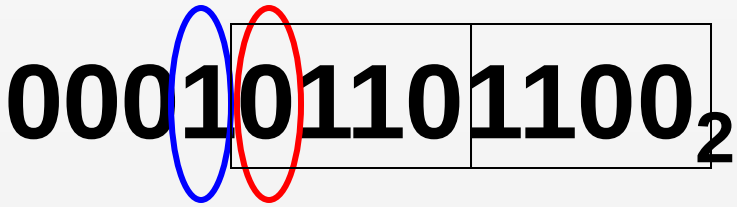
Převod 10 → 2

Číslo před desetinnou čárkou dělíme dvěma a zapisujeme zprava doleva zbytky při dělení:

364	**: 2 = 182**	**zb. 0**
182	**: 2 = 91**	**zb. 0**
91	**: 2 = 45**	**zb. 1**
45	**: 2 = 22**	**zb. 1**
22	**: 2 = 11**	**zb. 0**
11	**: 2 = 5**	**zb. 1**
5	**: 2 = 2**	**zb. 1**
2	**: 2 = 1**	**zb. 0**

Převod **desítkového** čísla do **dvojkového**

$$16C_H$$



Převod 10 → 16

Číslo dělíme šestnácti a zapisujeme zprava doleva zbytky při dělení:

364 : 16 = 22 zb. 12

Převod **desítkového** čísla do **šestnáctkového (hexa)**

$1 \rightarrow 1$ $10 \rightarrow A$

$2 \rightarrow 2$ $11 \rightarrow B$

...

$9 \rightarrow 9$ $15 \rightarrow F$

C_{16}

Převod 10 → 16

Číslo dělíme šestnácti a zapisujeme zprava doleva zbytky při dělení:

364	**: 16 = 22**	**zb. 12**
22	**: 16 = 1**	**zb. 6**

Převod **desítkového** čísla do **šestnáctkového (hexa)**

$$1 \rightarrow 1 \quad 10 \rightarrow A$$
$$2 \rightarrow 2 \quad 11 \rightarrow B$$
$$\ldots$$
$$9 \rightarrow 9 \quad 15 \rightarrow F$$

$$16C_{16}$$

Převod 10 → 16

Číslo dělíme šestnácti a zapisujeme zprava doleva zbytky při dělení:

364 : 16 = 22 zb. 12

22 : 16 = 1 zb. 6

Převod **desítkového** čísla do **šestnáctkového (hexa)**

$16C_{16}$

1 → 1 10 → A

2 → 2 11 → B

…

9 → 9 15 → F

Zobrazování záporných čísel

a) Přímý kód – přidáme znaménkový bit
+ … 0 - … 1 (např.: –6 … 1 0110)
složité aritmetické operace – *nepoužívá se*

b) Dvojkový doplněk
inverze čísla zvětšenou o 1

Př.

zobrazení čísla **–14**

01110 → inverze = **10001**

přičíst 1 → **10001** + **1** = **10010**

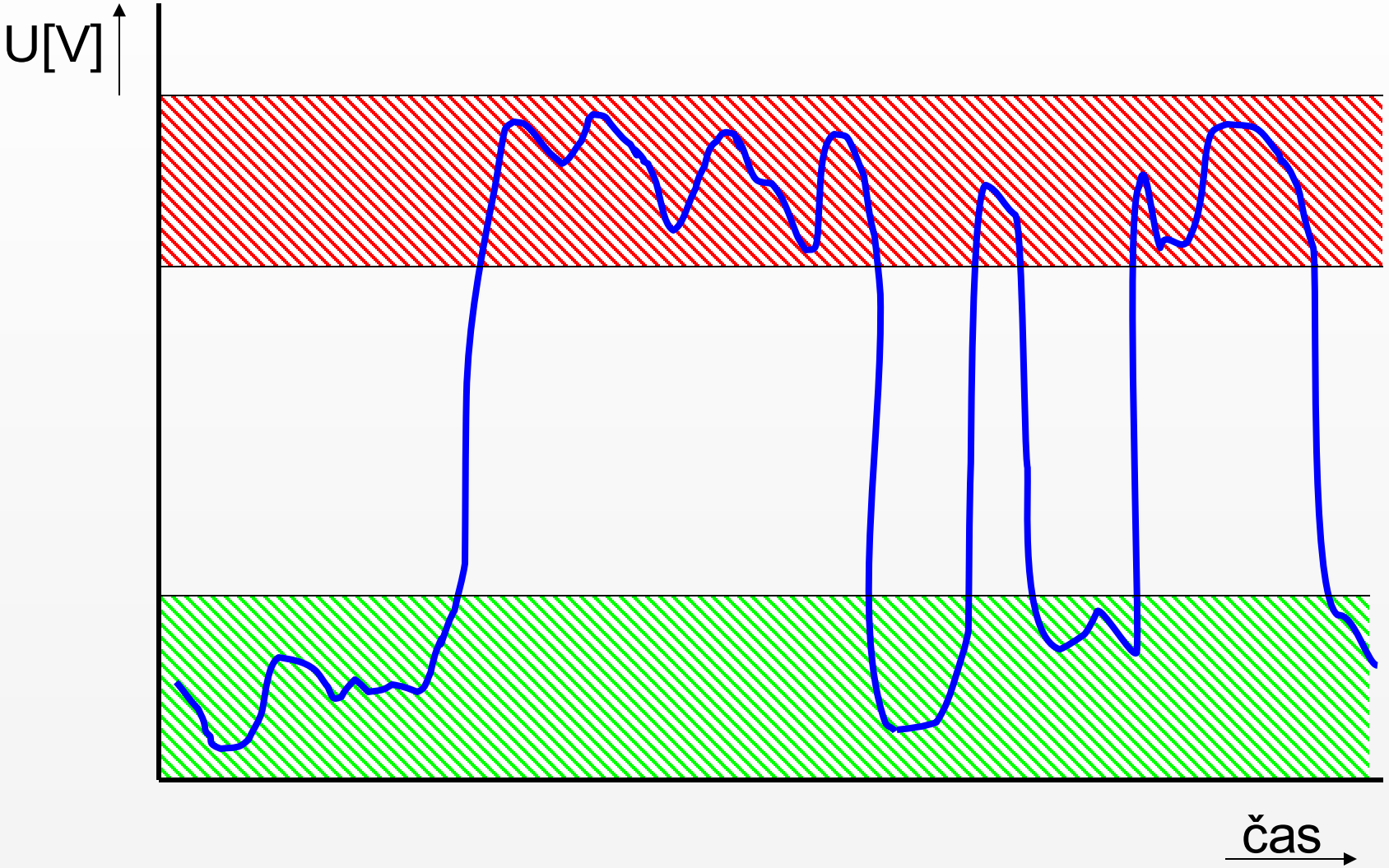
Sčítání čísel

14_{10} 01110_2

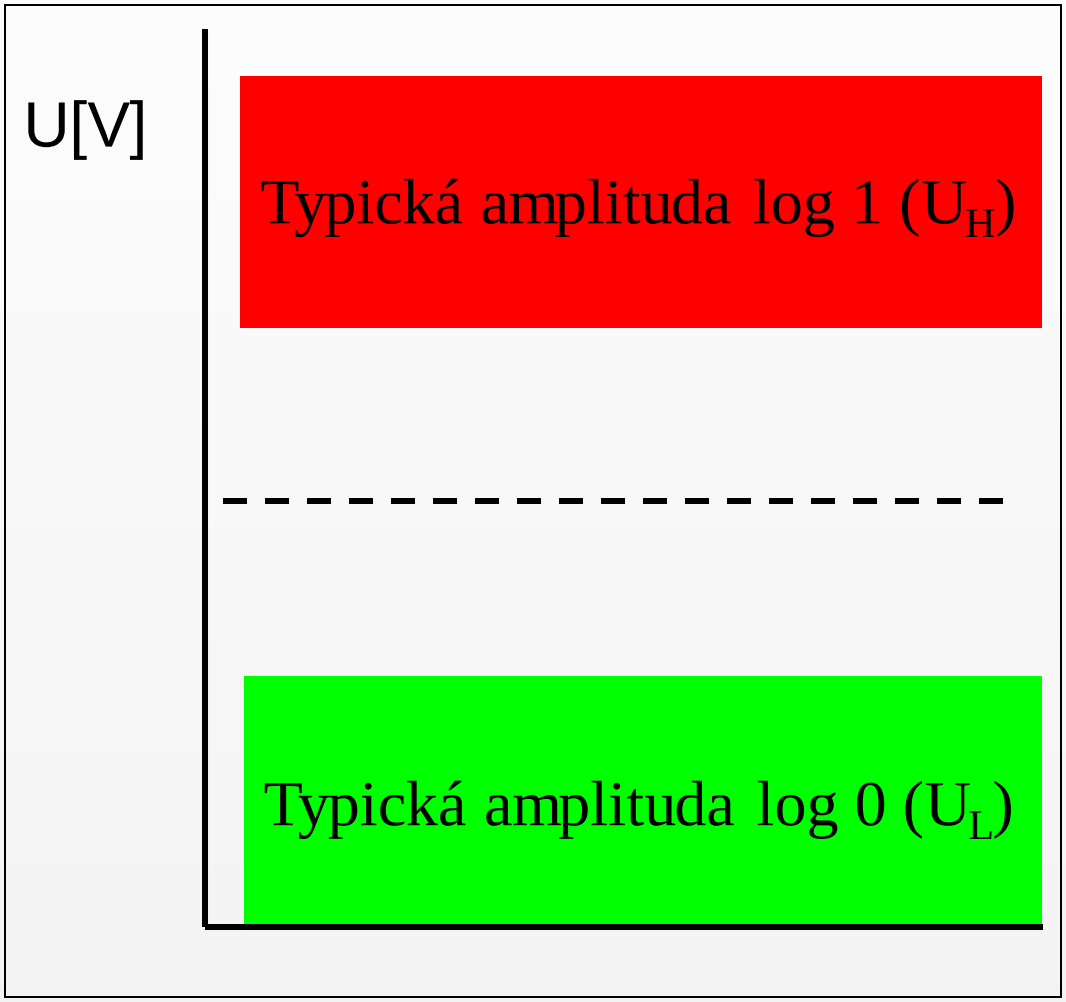
$\underline{\quad 6_{10}}$ $\underline{00110_2}$

20_{10} 10100_2

Číslicové obvody



Číslicové obvody



horní mez “1“

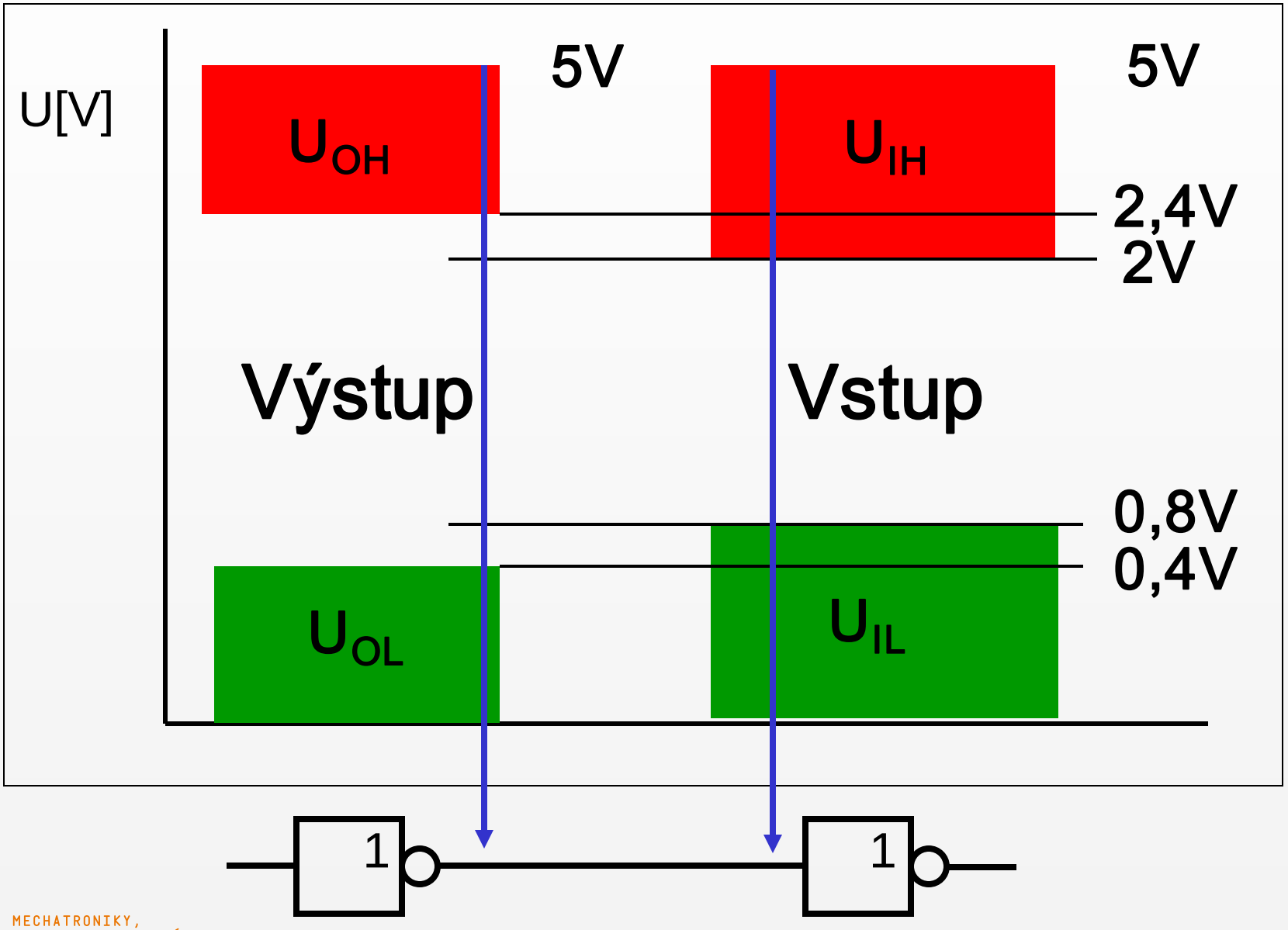
dolní mez “1“

Rozhodovací amplituda

horní mez “0“

dolní mez “0“

Číslicové TTL obvody



Logické stavy

Zjednodušeně lze popsat **dvěma stavy** (binární proměnné)

ZAPNUTO / VYPNUTO

PRAVDA / NEPRAVDA

Log.1 / Log.0

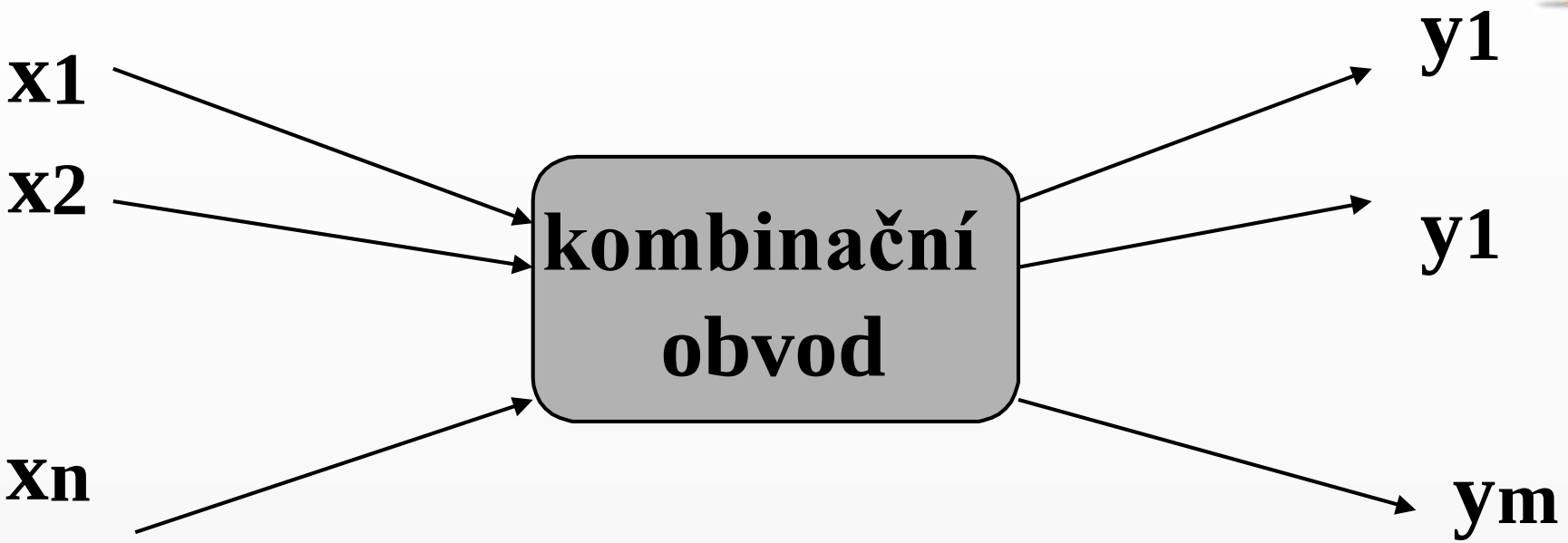
True / False

High / Low

Vysoké / nižší napětí

!! pozitivní / negativní logika (hodnota $X_A < X_B$)

Číslicové kombinační obvody



X – vstupní n-bitová proměnná

Y – výstupní m-bitová proměnná

Výstupní hodnoty ***Y*** určeny pouze aktuálními vstupními hodnotami ***X*** (reálně s určitým zpožděním)
– t.j. kombinační obvod nemá paměť předchozích vstupních hodnot či vnitřního stavu.

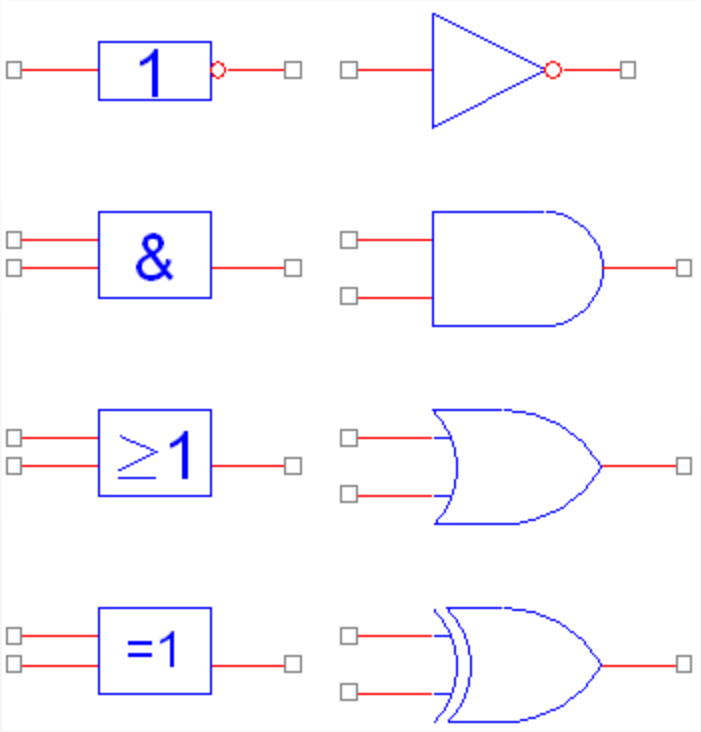
Značení logických hradel

invertor

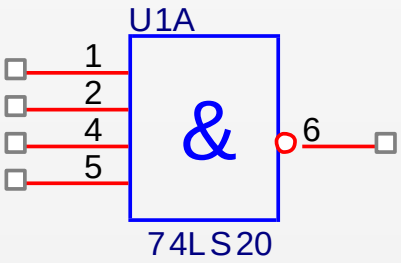
AND

OR

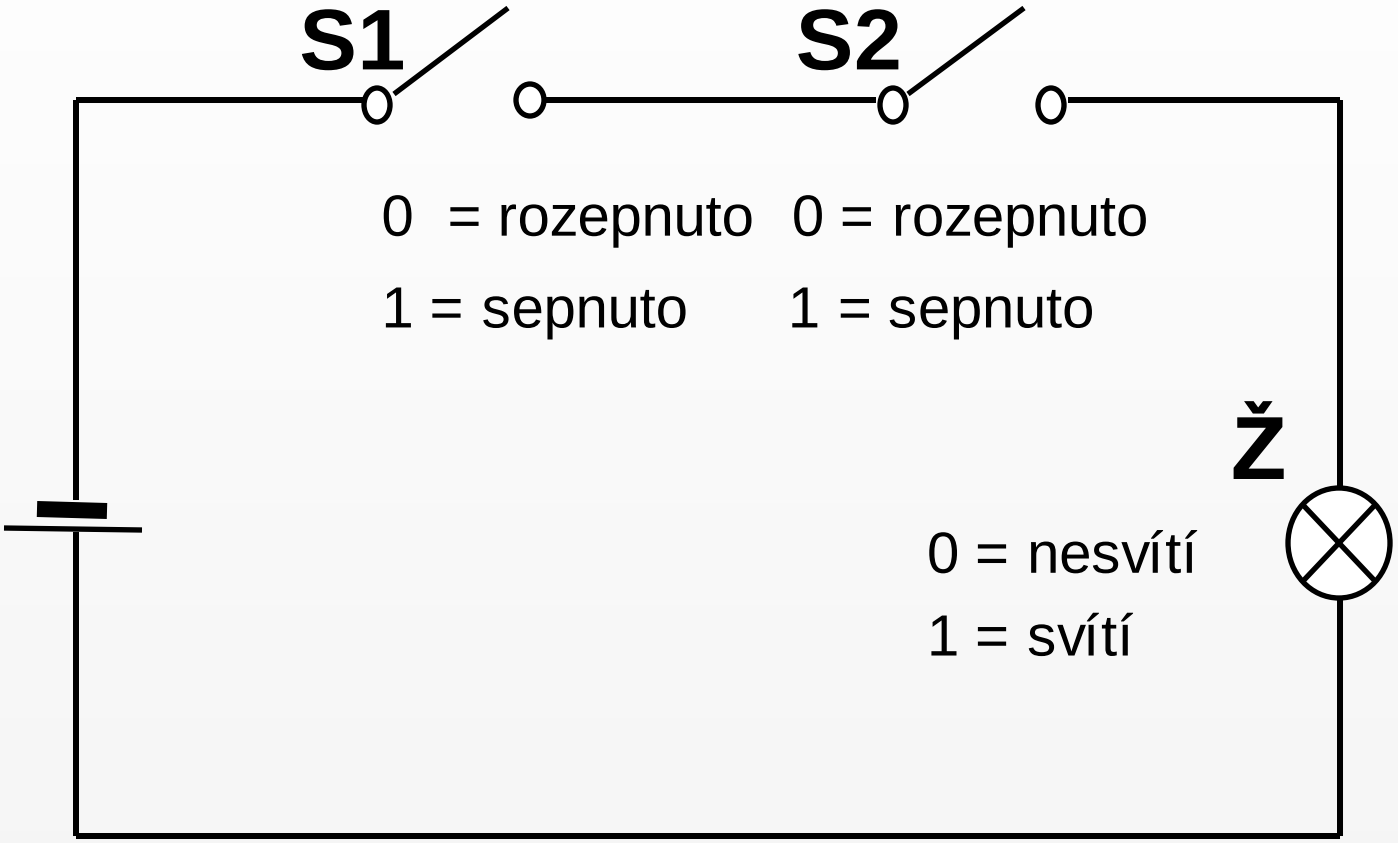
XOR



– vícebrany; společný vodič a napájecí svorky se nekreslí
– kroužek = negace (i na vstupu)



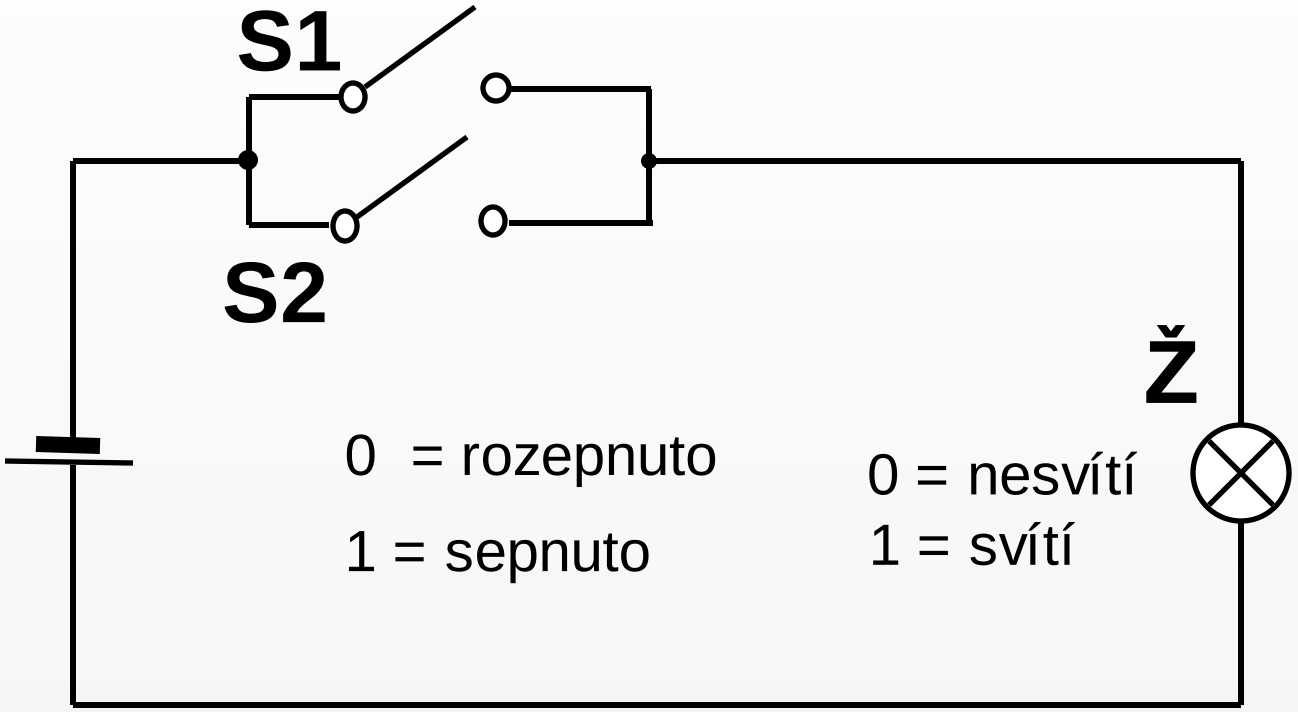
Logické operace



S1	S2	Ž
0	0	0
0	1	0
1	0	0
1	1	1

AND – logický součin

Logické operace



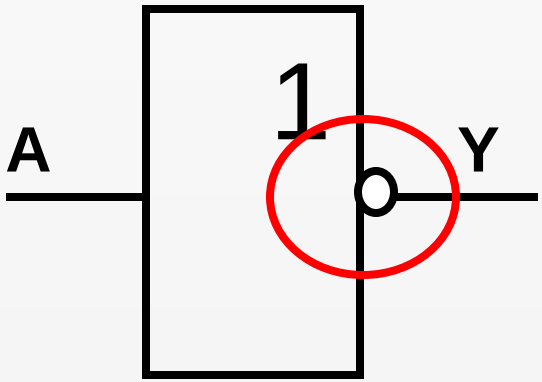
S1	S2	Ž
0	0	0
0	1	1
1	0	1
1	1	1

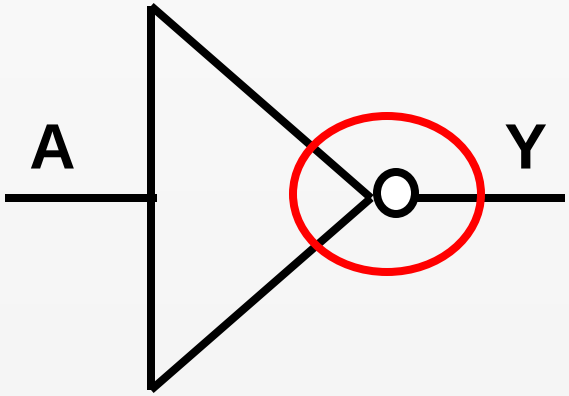
OR – logický součet

Základní logické funkce

Negace NOT, INVERT

$Y = \overline{A}$, Y = /A, Y = ‘A



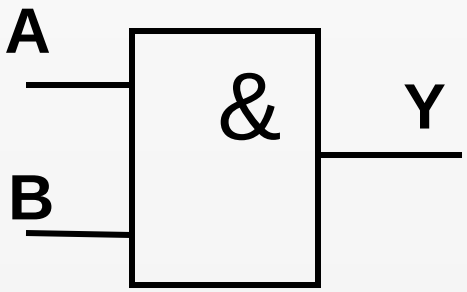


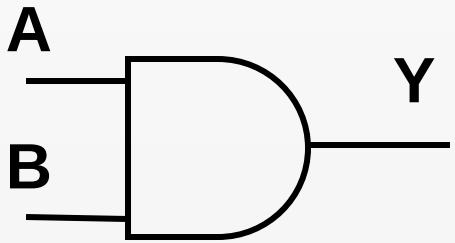
A	Y
0	1
1	0

Základní logické funkce

Logický součin AND

$Y = A.B$ $Y = AB$



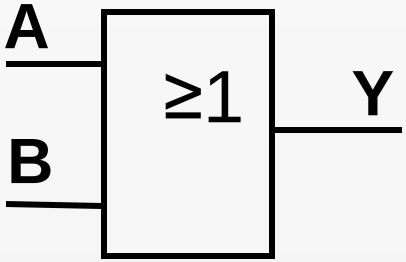


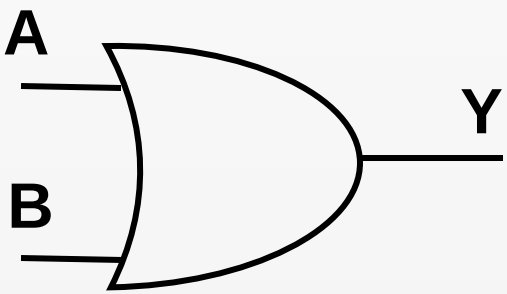
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Základní logické funkce

Logický součet OR

Y = A+B



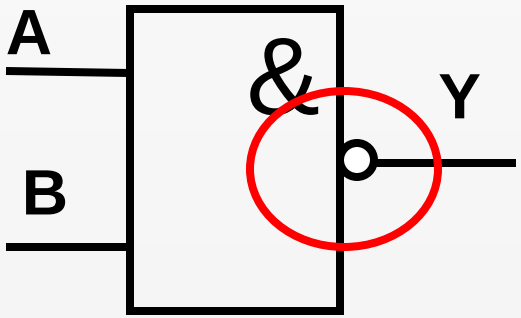


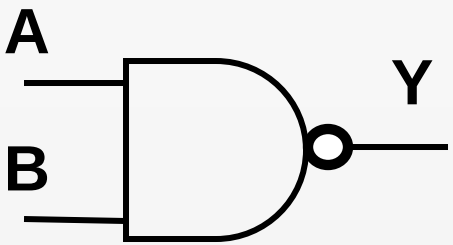
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Další logické funkce

Negovaný logický součin NAND

$$Y = \overline{A.B},\ Y = \overline{AB},\ Y = \text{‘}(A.B)$$



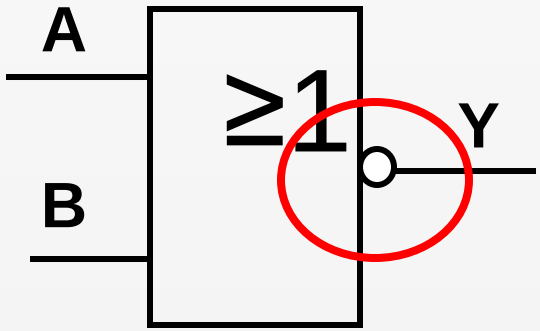


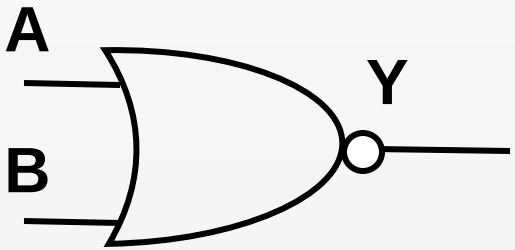
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Další logické funkce

Negovaný logický součet NOR

$$Y = \overline{A+B},\ Y = \text{‘}(A+B)$$



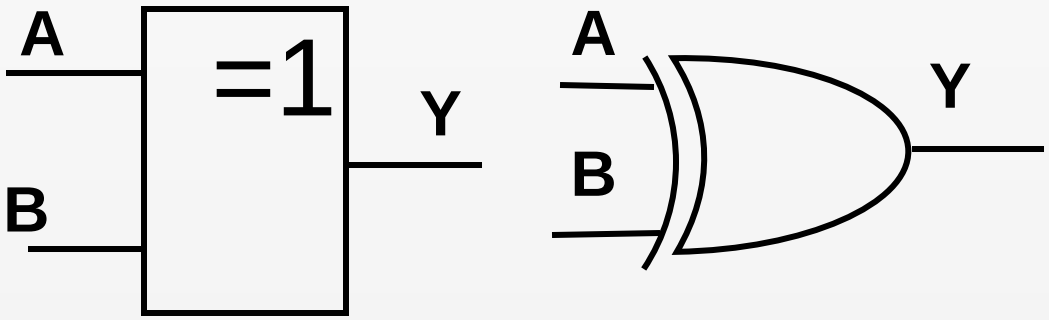


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Další logické funkce

Neekvivalence XOR *(výhradní OR)*

$$Y = A \oplus B$$

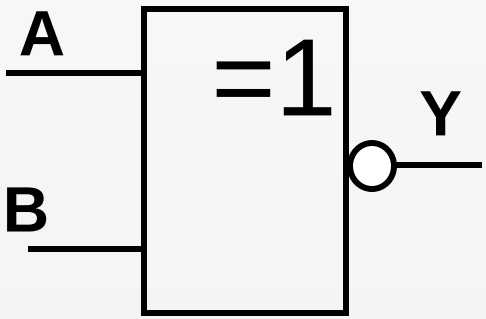


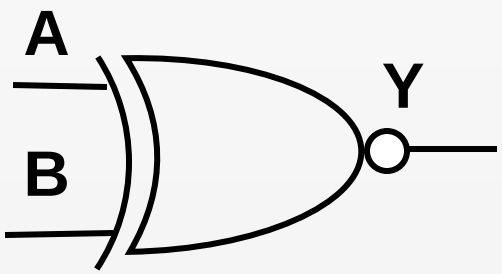
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Další logické funkce

ekvivalence XNOR

$$Y = \overline{A \oplus B}$$



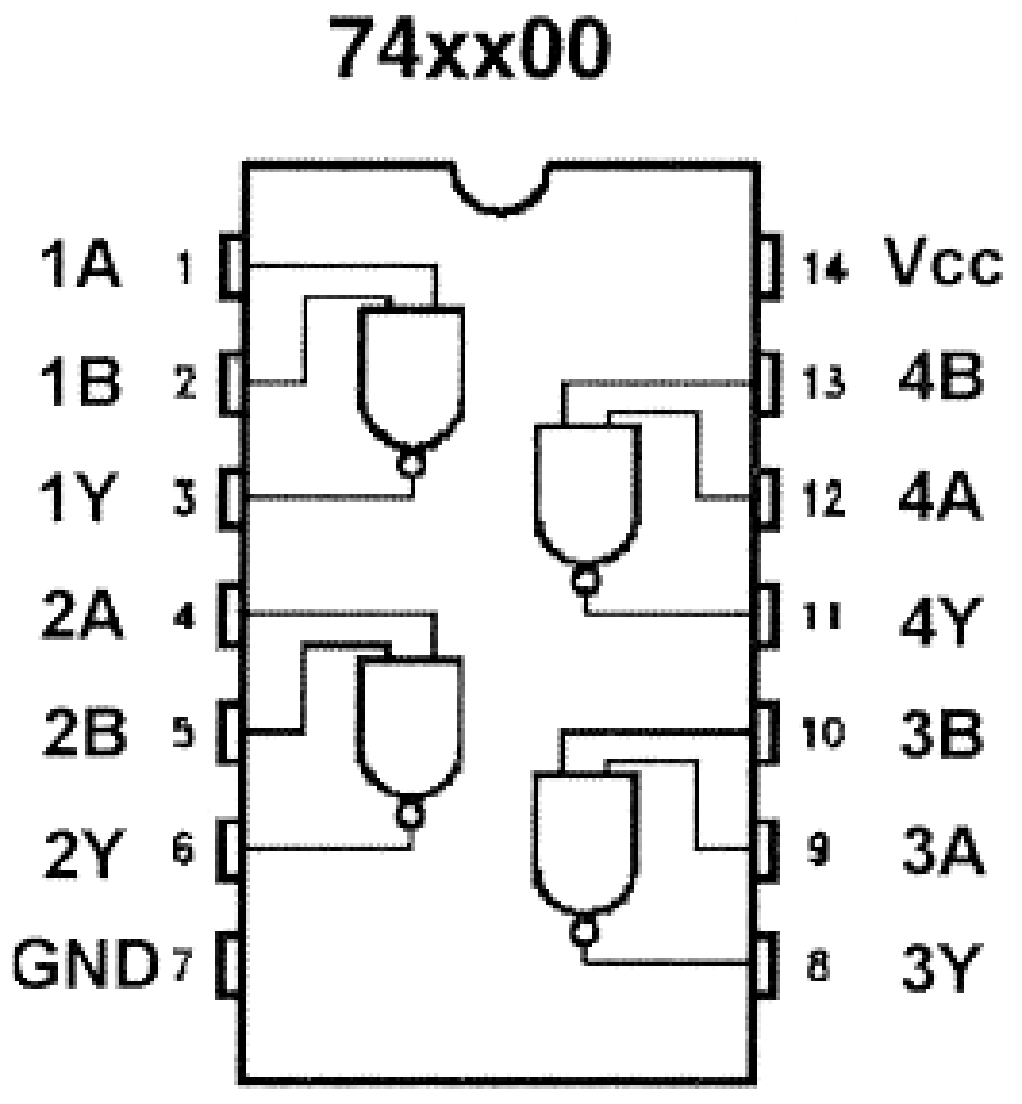


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Základní logické funkce

A	B	NOT	AND	NAND	OR	NOR	XOR	XNOR
0	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	1	0
1	0		0	1	1	0	1	0
1	1		1	0	1	0	0	1

Realizace hradel



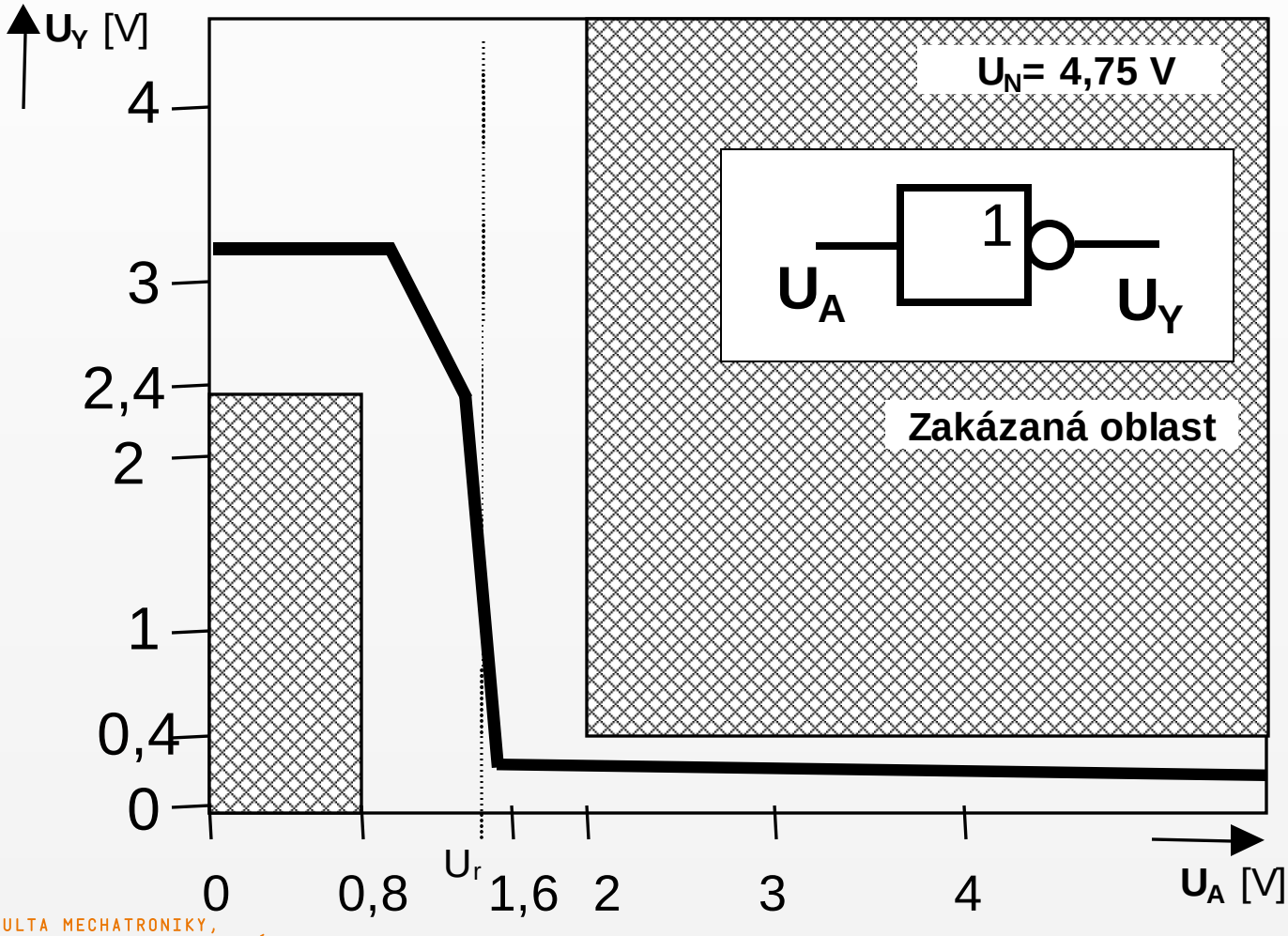
DIP (DIL) -14 , -16, -24, -28, -40

SO -14 ,-16, -24, -28, -40

Napájení !!

Převodní charka IO

Převodní charakteristika TTL invertoru



0,8 V– $U_{A0\ MAX}$

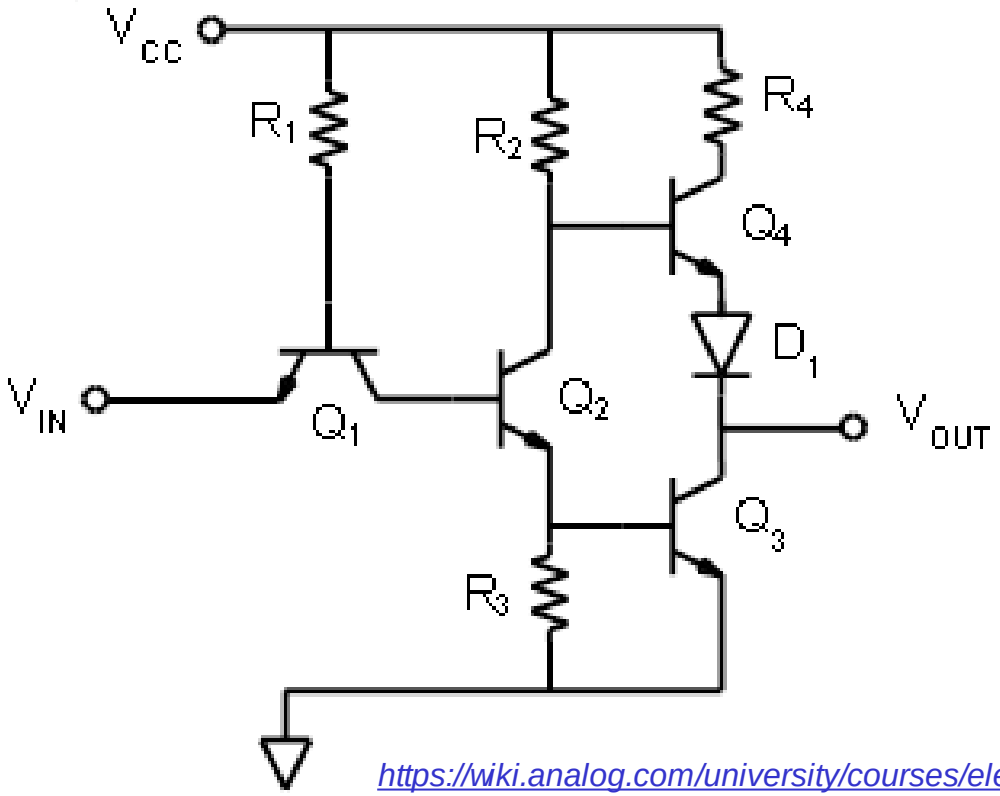
2,0 V – U_{A1MIN}

2,4 V – U_{Y0MAX}

0,4 V – U_{Y1MIN}

U_r – překlápěcí úroveň vstupního napětí

Převodní charka IO

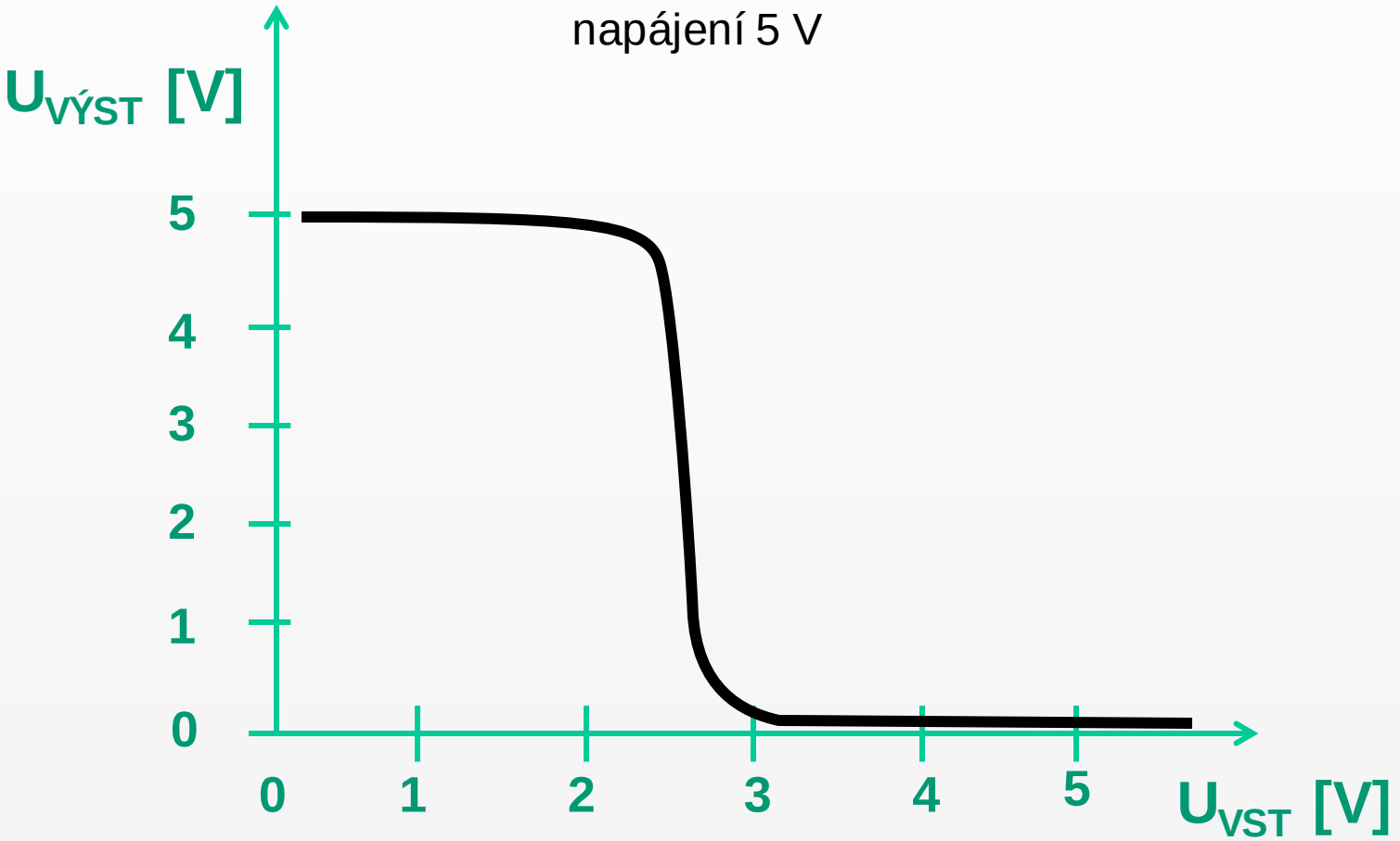


https://wiki.analog.com/university/courses/electronics/electronics-lab-27

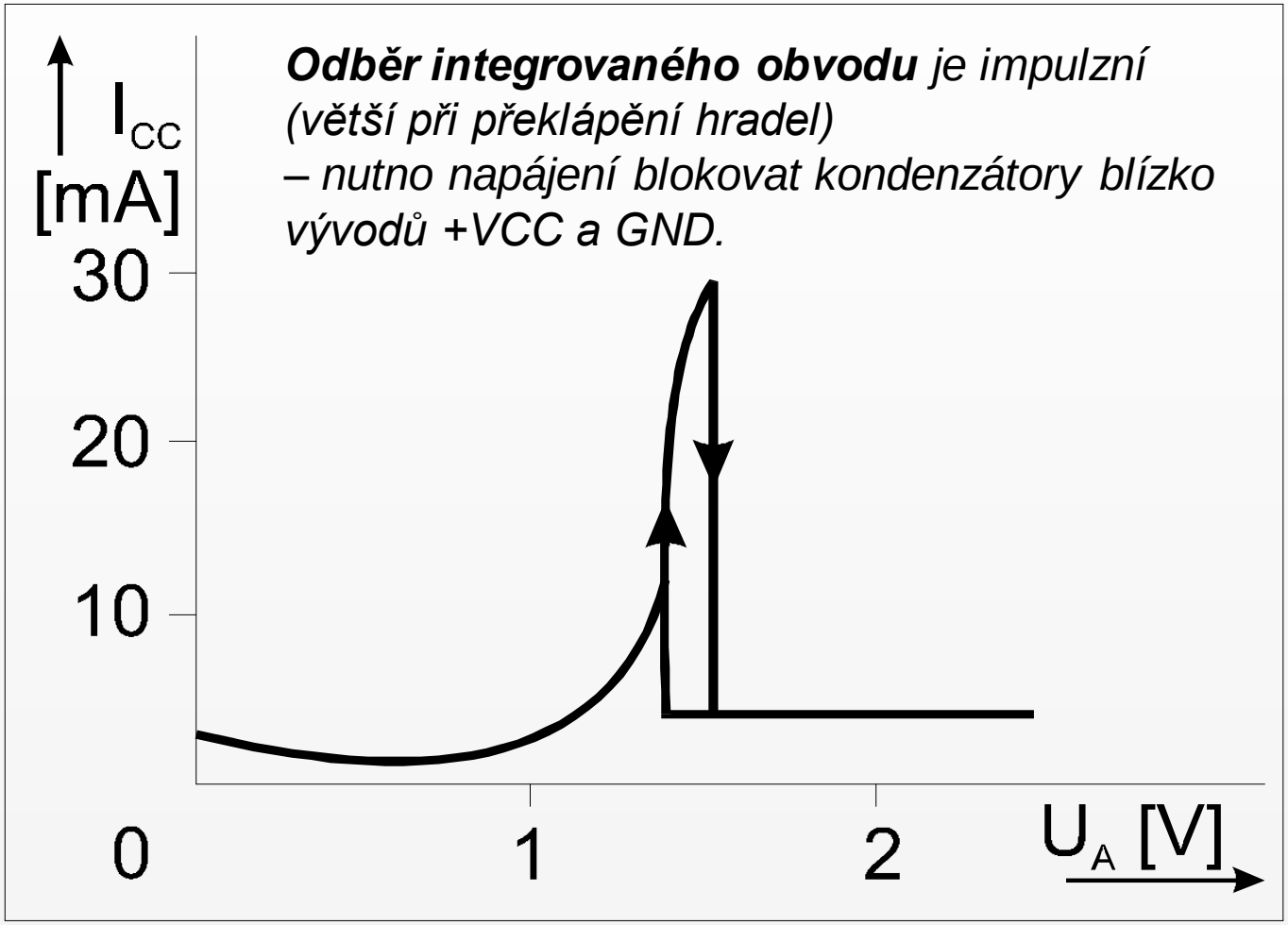
napětí

0,4
0
0
0,8
U_r
1,6
2
3
4
U_A [V]

Převodní charka IO



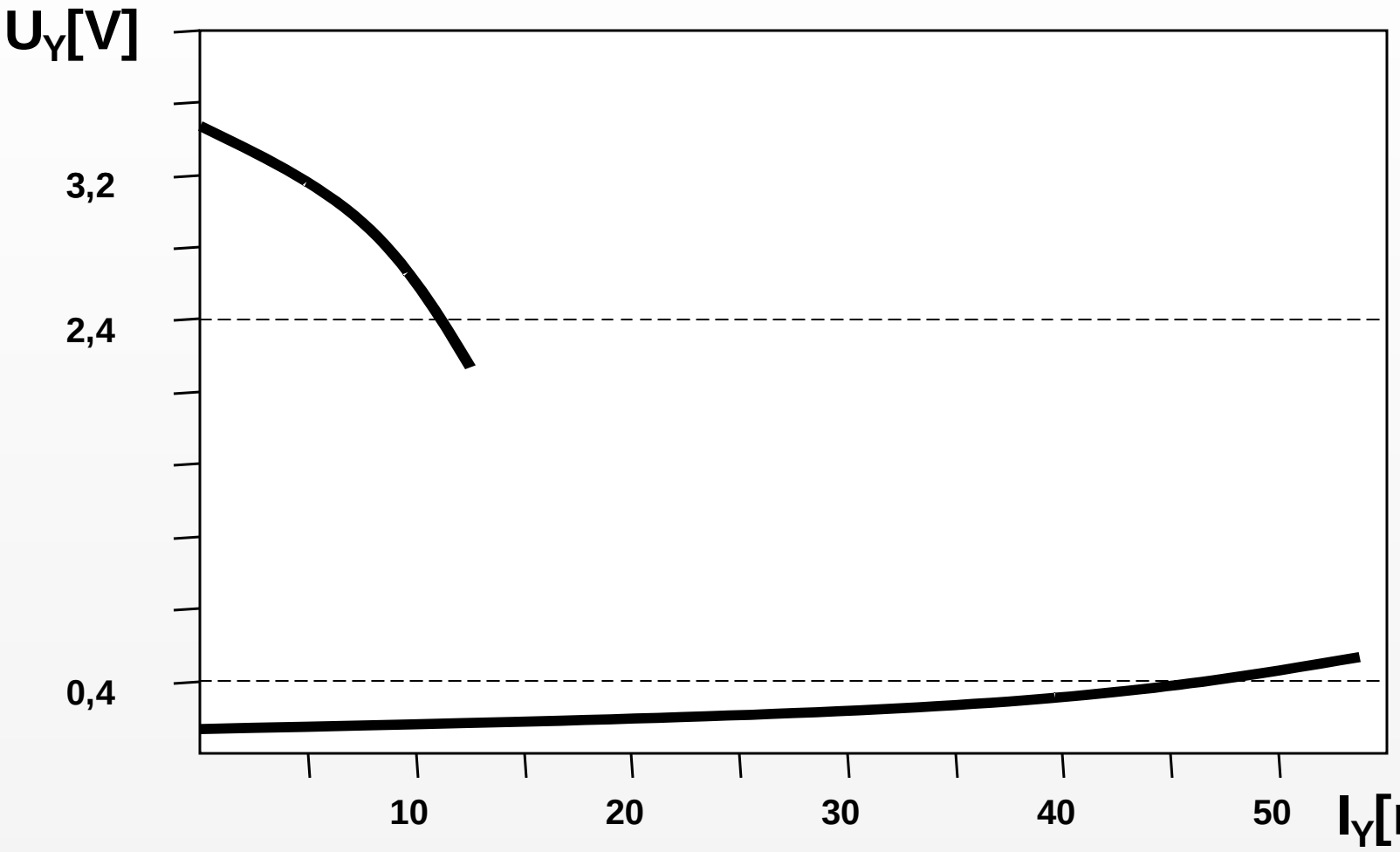
Odběrová ch. invertoru TTL



U_A - vstupní napětí na invertoru,

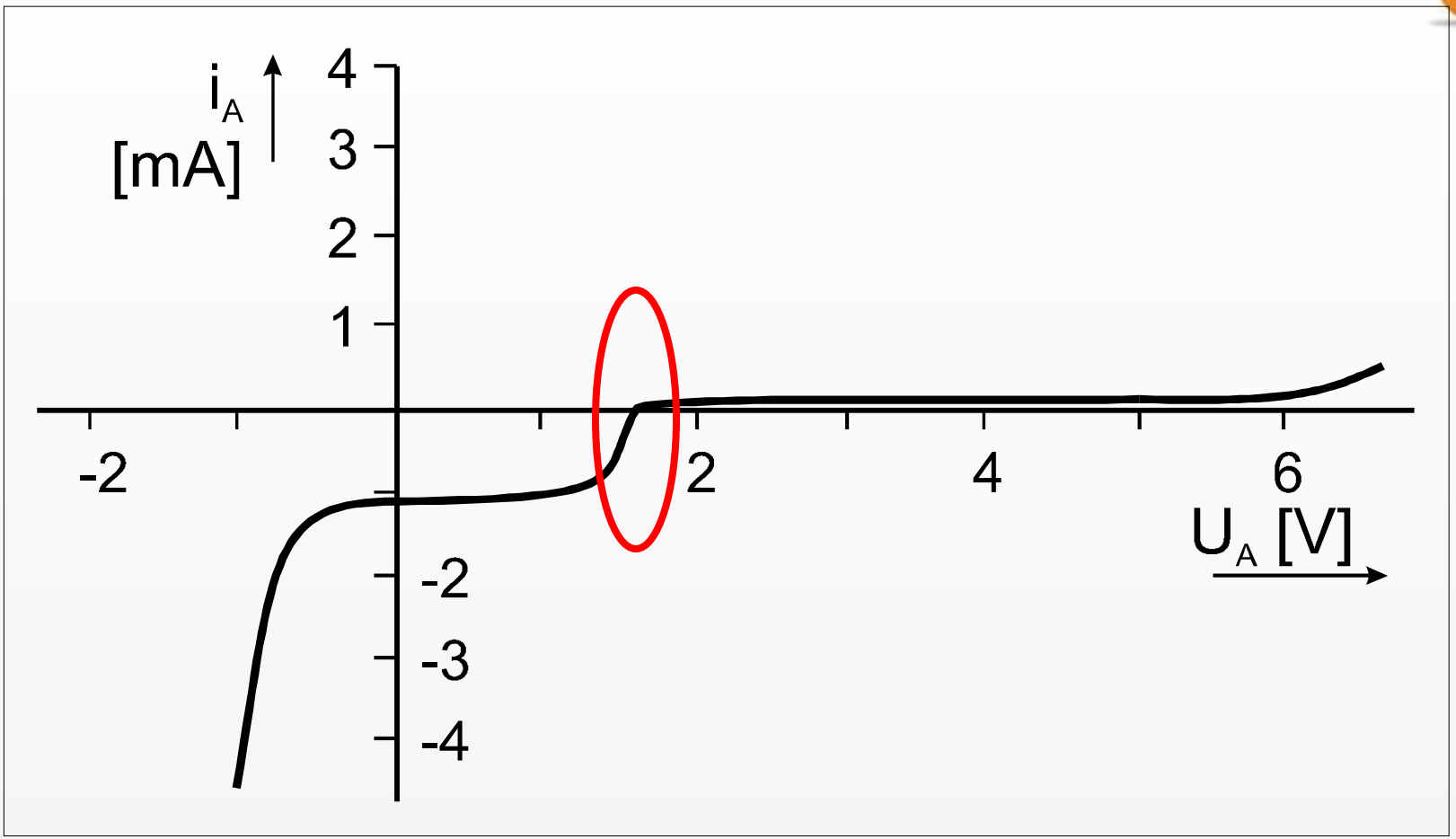
I_{cc} – proud odebíraný z napájecího zdroje

Zatěžovací ch. TTL hradla



zatěžování výstupu nastaveného do **log. 1** a do **log. 0**

Vstupní ch. TTL hradla



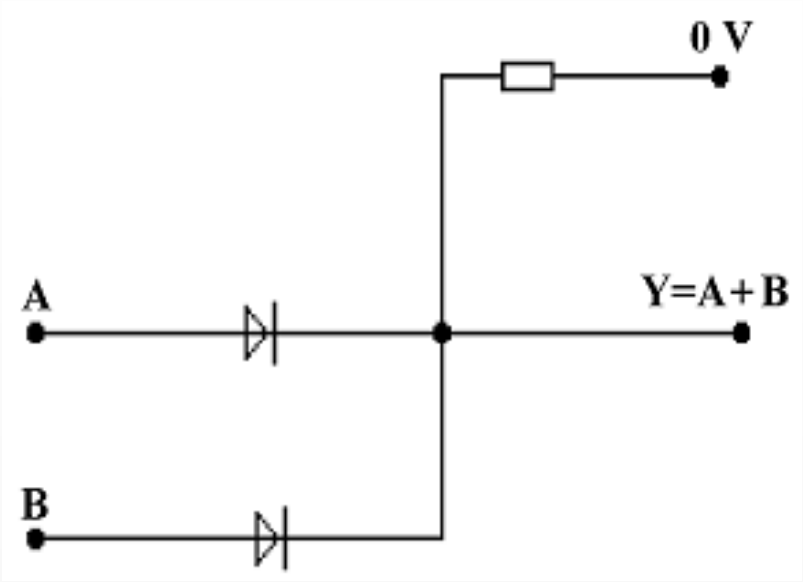
!! **1,5 V** vstupního napětí pro **nulový** vstupní proud. Tato hodnota se ustálí na nezapojených vstupech hradla !!

Ošetření nepoužitých vstupů

Ze vstupní charakteristiky vyplývá, že **není vhodné ponechávat** u logiky TTL **nezapojené vstupy**…
(nezapojený vstup je zpravidla interpretován jako log. 1. Napětí, které se ustálí na tomto vstupu však leží v zakázaném pásmu a existuje nebezpečí špatné interpretace tohoto napětí obvodem).

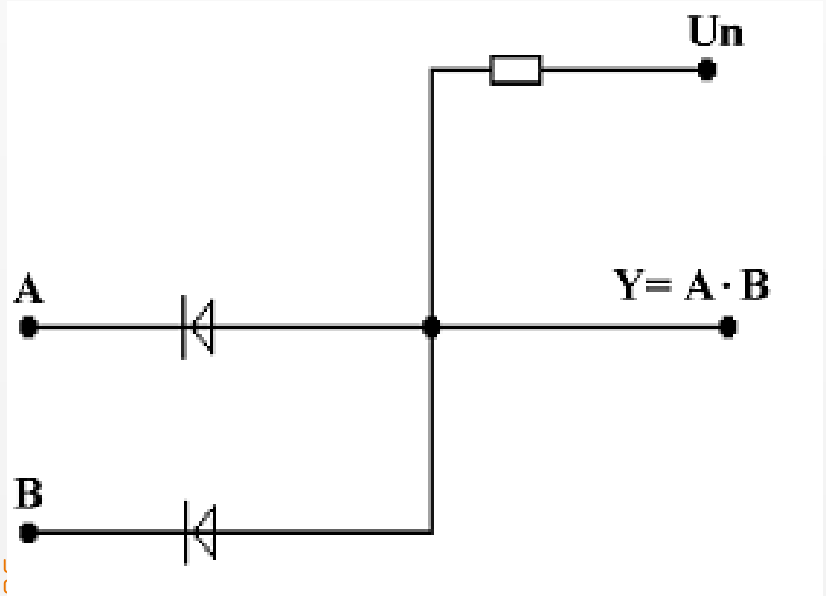
U členů **NAND** je možno nepoužité vstupy paralelně připojit ke vstupům použitým nebo je připojit na úroveň H.
U členů **NOR** je třeba nepoužité vstupy připojit na úroveň L.

Technologie výroby IO



OR

Diodová logika



AND

Bez aktivního prvku nelze realizovat negaci.
Nelze řadit mnoho obvodů za sebou bez obnovení správných log. úrovní.

⇒ **diodově-tranzistorová logika**

Provedení TTL obvodů

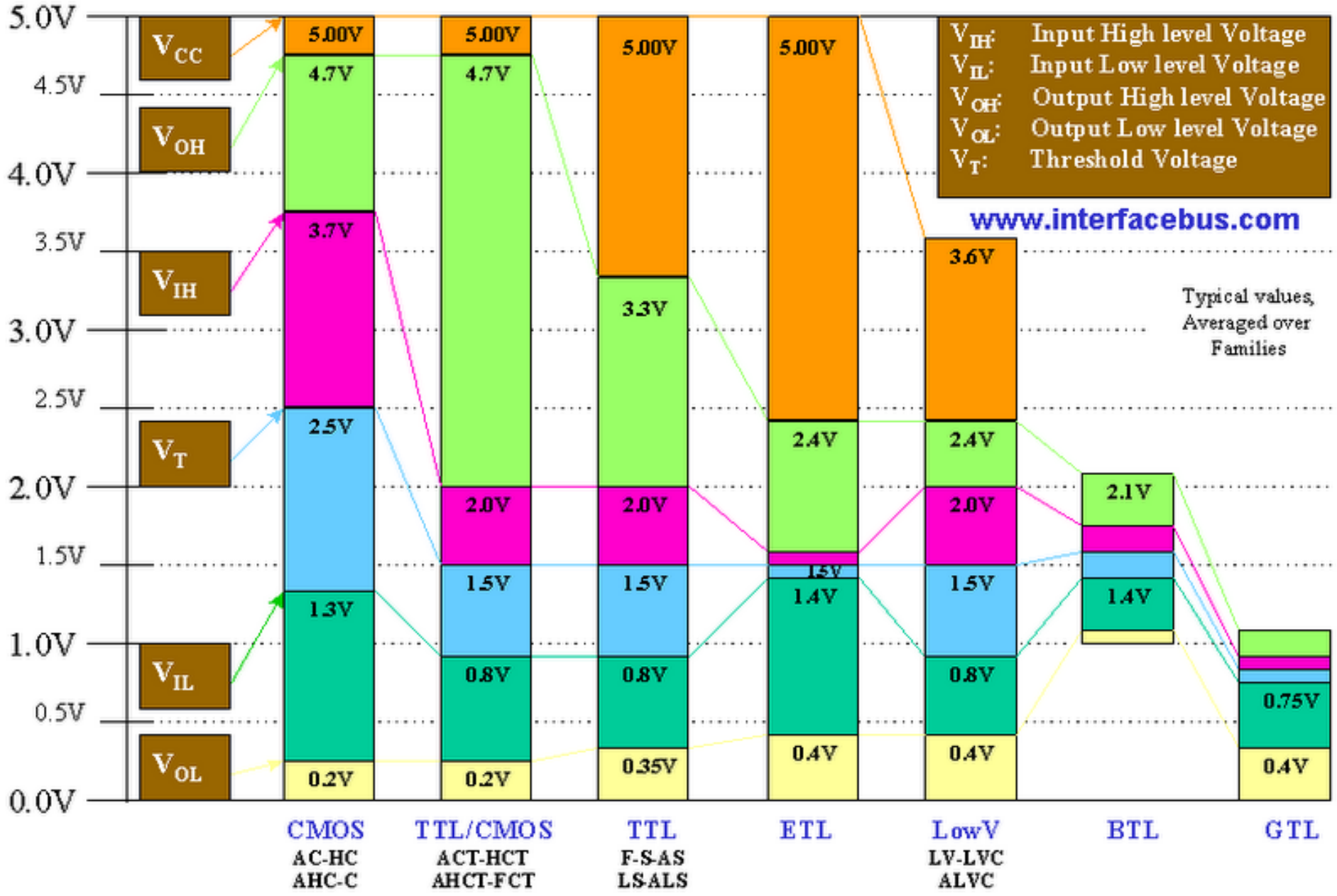
technologické varianty těchto obvodů:

- H – rychlá CMOS (dříve F)
- N – normální
- C – CMOS
- T – kompatibilita s TTL
- L – nízká spotřeba
- S – se Schotkyho diodami (zabraňují saturaci transistorů)
- LS, ALS – nízká spotřeba + Schotkyho diody.

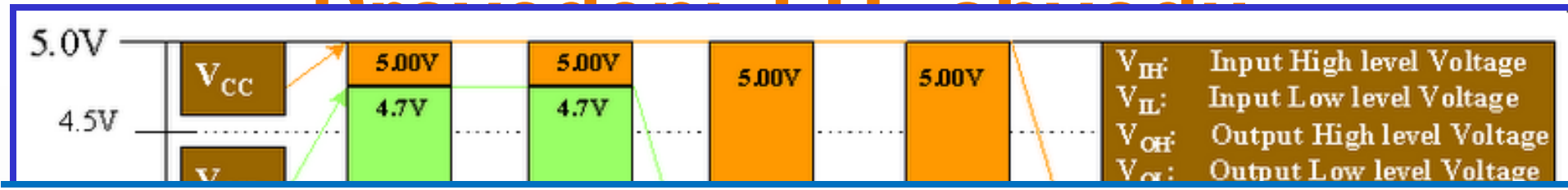
(např. 74**ALS**00, 74**HCT**04)

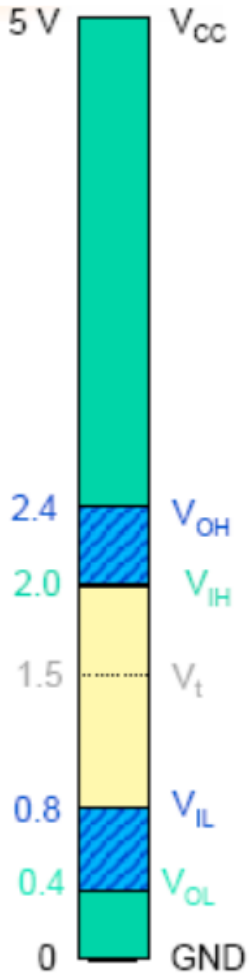
(viz. např. www.ti.com)

Propustná TTL obvodů

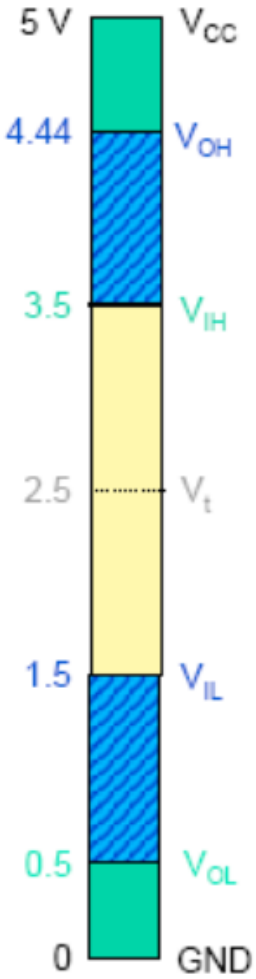


Propojování TTL obvodů



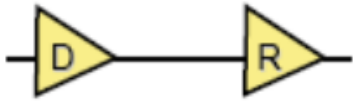


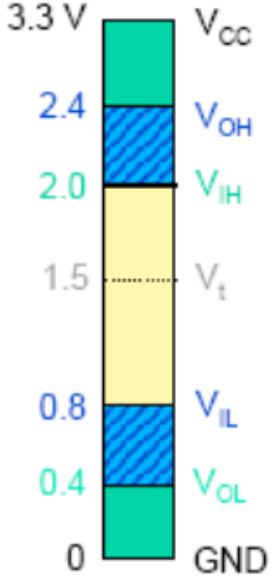
5-V TTL



5-V CMOS

Is V_{OH} higher than V_{IH}?
Is V_{OL} less than V_{IL}?

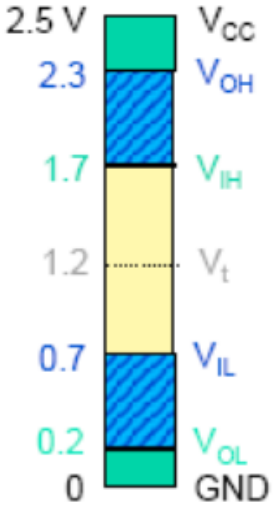




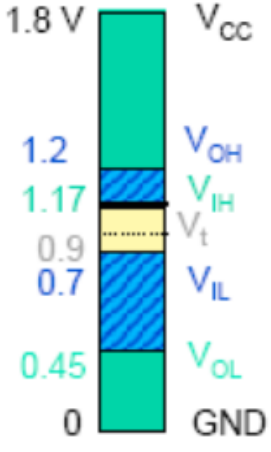
3.3-V LVTTL

R / D	5TTL	5CMOS	3LVTTL	2.5CMOS	1.8CMOS
5TTL	Yes	No	Yes *	Yes*	Yes*
5 CMOS	Yes	Yes	Yes*	Yes*	Yes*
3 LVTTL	Yes	No	Yes	Yes*	Yes*
2.5 CMOS	Yes	No	Yes	Yes	Yes*
1.8 CMOS	No	No	No	No	Yes

* Requires V_{IH} Tolerance

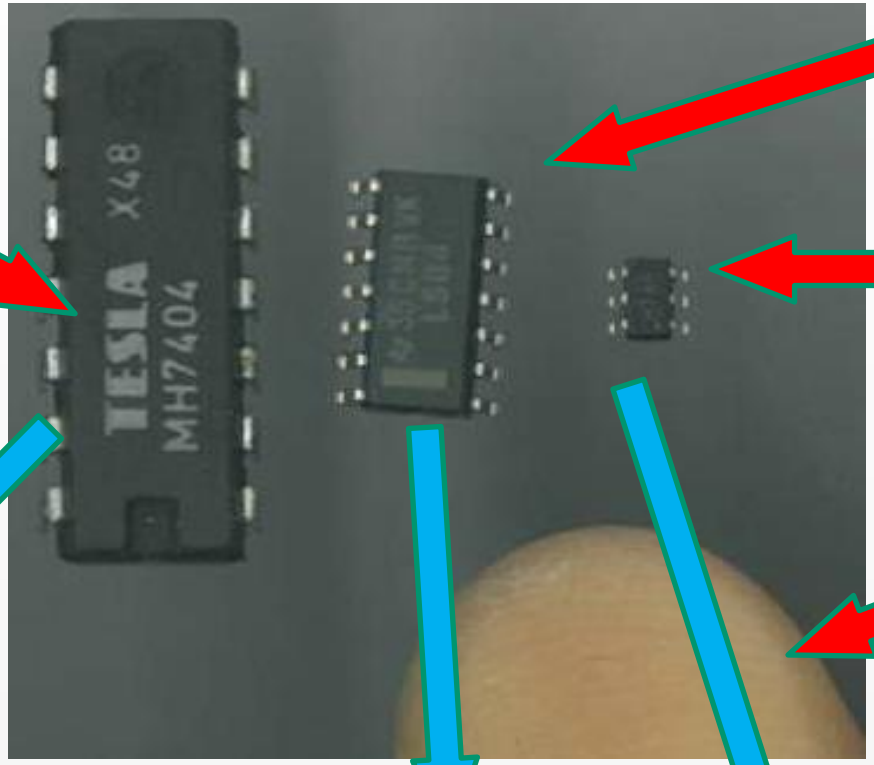


2.5-V CMOS



1.8-V CMOS

Logické obvody - provedení



Pouzdro SO

Pouzdro DIL

Pouzdro PDSO

Ukazováček (můj)

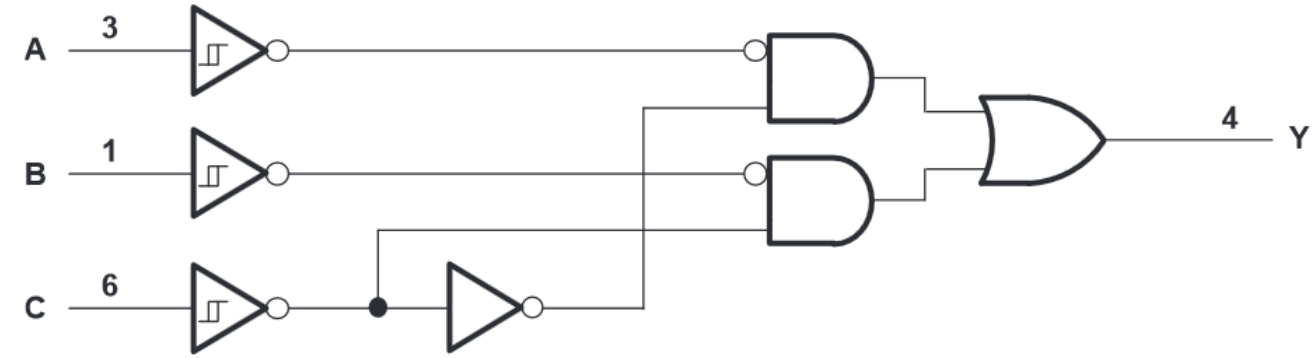
6× INVERTOR

6× INVERTOR

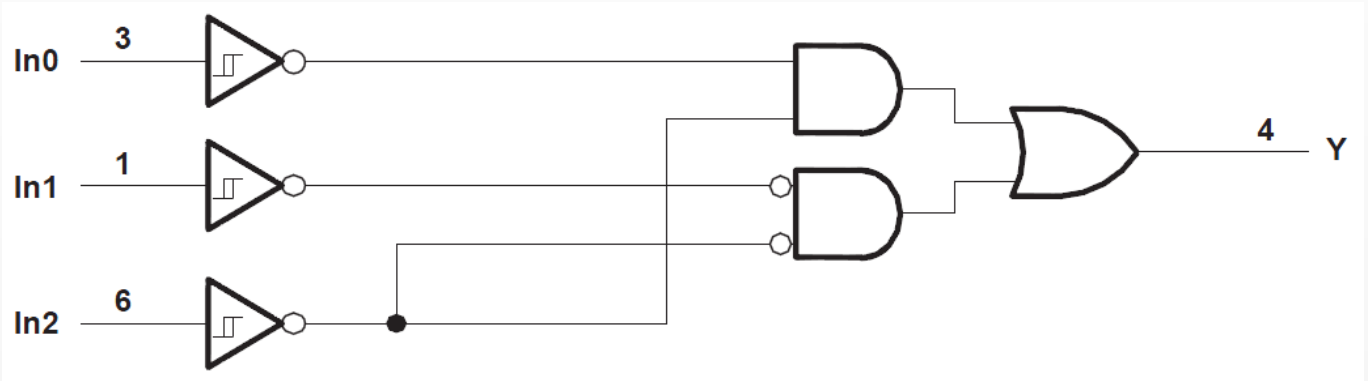
???????

Logické obvody - provedení

sn74aup1t97

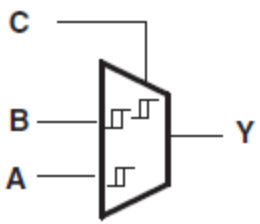


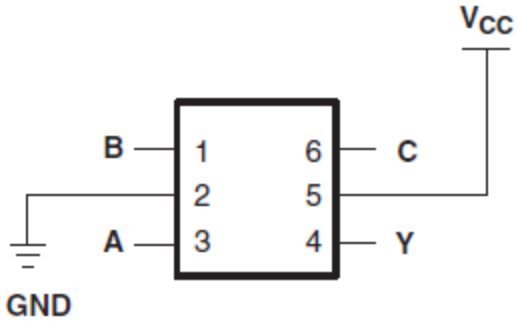
sn74lvc1g57



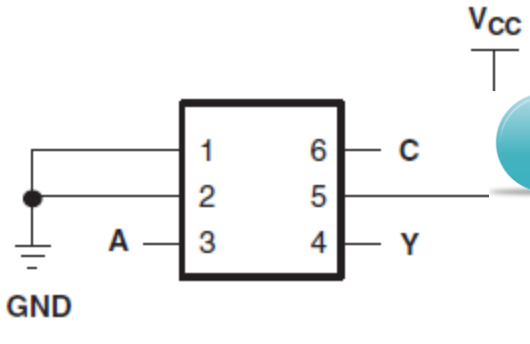
Datasheet TI: sn74aup1t97.pdf, sn74lvc1g57.pdf

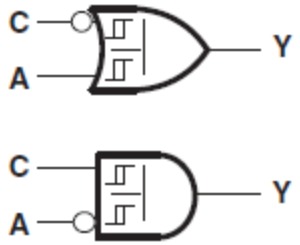
?????? ***K čemu to je??***

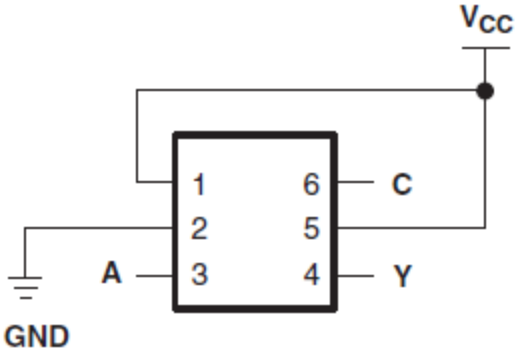

C
B
A
Y

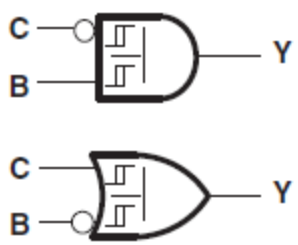

Vcc
B
A
1
2
3
6
5
4
C
Y
GND

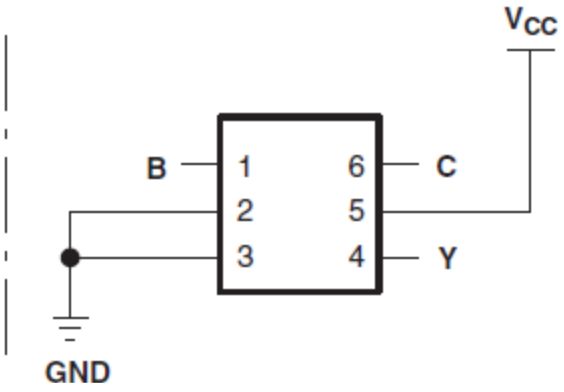

C
A
Y


Vcc
1
2
3
6
5
4
C
A
Y
GND

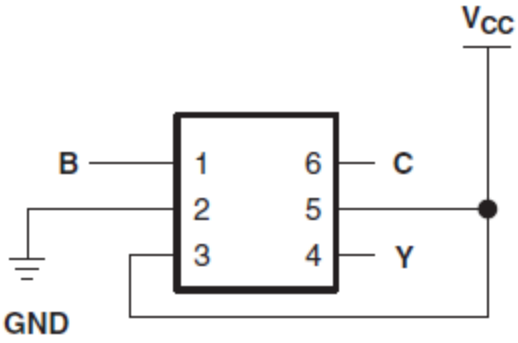

C
A
Y
C
A
Y


Vcc
1
2
3
6
5
4
C
A
Y
GND

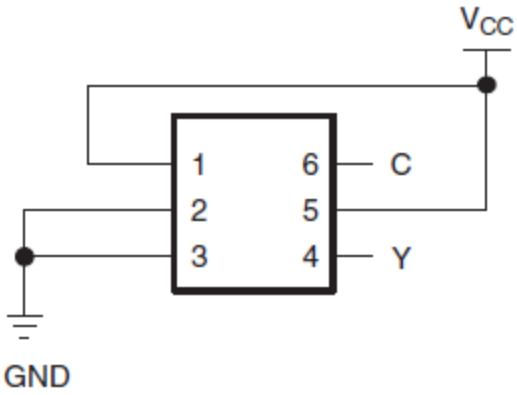

C
B
Y
C
B
Y

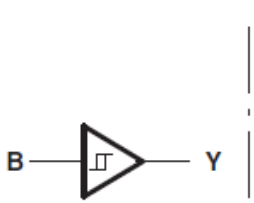

Vcc
B
1
2
3
6
5
4
C
Y
GND

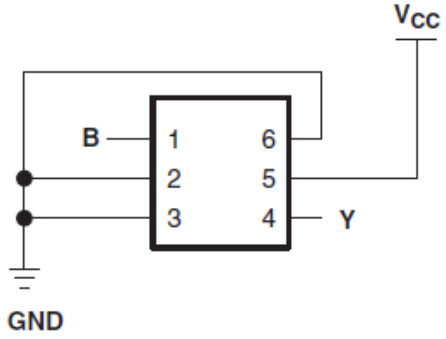

C
B
Y


Vcc
B
1
2
3
6
5
4
C
Y
GND

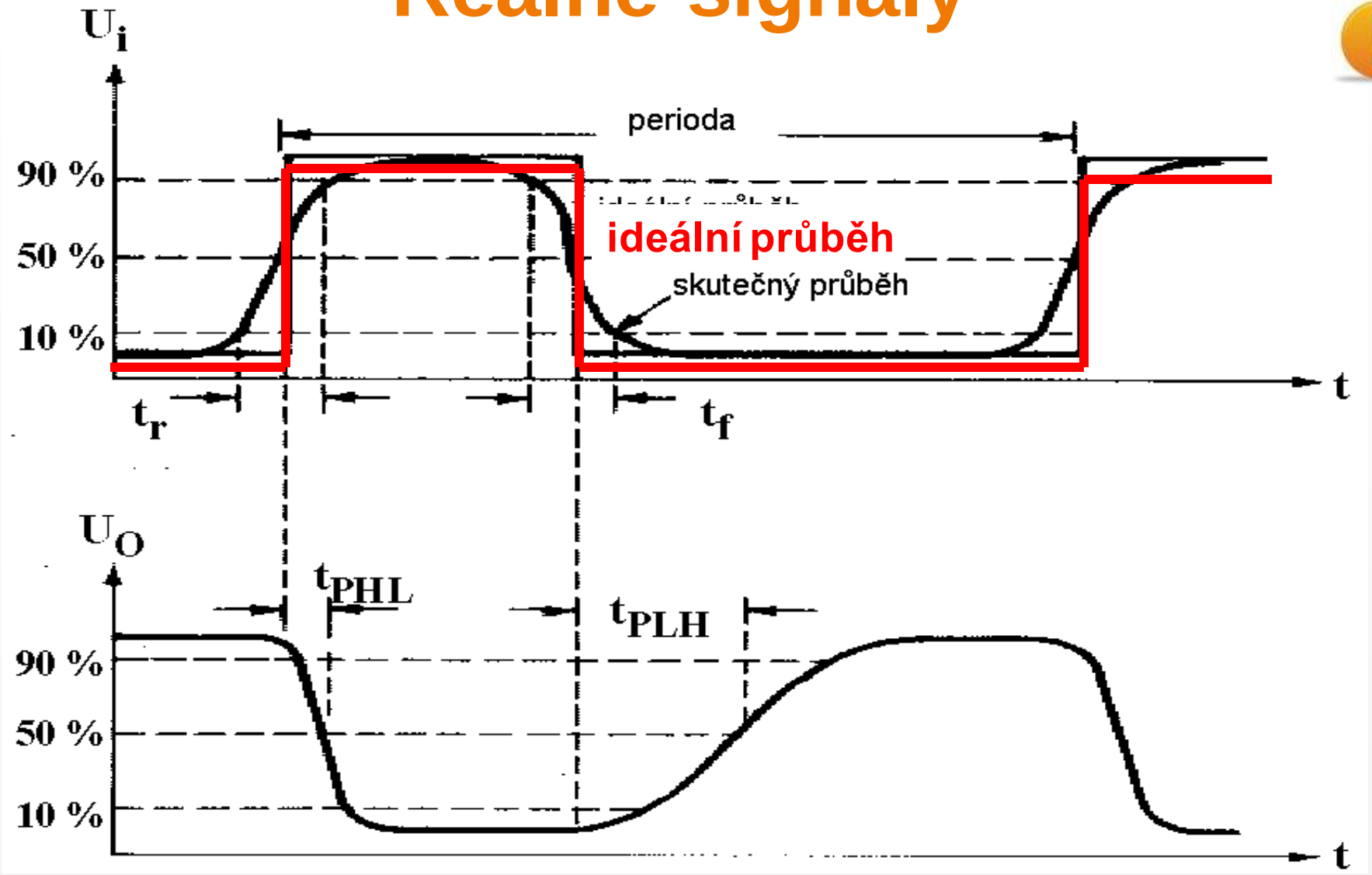

C
Y


Vcc
1
2
3
6
5
4
C
Y
GND


B
Y


Vcc
B
1
2
3
6
5
4
Y
GND

Reálné signály



Zpoždění signálu při průchodu hradlem (měřeno na úrovni 50 % ustálené změny) :

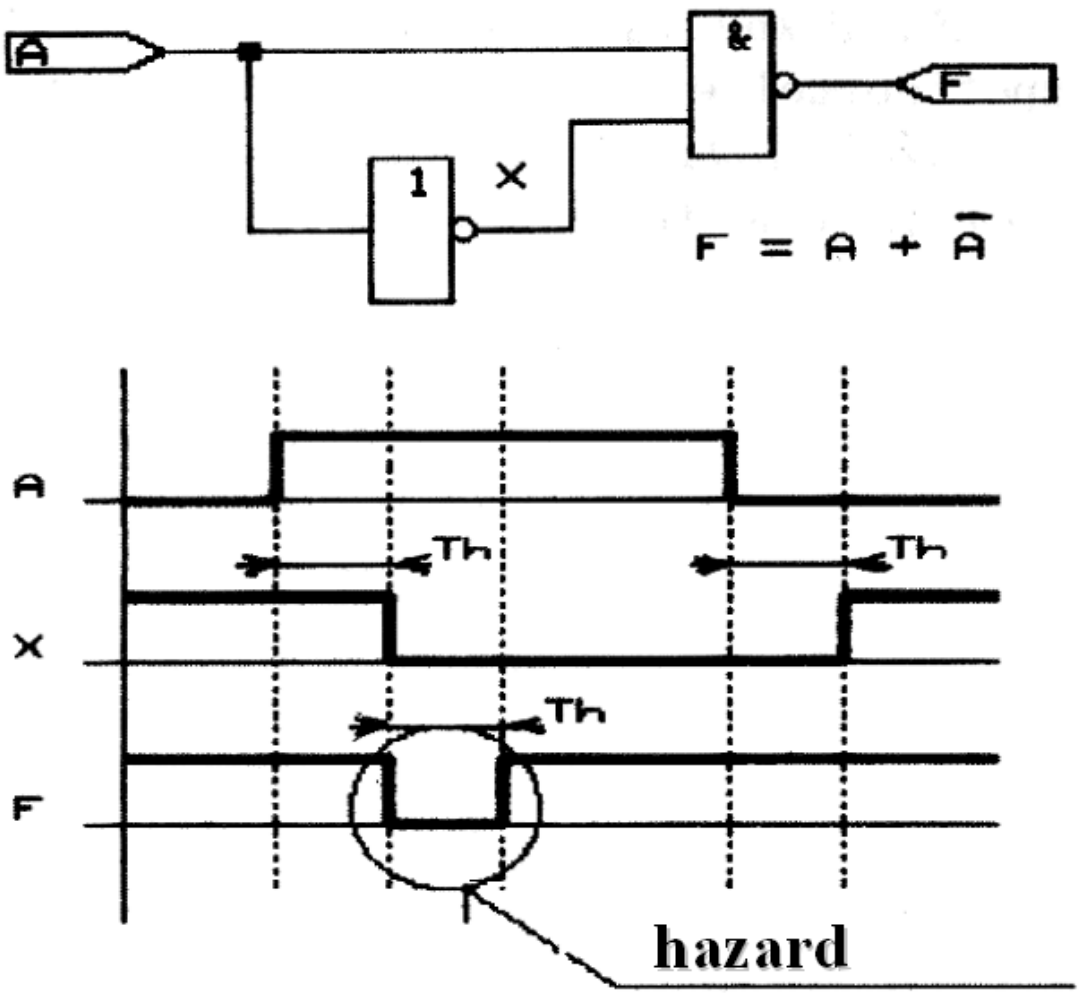
t_{PHL}, t_{PLH} ... při změně výstupu H → L resp. L → H : 10^{-9}... 10^{-7} s (dle technologie)

Hazardy

Statický logický hazard

Při změně vstupních proměnných se krátce změní úroveň (zákmit) na výstupu, na kterém se zrovna měnit nemá – způsobeno zpožděním logických obvodů.

Na nežádoucí impulz může reagovat připojený sekvenční obvod (např. čítač).



CMOS obvody

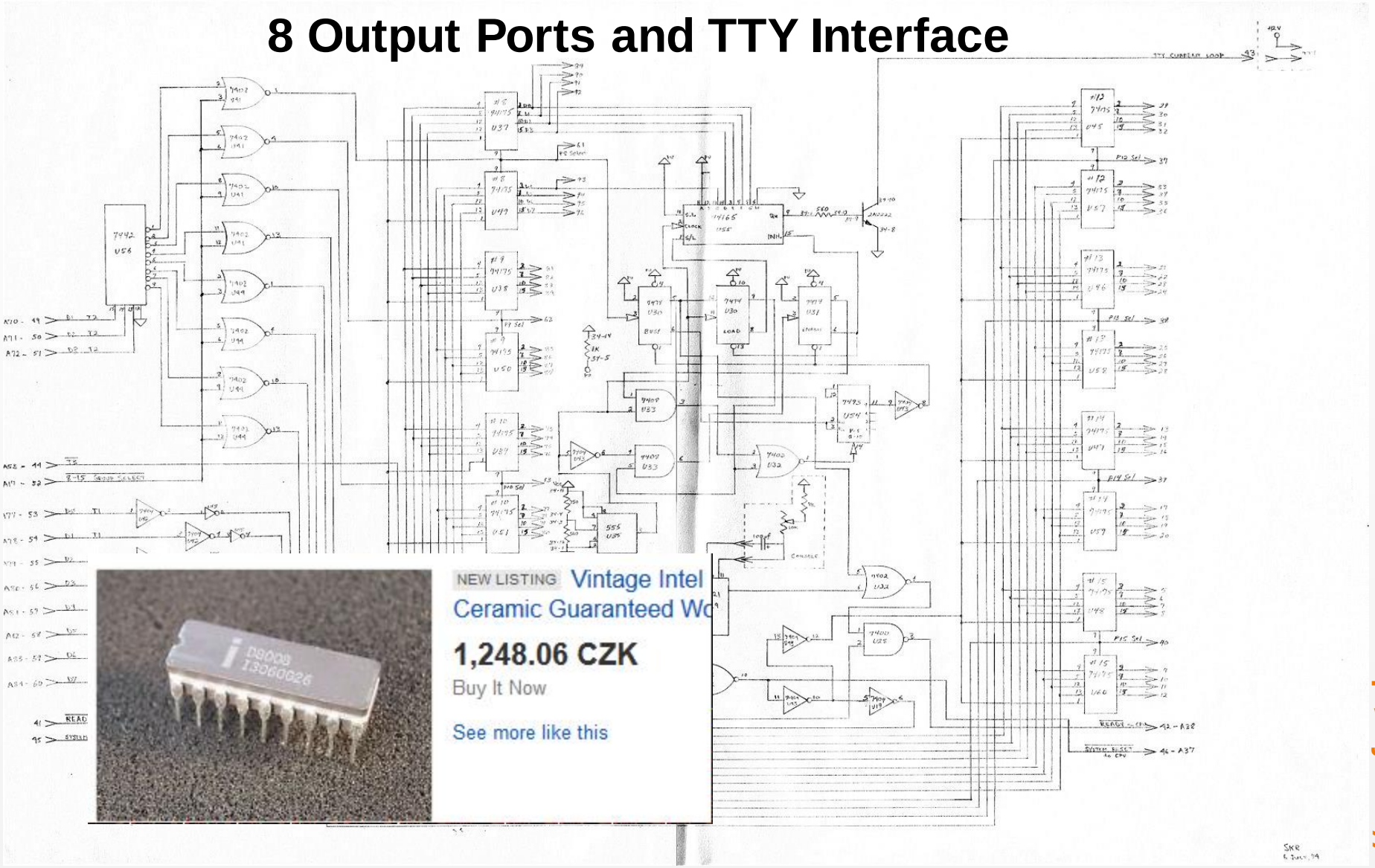
- V současnosti nejběžnější technologie číslicových obvodů.
- Relativně velké odběrové maximum při překlápění
- Umožňuje vysokou hustotu integrace.
- Maximální počet vstupů do jednoho hradla je zpravidla „2“.
  Větší počet vstupů do jednoho hradla je nevýhodný z hlediska dynamických vlastností takovéhoto zapojení a bývá nahrazován kaskádou dvouvstupových hradel.
- Hodnotu napájecího napětí je možno volit v rozsahu od cca 1,5 V do 15 V.
- Volba napájecího napětí ovlivňuje zejména tyto parametry: rychlost obvodu, šumovou imunitu a spotřebu z napájecího zdroje.
- *U obvodů CMOS je třeba vždy připojit vstupy obvodu na výstup jiného obvodu, napájecí napětí nebo na zem*.

CMOS obvody

- Obvody je třeba budit signály, které mají dostatečně strmé náběžné a sestupné hrany, neboť při pomalejších změnách prudce vzrůstá spotřeba obvodu.
- Klidová spotřeba je velmi nízká (v každé cestě mezi napájecími vstupy hradla je alespoň jeden z tranzistorů uzavřen).
- Spotřeba obvodu je úměrná frekvenci změn vstupních signálů.
- Jestliže obvody nejsou chráněny substrátovými diodami proti přepětí na vstupu je třeba zabránit vzniku a uplatnění statické elektřiny, která může zničit obvod.
- Maximální výstupní větvení je větší než 100 - vstupní proud do tranzistoru FET je zanedbatelný. Naopak velká vstupní kapacita obvodů zhoršuje dynamické vlastnosti.

Logické obvody

8 Output Ports and TTY Interface



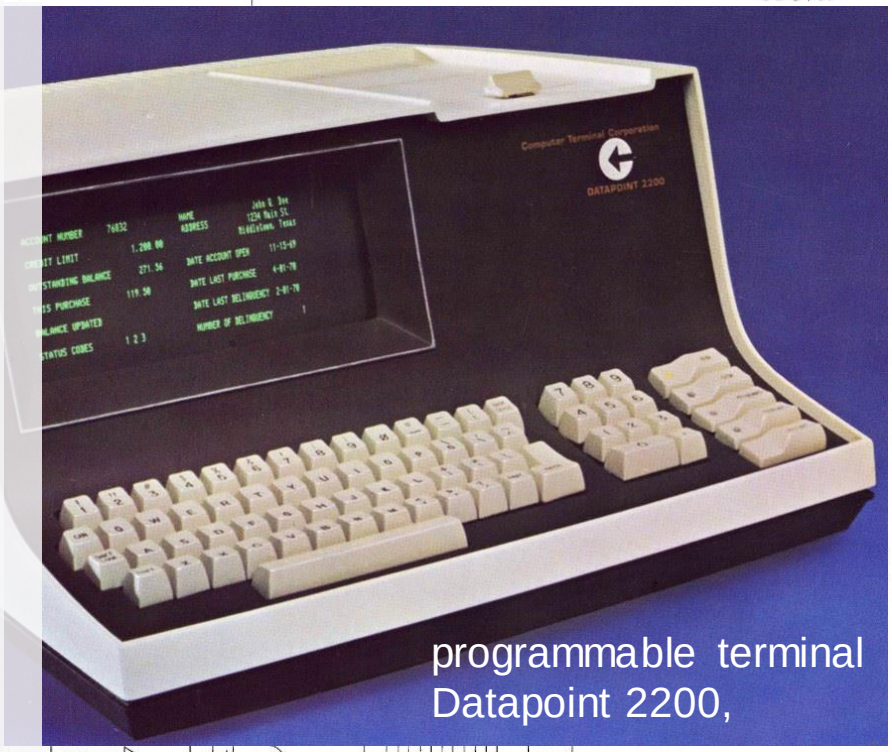
https://microship.com/homebrew-8008-computer-schematics/

Logické obvody

Vývoj počtu tranzistorů

4004 (1971, 14 mm²)	2 300
8080	5 000
8086	2 900
80186	55 000
80286	134 000
80386	275 000
80486	1 180 235
Pentium	3 100 000
PII	7 500 000
PIII	21 000 000
P4	112 000 000
Core2	291 000 000
I7	731 000 000
Itanium2 (2006, 596 mm²)	1 700 000 000
Core i7-5960X	2 600 000 000
AMD Ryzen 7 (213 mm²)	4 800 000 000
SPARC	10 000 000 000
Stratix10	30 000 000 000
AMD Zen2 (2019, 7nm, 205+ mm²)	39 540 000 000
Apple M2 Max (2022, 5 nm…)	67,000,000,000

to 300 million per square millimeter

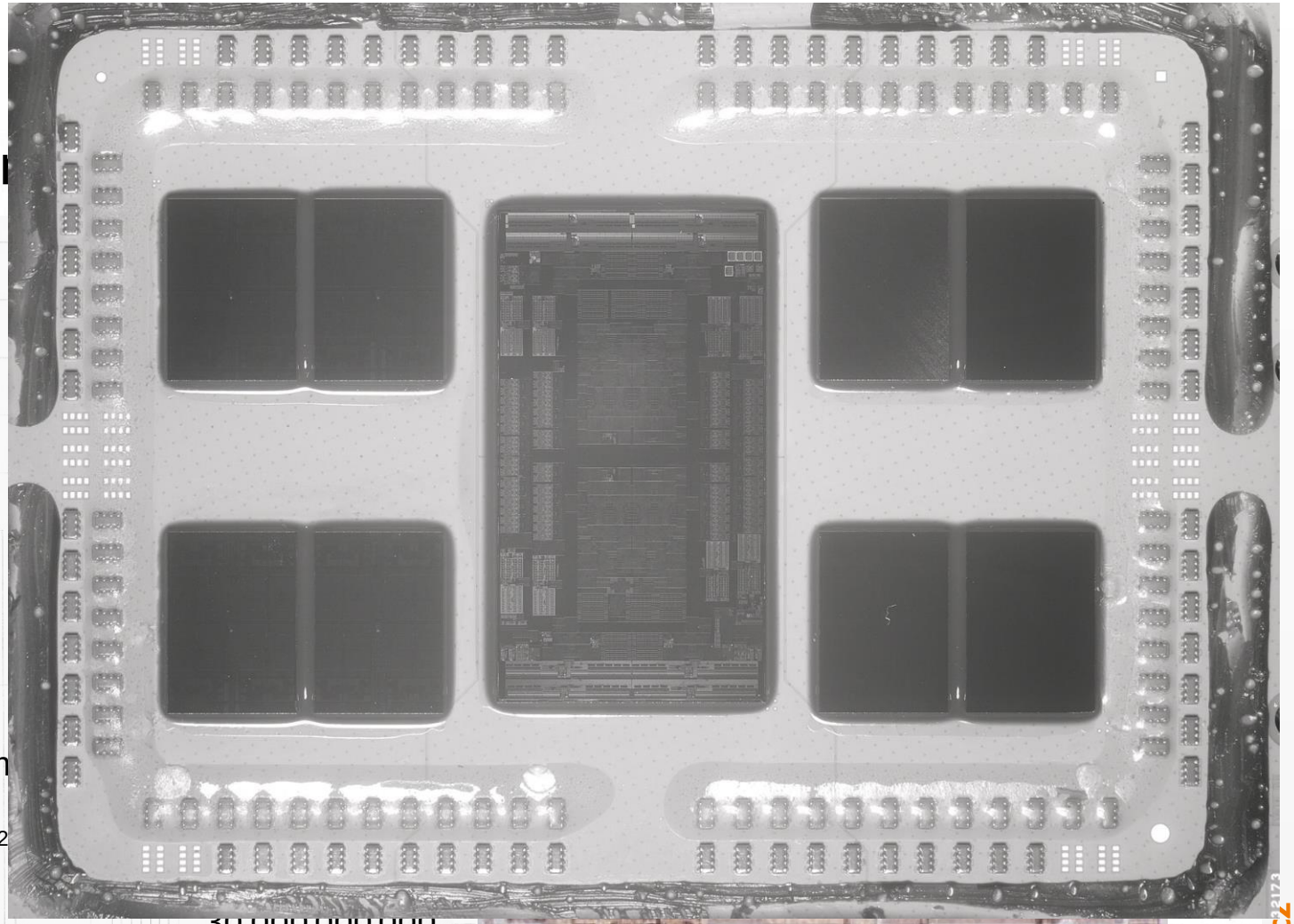


programmable terminal Datapoint 2200,

I-8008
2 kB memory (8 bit)
Displej 12 x 80 znaků

https://microship.com/homebrew-8008-computer-schematics/, https://en.wikipedia.org/wiki/Transistor_count

Vývoj počtu tr



4004 (1971, 14 mm²)
8080
8086
80186
80286
80386
80486
Pentium
PII
PIII
P4
Core2
I7
Itanium2 (2006, 596 mm
Core i7-5960X
AMD Ryzen 7 (213 mm²
SPARC
Stratix10

AMD Zen2 (2019, 7nm, 205+ mm²) 39 540 000 000

2 kB memory (8 bit)
Displej 12 x 80 znaků

https://microship.com/homebrew-8008-computer-schematics/

Pneumatické logické prvky

Výhody:

- mohou pracovat v prostředí s nebezpečím výbuchu (nejiskří)
- mohou pracovat i ve vlhkém a prašném prostředí
- pohony se vyznačují velkými přestavnými silami a délkami a jsou rychlejší než elektrické
- snesou velká i nárazová zatížení
- schopnost práce ve vyšších teplotách

Nevýhody:

- větší energetické ztráty (energie stlačeného vzduchu je 5 až 7x dražší než elektrická)
- větší pořizovací a provozní náklady tlakovzdušných stanic
- velké zpoždění signálu

Pneumatické logické prvky

Stavební prvky:

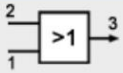
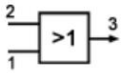
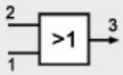
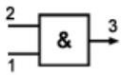
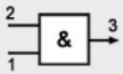
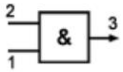
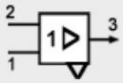
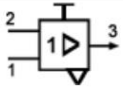
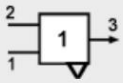
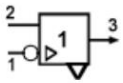
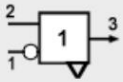
- základní logické prvky, tj. negace, konjunkce (součin), disjunkce (součet) a další
- sekvenční logické funkce, např. paměti, klopné obvody, zpožďovací funkce
- reléové funkce (spínací a rozpínací relé)
- časové funkce (časové relé, generátory impulsů apod.)

Pneumatické logické prvky

Analogie s elektronickými obvody

- *napětí* - tlak vzduchu
- *proud* - průtok vzduchu
- *rezistor* - kapiláry, clony, trysky
- *proměnné rezistory* - kuželové ventilky, převodník klapka - tryska
- *kondenzátor* - slepé komory
- *ladící kondenzátor* - slepá komora s proměn. odporem nebo objemem
- *operační zesilovač* - převodník (rozvod) klapka-tryska

Pneumatické logické prvky

		**81 521 501**	OR
		**81 540 001**	OR
		**81 540 005**	OR
		**81 522 501**	AND
		**81 541 001**	AND
		**81 541 005**	AND
		**81 501 025**	YES
		**81 501 065**	YES
		**81 503 025**	YES
		**81 504 025**	NOT
		**81 506 025**	NOT

https://www.sentronic.com/frontend/scripts/index.php?setMainAreaTemplatePath=mainarea_productlist.html&groupId=440&groupNavRiderSel=1&setLanguageId=2

http://cz.rs-online.com/web/c/pneumatika-hydraulika-a-prenos-vykonu/pneumaticka-pocitadla-logicke-regulatory-a-casovace/pneumaticke-logicke-regulatory/

https://youtu.be/betlNcZ27CM
https://www.youtube.com/watch?v=otR1b3ACqrA

Pneumatické logické prvky

http://hydraulicspneumatics.com/
https://youtu.be/hxNp6qYKj_g

Kombinační obvody, optimalizace zapojení

Metody zápisu logických funkci

m = počet logických funkcí; *n* = počet vstupních proměnných: $m = 2^{2^n}$

1. PRAVDIVOSTNÍ TABULKA

Tabulka má **2^n řádků**

Případný symbol **X** vyznačuje hodnotu nedefinovanou (na které nezáleží, zda je 0 n. 1)

- pro vstupní proměnné (umožňuje úspornější zápis sloučením řádků)
- pro funkční hodnotu

2. VÝČET JEDNIČKOVÝCH STAVŮ

Stavové indexy z pravd. tabulky, kde je funkční hodnota 1

př.: f = Σ(1,2,4,6)

Metody zápisu logických funkci

3. ÚPLNÝ ZÁPIS LOGICKÉ FUNKCE

- pomocí elementárních logických funkcí (úprava do minimalizovaného tvaru)
- algebraické vyjádření (součet součinů (mintermů) vs. součin součtů)

př. XOR: $Y = \overline{A} \cdot B + A \cdot \overline{B}$, $Y = (A+B) \cdot (\overline{A}+\overline{B})$

4. VÉNŮV DIAGRAM

5. MAPY

6. HDL POPIS

Pravdivostní tabulka

stavový index s	c b a	funkční hodnota $f(c,b,a)$	minterm P_s
	MSB	LSB	
0	0 0 0	0	$\bar{c}\bar{b}\bar{a}$
1	0 0 1	1	$\bar{c}\bar{b}a$
2	0 1 0	1	$\bar{c}b\bar{a}$
3	0 1 1	0	$\bar{c}ba$
4	1 0 0	1	$c\bar{b}\bar{a}$
5	1 0 1	0	$c\bar{b}a$
6	1 1 0	1	$cb\bar{a}$
7	1 1 1	0	cba

Pravdivostní tabulka

stavový index s	c b a	funkční hodnota $f(c,b,a)$	minterm P_s
0	0 0 0	0	$\bar{c}\bar{b}\bar{a}$
1	0 0 1	1	$\bar{c}\bar{b}a$
2	0 1 0	1	$\bar{c}b\bar{a}$
3	0 1 1	0	$\bar{c}ba$
4	1 0 0	1	$c\bar{b}\bar{a}$
5	1 0 1	0	$c\bar{b}a$
6	1 1 0	X	$cb\bar{a}$
7	1 1 1	X	cba

Nedefinované hodnoty

Pravdivostní tabulka

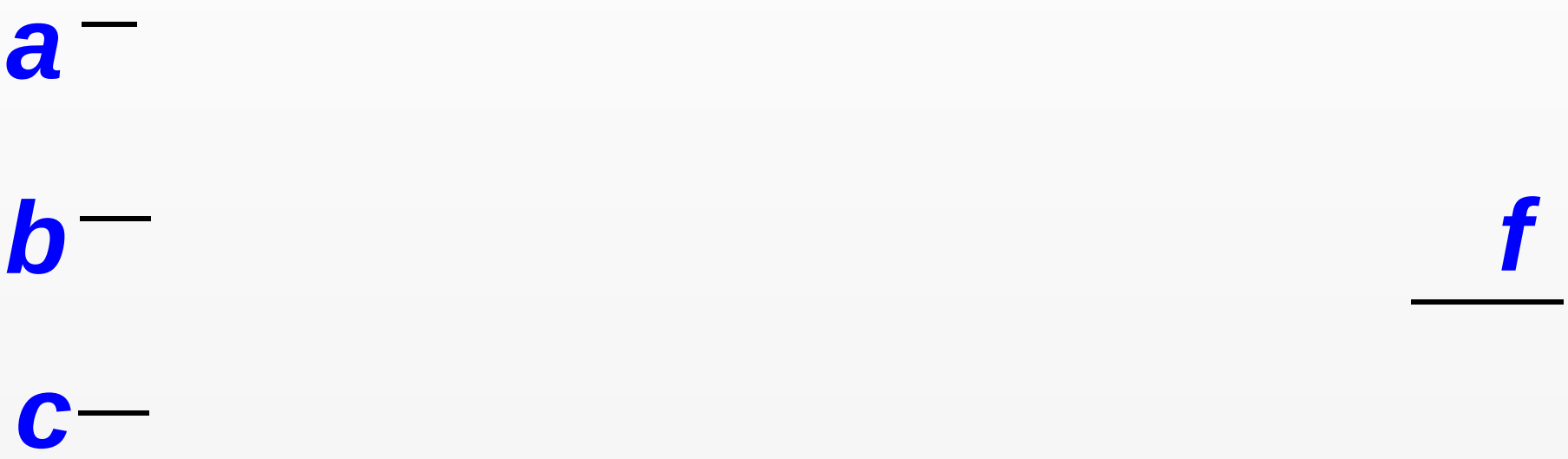
stavový index s	c b a	funkční hodnota $f(c,b,a)$	minterm P_s
0	0 0 0	0	$\bar{c}\bar{b}\bar{a}$
1	0 0 1	1	$\bar{c}\bar{b}a$
2	0 1 0	1	$\bar{c}b\bar{a}$
3	0 1 1	0	$\bar{c}ba$
4	1 0 0	1	$c\bar{b}\bar{a}$
5	1 0 1	0	$c\bar{b}a$
6	1 1 0	1	$cb\bar{a}$
7	1 1 1	0	cba

Sloupec **minterm** ukazuje, jak je možno jednotlivým řádkům tabulky přiřadit logický výraz.

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + cb\bar{a} + cb\bar{a}$$

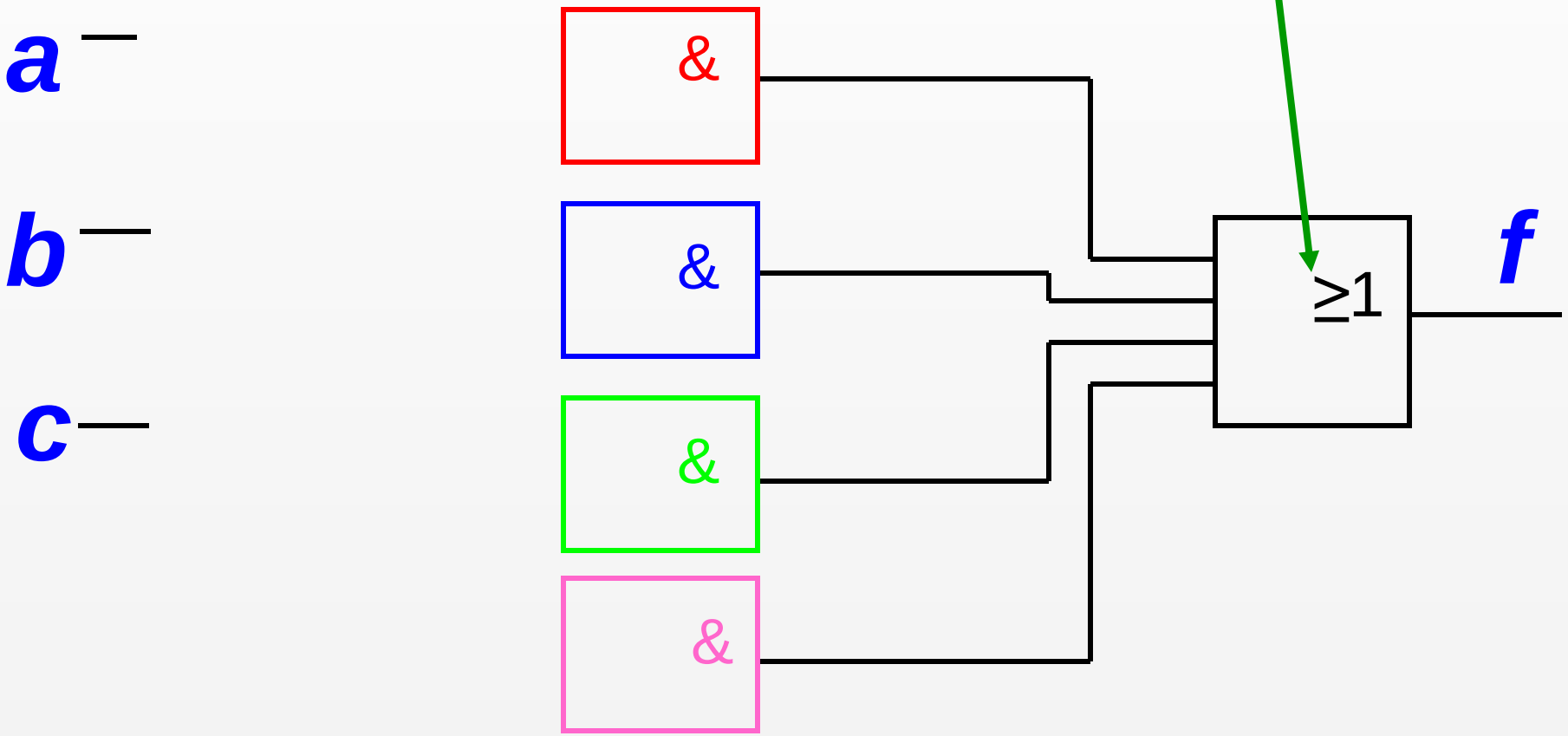
Realizace logické funkce

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + c\bar{b}\bar{a} + cb\bar{a}$$



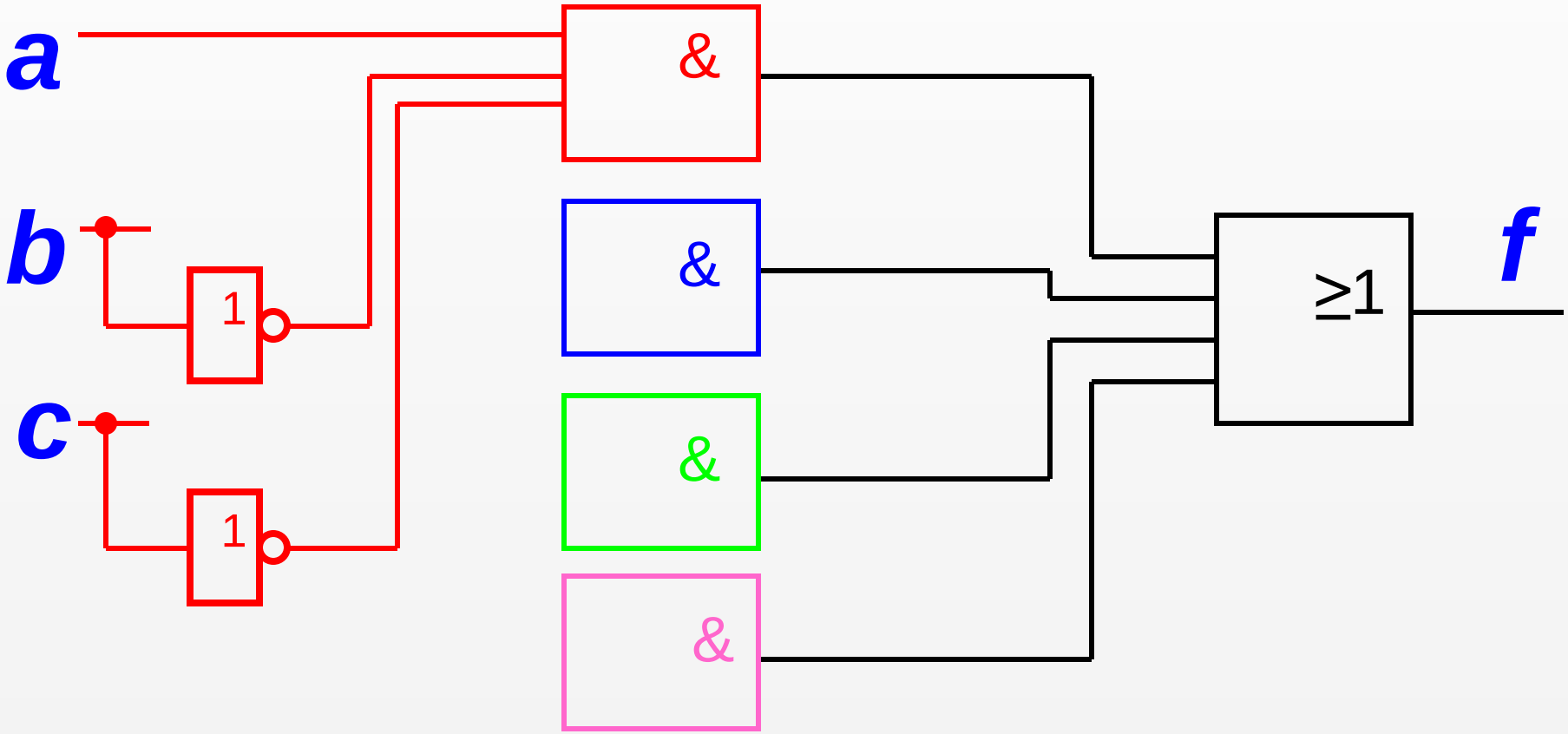
Realizace logické funkce

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + c\bar{b}\bar{a} + cb\bar{a}$$



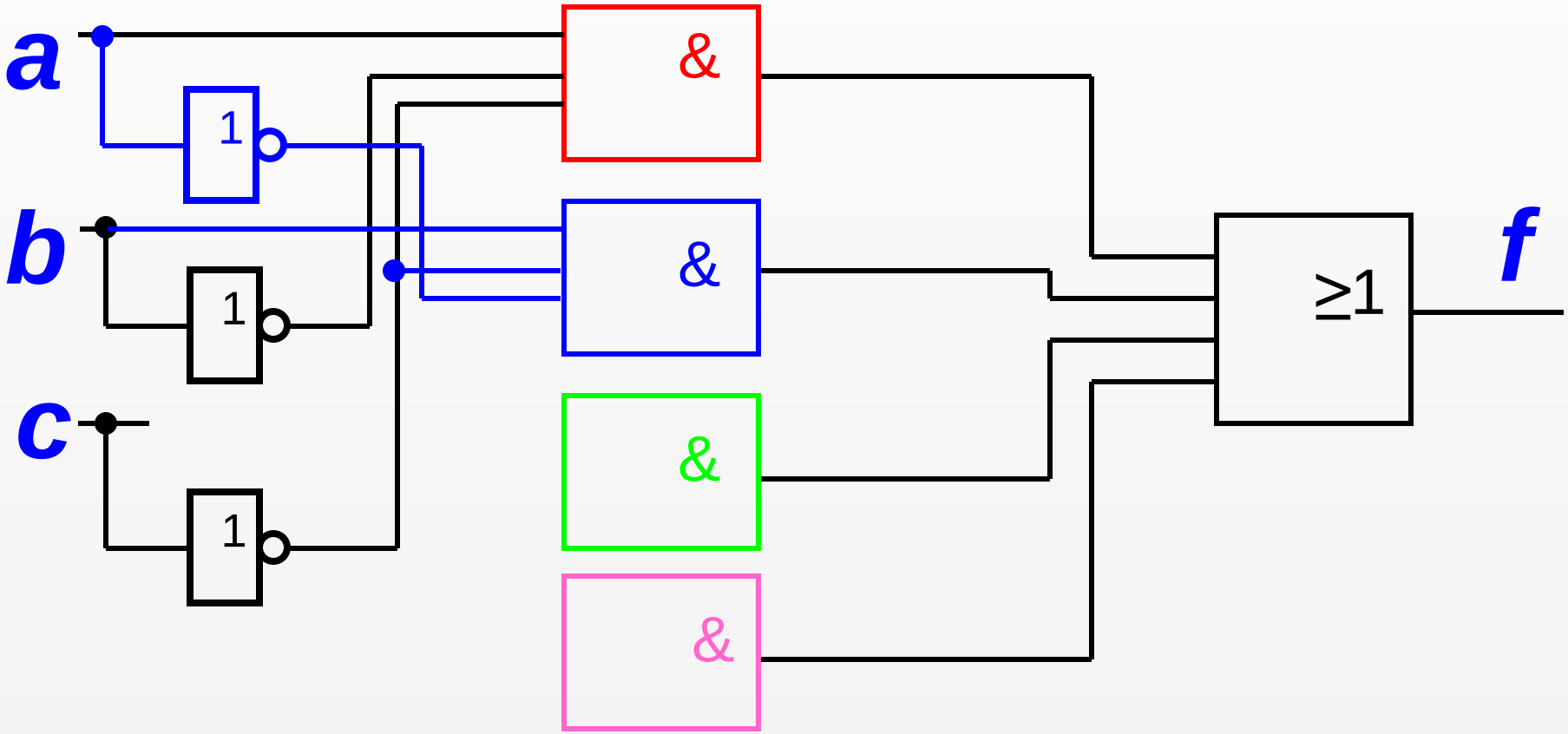
Realizace logické funkce

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + cb\bar{a} + cb\bar{a}$$



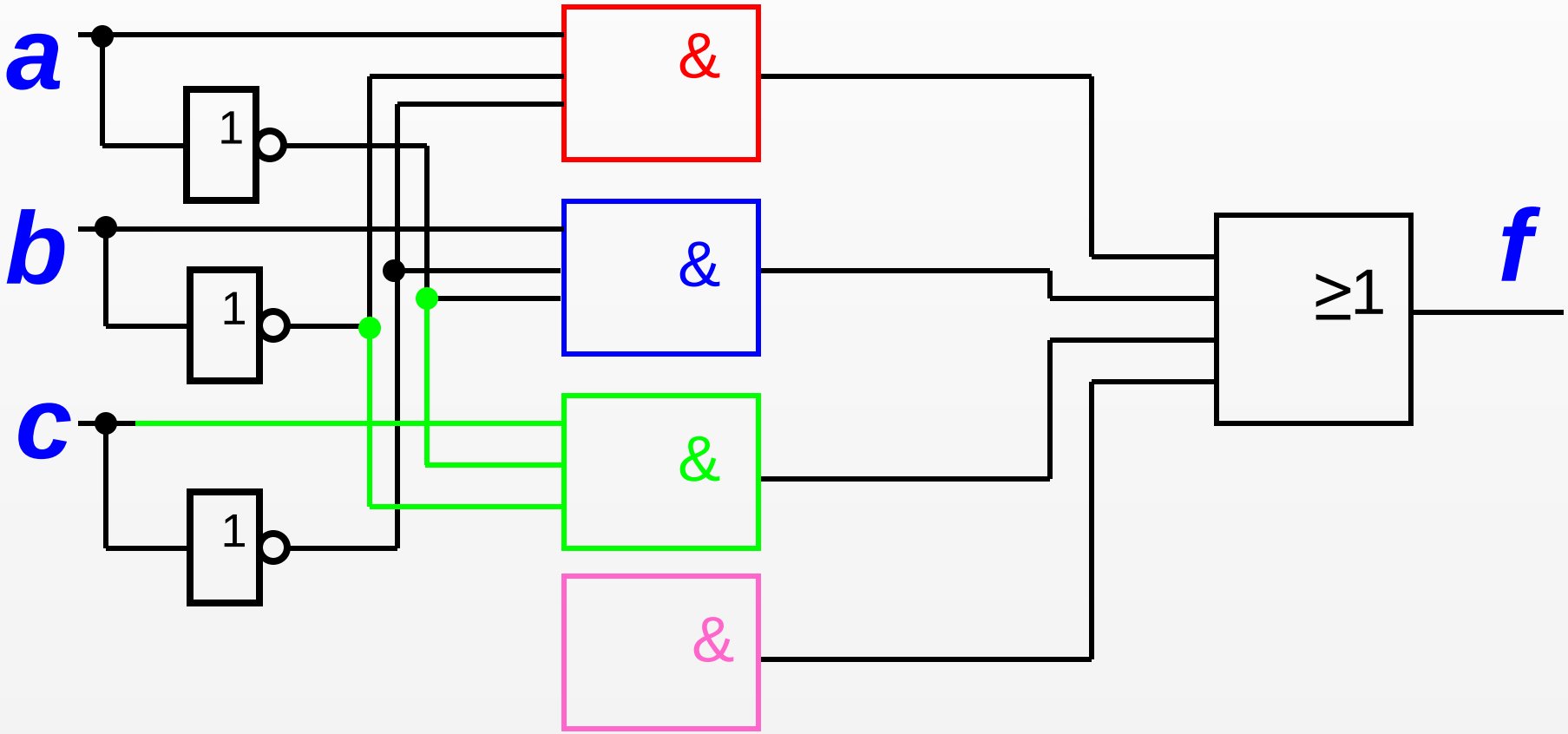
Realizace logické funkce

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + cb\bar{a} + cb\bar{a}$$



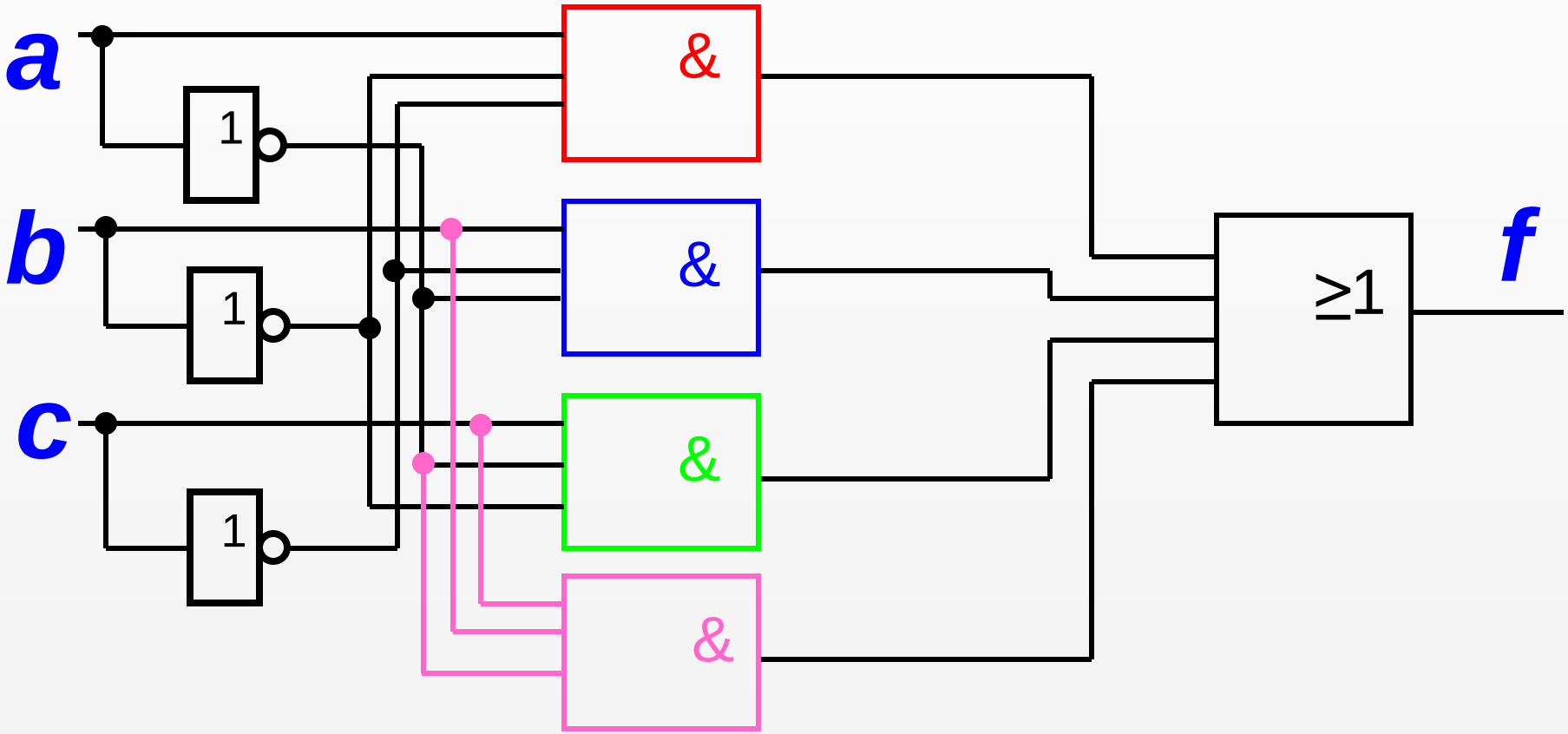
Realizace logické funkce

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + c\bar{b}\bar{a} + cb\bar{a}$$



Realizace logické funkce

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + c\bar{b}\bar{a} + cb\bar{a}$$



Booleova algebra

Dvojí negace:

$$\overline{\overline{a}} = a$$

De Morganovo pravidlo:

$$\overline{(a + b)} = \overline{a} \cdot \overline{b} \qquad \overline{(a \cdot b)} = \overline{a} + \overline{b}$$

Realizace logické funkce

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

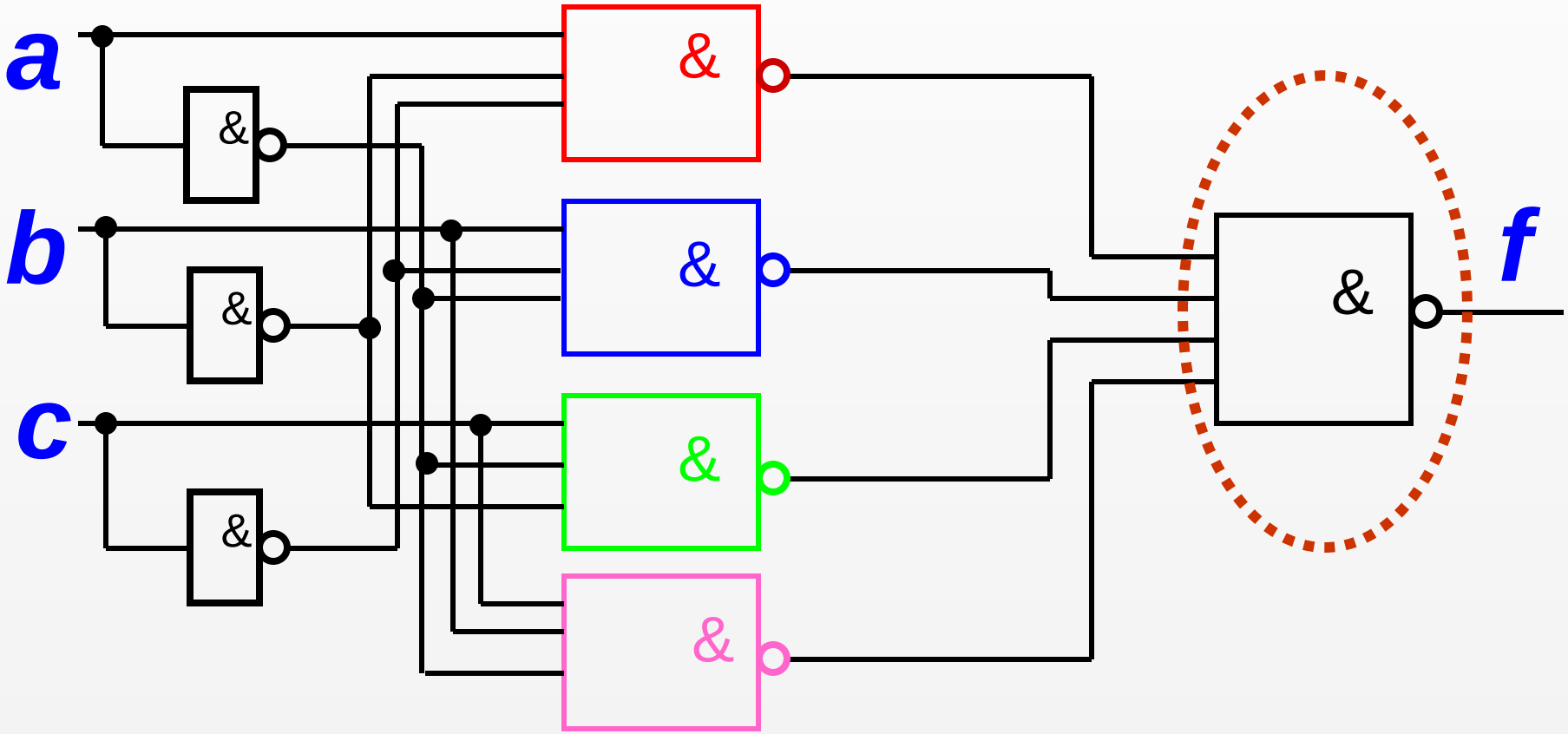
(de Morganovo pravidlo)

$$f = \overline{c}\,\overline{b}\,a + \overline{c}\,b\,\overline{a} + c\,\overline{b}\,\overline{a} + c\,b\,\overline{a}$$

$$f = \overline{\overline{\overline{c}\,\overline{b}\,a} * \overline{\overline{c}\,b\,\overline{a}} * \overline{c\,\overline{b}\,\overline{a}} * \overline{c\,b\,\overline{a}}}$$

Realizace logické funkce

$$f = \overline{\bar{c}\bar{b}a} * \overline{\bar{c}b\bar{a}} * \overline{c\bar{b}\bar{a}} * \overline{cb\bar{a}}$$



Realizace logické funkce

- http://tma.main.jp/logic/index_en.html
- https://ddd.fit.cvut.cz/prj/BOOM/
- A další „truth-table solver/minimiser“

Minimalizace výrazů

Redukce počtu termů, jejich vstupních proměnných a výsledných negací.

Při praktickém návrhu jsou různá kritéria a omezení :

- použití omezeného typu hradel (AND, OR, ...)
- max. počet vstupů hradla (4, 8)
- minimalizace počtu hradel nebo pouzder
- max. výstupní větvení (10 u TTL)
- počet hradel v nejdelší cestě (doba šíření signálu), omezení logických hazardů

- ❑ Algebraickou úpravou logického výrazu
- ❑ Algoritmy úprav vhodné pro počítačový návrh obvodů
- ❑ Ručně pro 3 ev. 4 proměnné : Karnaughova mapa

Karnaughova mapa

st. index	d	c	b	a	*f*
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Karnaughova mapa

0	1
1	1
2	0
3	0
4	0
5	1
6	0
7	0
8	1
9	1
10	1
11	0
12	1
13	1
14	1
15	0

d c \ b a	0 0	0 1	1 1	1 0
0 0	0	1	3	2
0 1	4	5	7	6
1 1	12	13	15	14
1 0	8	9	11	10

Karnaughova mapa

0	1
1	1
2	0
3	0
4	0
5	1
6	0
7	0
8	1
9	1
10	1
11	0
12	1
13	1
14	1
15	0

d c \ b a	0 0	0 1	1 1	1 0
0 0	1	1	0	0
0 1	0	1	0	0
1 1	1	1	0	1
1 0	1	1	0	1

$$f = \overline{a}.\overline{d} + \overline{a}.\overline{b} + \overline{b}.\overline{c}$$

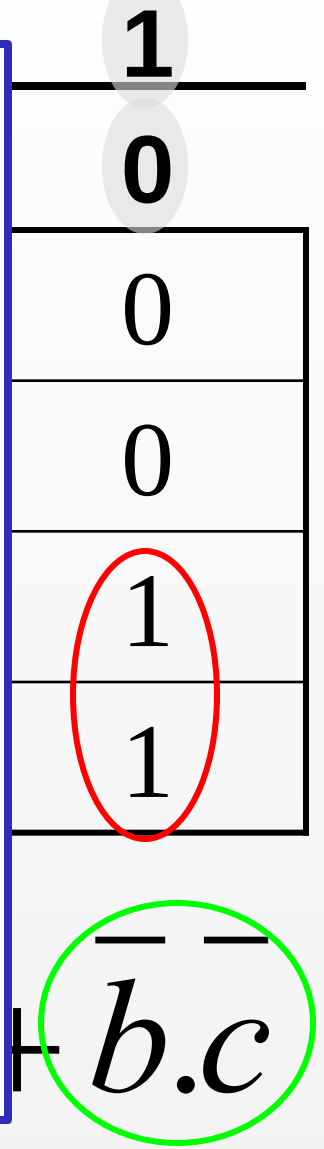
Karnaughova mapa

0	1
1	1
2	0
3	0
4	0
5	1
6	0
7	0
8	1
9	1
10	1
11	0
12	1
13	1
14	1
15	0

Truth table (4input)

a	b	c	d	Output
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

go



Truth table (4input)

a	b	c	d	Output
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
1	1	1	1	0

go

0	1
1	1
2	0
3	0
11	0
12	1
13	1
14	1
15	0

DNF (with ~) = ~a~b~c~d + ~a~b~cd + ~ab~cd + a~b~c~d + a~b~cd + a~bc~d + ab~c~d + ab~cd + abc~d

DNF (with overline) = $\bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}b\bar{c}d + a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d}$

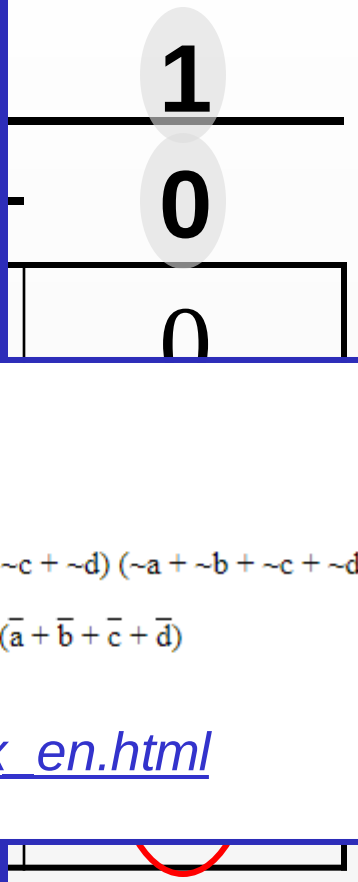
CNF (with ~) = (a + b + ~c + d) (a + b + ~c + ~d) (a + ~b + c + d) (a + ~b + ~c + d) (a + ~b + ~c + ~d) (~a + b + ~c + ~d) (~a + ~b + ~c + ~d)

CNF (with overline) = $(a + b + \bar{c} + d)(a + b + \bar{c} + \bar{d})(a + \bar{b} + c + d)(a + \bar{b} + \bar{c} + d)(a + \bar{b} + \bar{c} + \bar{d})(\bar{a} + b + \bar{c} + \bar{d})(\bar{a} + \bar{b} + \bar{c} + \bar{d})$

Minimal Form (with ~) = ~b~c + ~cd + a~d

Minimal Form (with overline) = $\bar{b}\bar{c} + \bar{c}d + a\bar{d}$

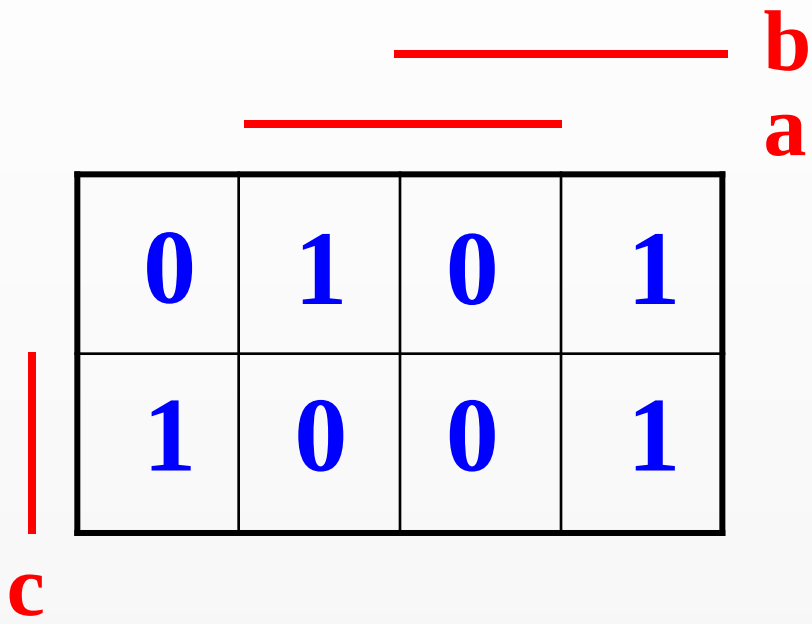
http://tma.main.jp/logic/index_en.html



$$f = \bar{a}.\bar{d} + a.\bar{b} + \bar{b}.\bar{c}$$

Realizace logické funkce

stavový index s	c b a	funkční hodnota $f(c,b,a)$	minterm P_s
0	0 0 0	0	$\bar{c}\bar{b}\bar{a}$
1	0 0 1	1	$\bar{c}\bar{b}a$
2	0 1 0	1	$\bar{c}b\bar{a}$
3	0 1 1	0	$\bar{c}ba$
4	1 0 0	1	$c\bar{b}\bar{a}$
5	1 0 1	0	$c\bar{b}a$
6	1 1 0	1	$cb\bar{a}$
7	1 1 1	0	cba

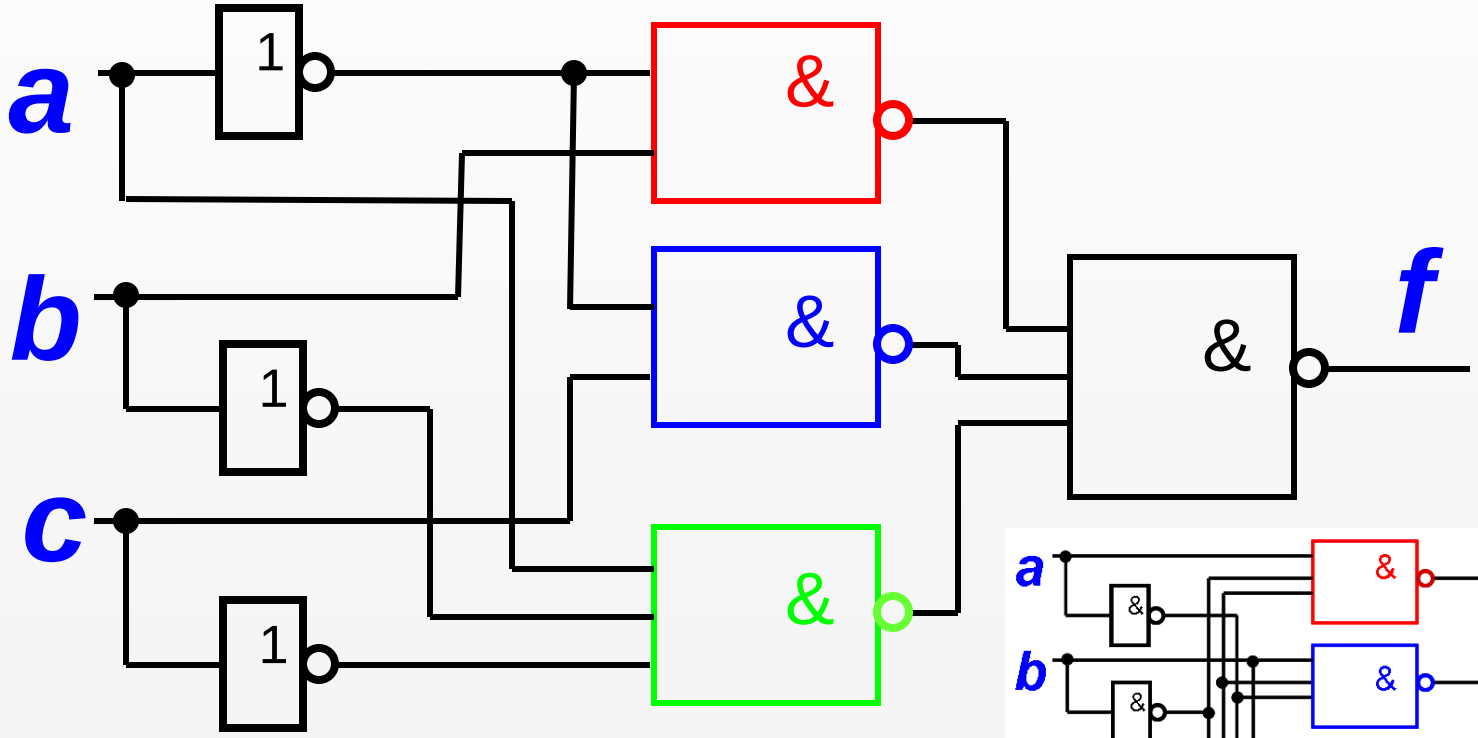


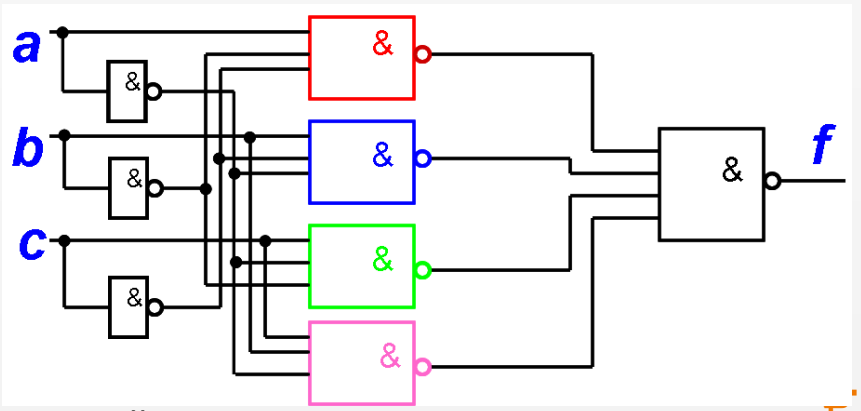
$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + c\bar{b}\bar{a} + cb\bar{a}$$

$$f = \bar{a}b + \bar{a}c + a\bar{b}\bar{c}$$

Realizace logické funkce

$$f = \bar{a}b + \bar{a}c + a\bar{b}\bar{c}$$





Zákony Booleovy algebry

komutativita:

$a + b = b + a$ $\qquad$ $a.b = b.a$

asociativita:

$a + (b + c) = (a + b) + c$ $\qquad$ $a.(b.c) = (a.b).c$

distributivita:

$a + (b.c) = (a + b).(a + c)$ $\qquad$ $a.(b + c) = (a.b) + (a.c)$

neutralita 0 a 1

$a + 0 = a$ $\qquad$ $a.1 = a$

Booleova algebra

vlastnosti komplementu:

$a + \bar{a} = 1$ $\qquad$ $a.\bar{a} = 0$

agresivita 0 a 1:

$a + 1 = 1$ $\qquad$ $a.0 = 0$

idempotence:

$a + a = a$ $\qquad$ $a.a = a$

Absorbce

$a + a.b = a$ $\qquad$ $a.(a + b) = a$

Booleova algebra

Dvojí negace:

$$\overline{\overline{a}} = a$$

Absorbce negace

$$a + \overline{a} \cdot b = a + b$$

$$a \cdot (\overline{a} + b) = a \cdot b$$

Booleova algebra

Consensus:

$$a \cdot b + \overline{a} \cdot c + b \cdot c = a \cdot b + \overline{a} \cdot c$$

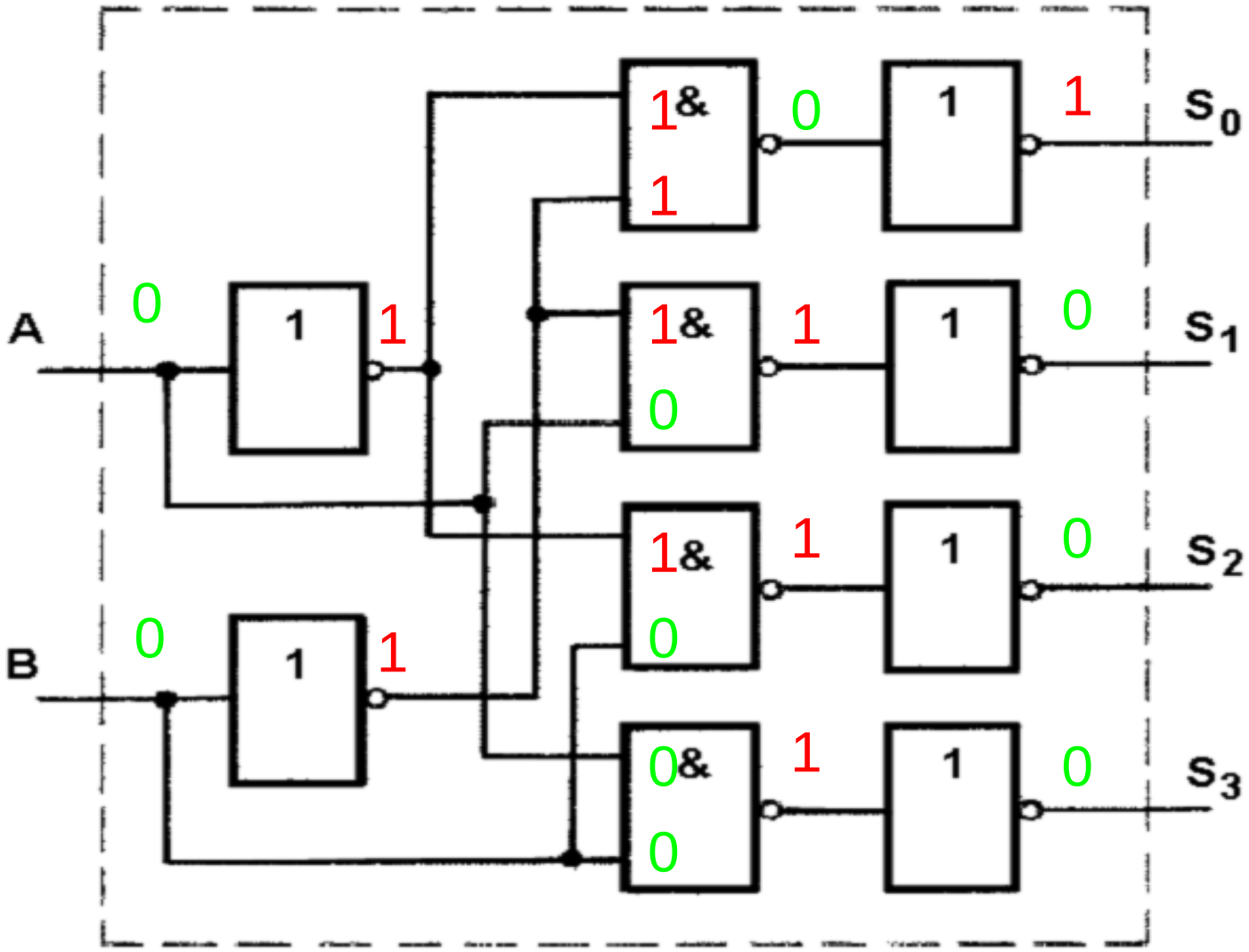
$$(a \cdot b + b \cdot c + c \cdot \overline{a} = a \cdot b + c \cdot \overline{a})$$

$$(a + b).(\overline{a} + c).(b + c) = (a + b).(\overline{a} + c)$$

Kombinační obvody

- **Dekodér** – změna kódu, šifrování, kombinace M : N
- **Multiplexer** – číslicový přepínač více vstupů

Dekodér



$$S_0 = \overline{B}.\overline{A}$$

$$S_1 = \overline{B}.A$$

$$S_2 = B.\overline{A}$$

$$S_3 = B.A$$

Dekodér

Vzpomínáte na P.T.?

stavový index s	c b a	funkční hodnota $f(c,b,a)$	minterm P_s
0	0 0 0	0	$\bar{b}\bar{a}$
1	0 0 1	1	$\bar{b}a$
2	0 1 0	1	$b\bar{a}$
3	0 1 1	0	ba
4	1 0 0	1	$c\bar{b}\bar{a}$
5	1 0 1	0	$c\bar{b}a$
6	1 1 0	1	$cb\bar{a}$
7	1 1 1	0	cba

$$S_0 = \overline{B}.\overline{A}$$

$$S_1 = \overline{B}.A$$

$$S_2 = B.\overline{A}$$

$$S_3 = B.A$$

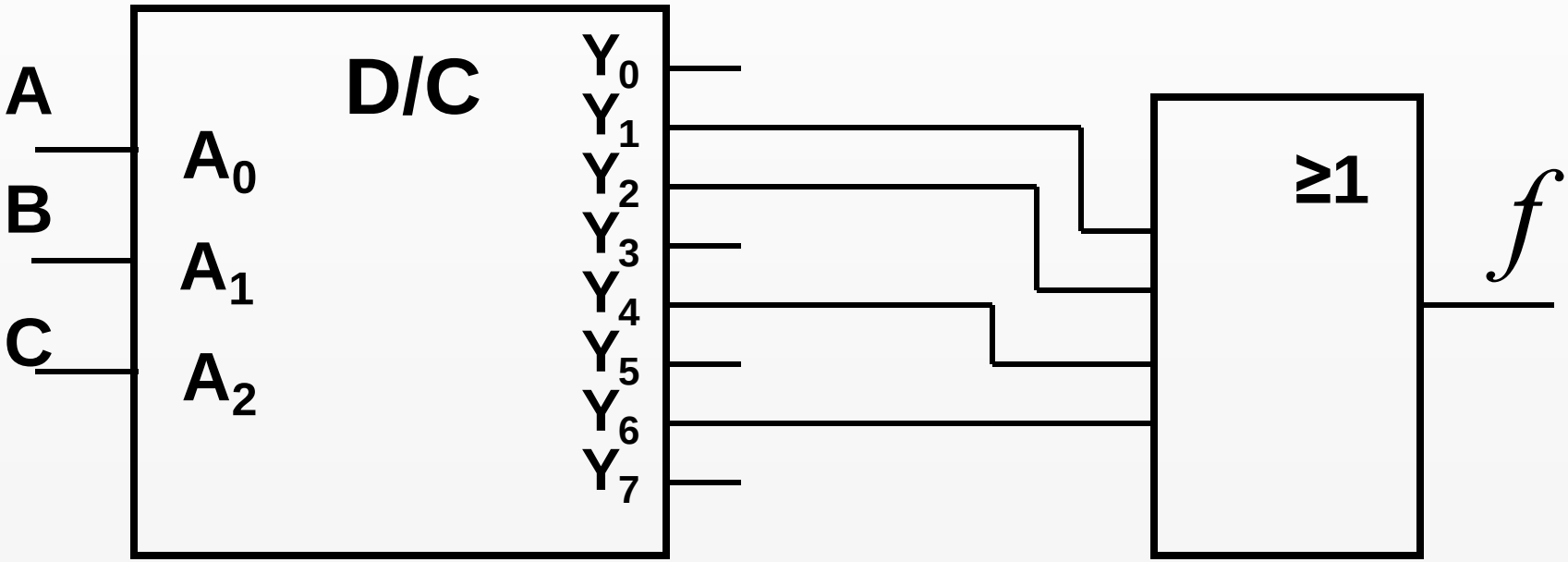
Dekodér

stavový index s	c b a	funkční hodnota $f(c,b,a)$	minterm P_s
0	0 0 0	0	$\bar{c}\bar{b}\bar{a}$
1	0 0 1	1	$\bar{c}\bar{b}a$
2	0 1 0	1	$\bar{c}b\bar{a}$
3	0 1 1	0	$\bar{c}ba$
4	1 0 0	1	$c\bar{b}\bar{a}$
5	1 0 1	0	$c\bar{b}a$
6	1 1 0	1	$cb\bar{a}$
7	1 1 1	0	cba

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + c\bar{b}\bar{a} + cb\bar{a}$$

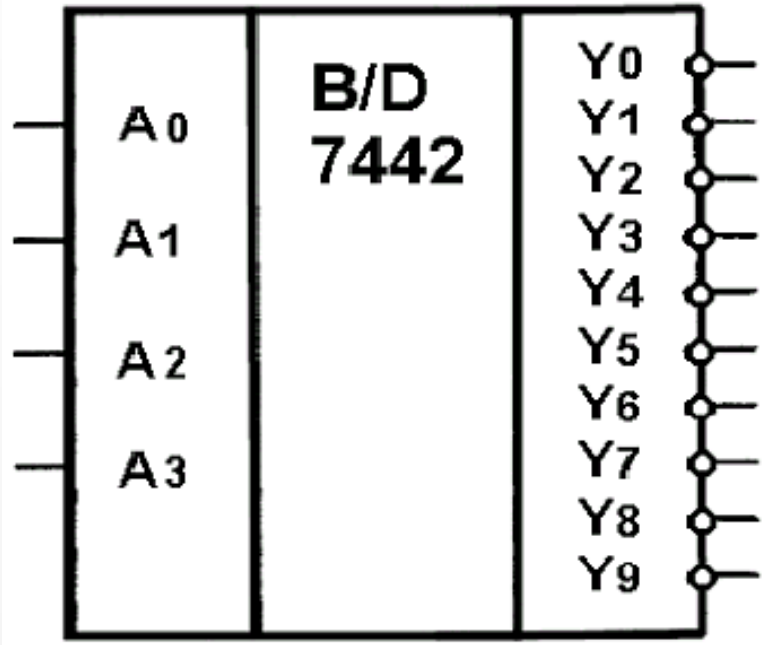
Dekodér

$$f = \bar{c}\bar{b}a + \bar{c}b\bar{a} + c\bar{b}\bar{a} + cb\bar{a}$$

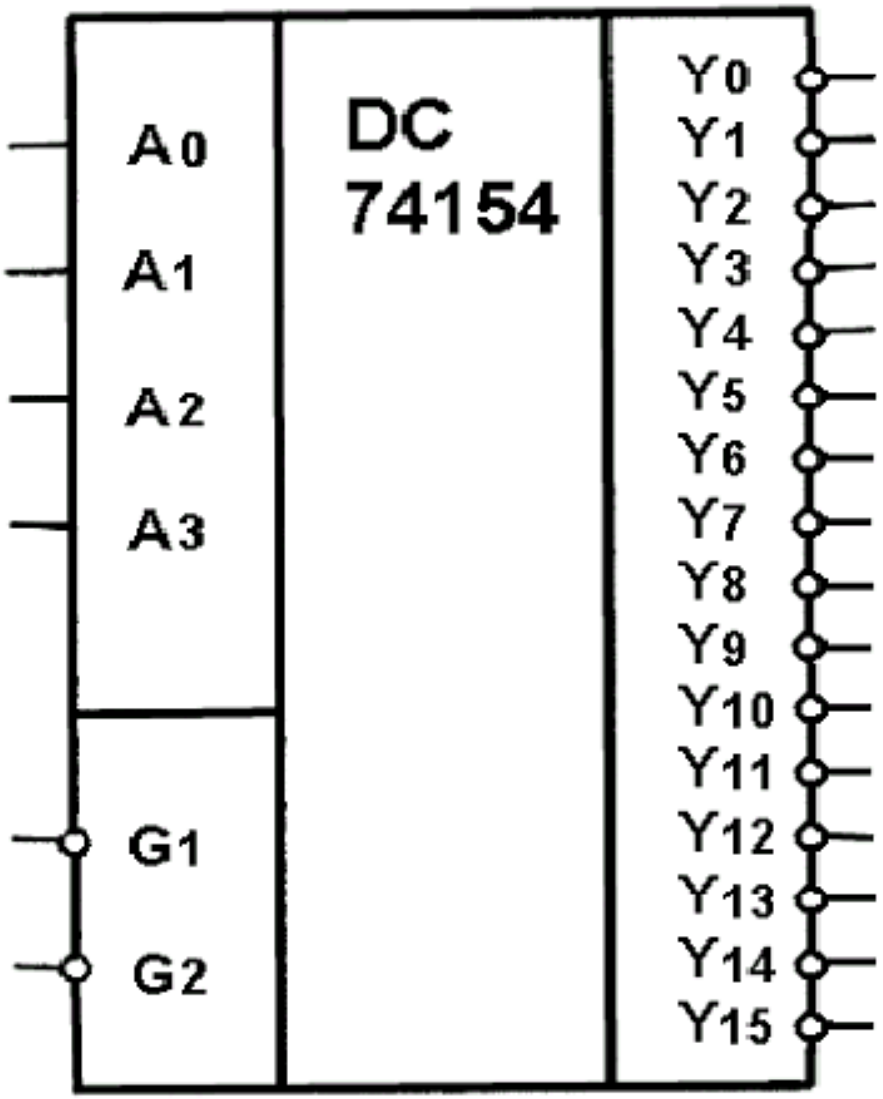


Dekodéry v IO

1 z 8
1 z 10
1 z 16

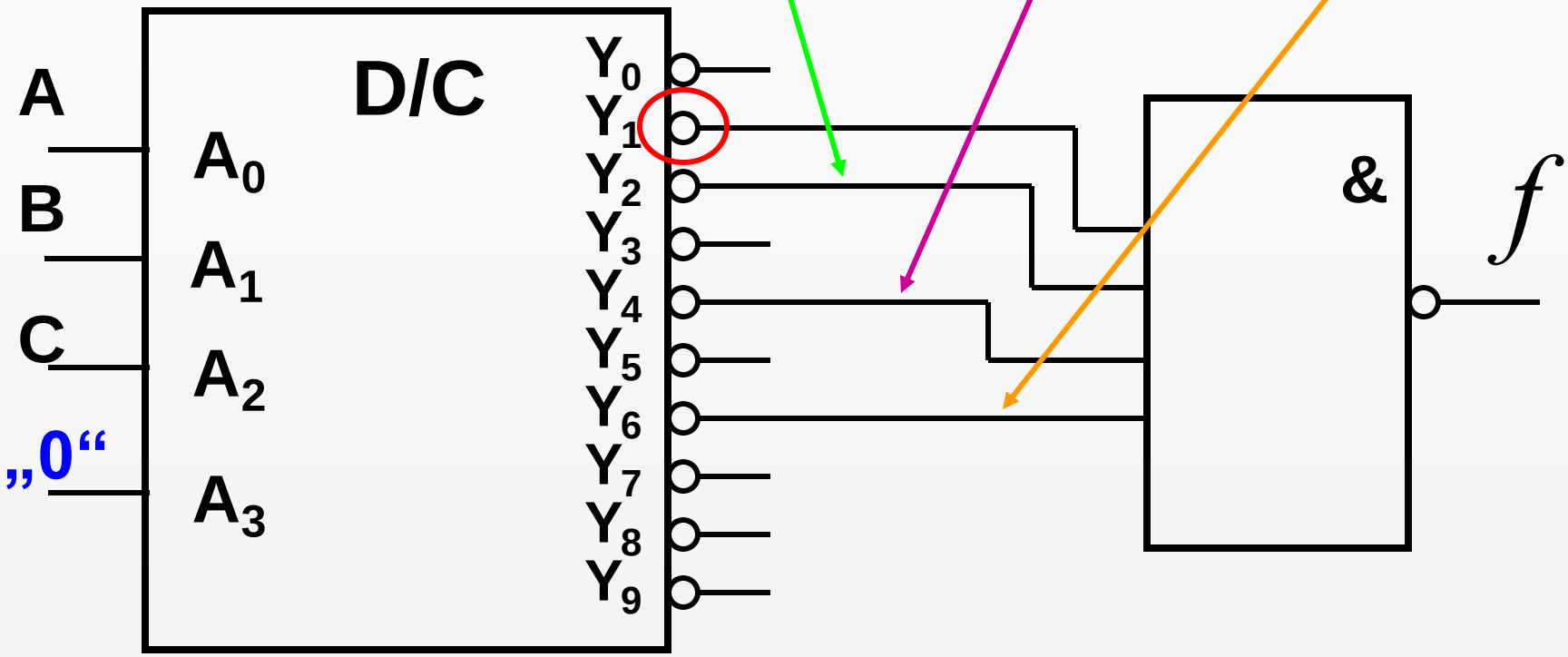


BCD (hexa) → 7-segmentový kód

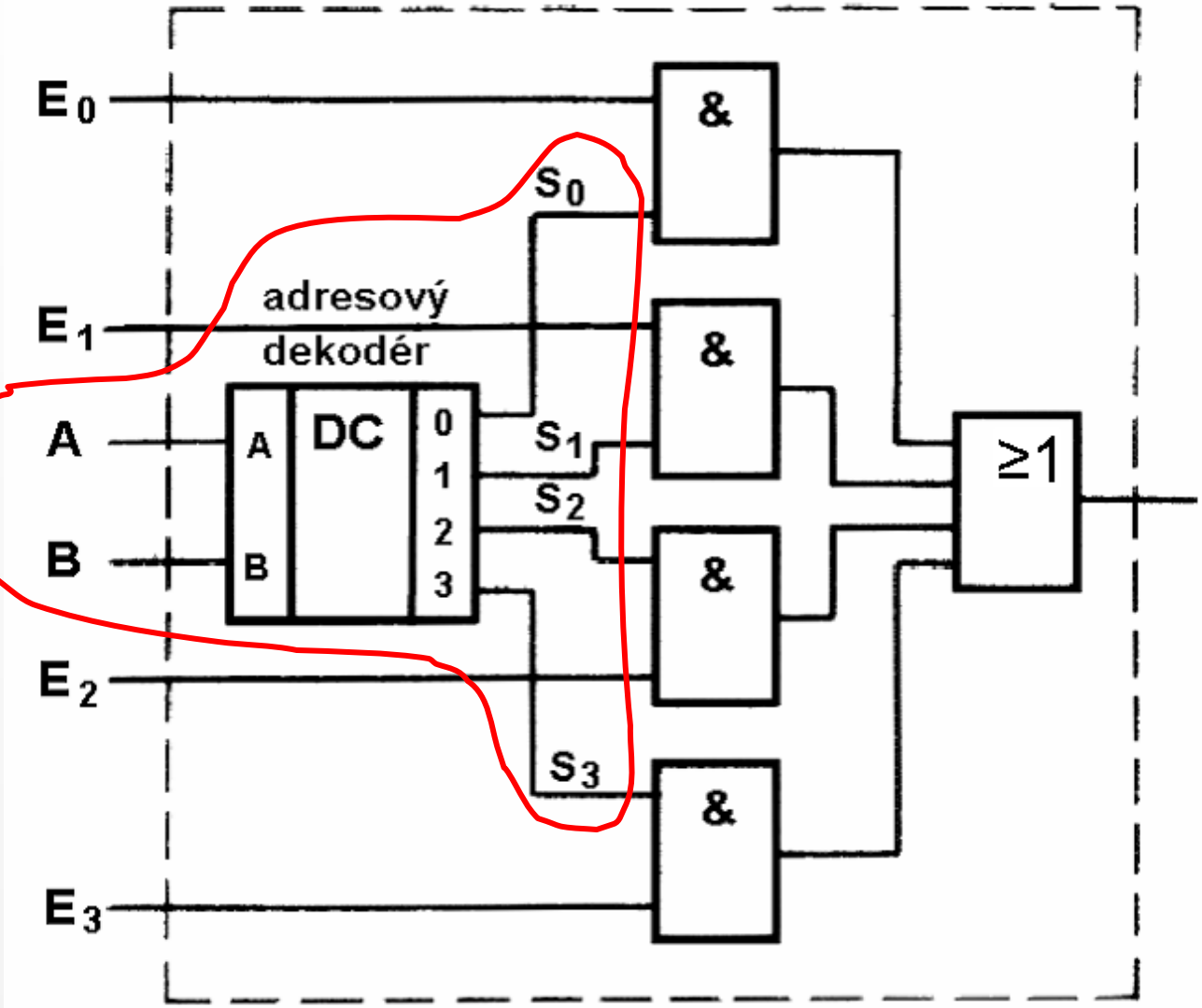


Dekodér

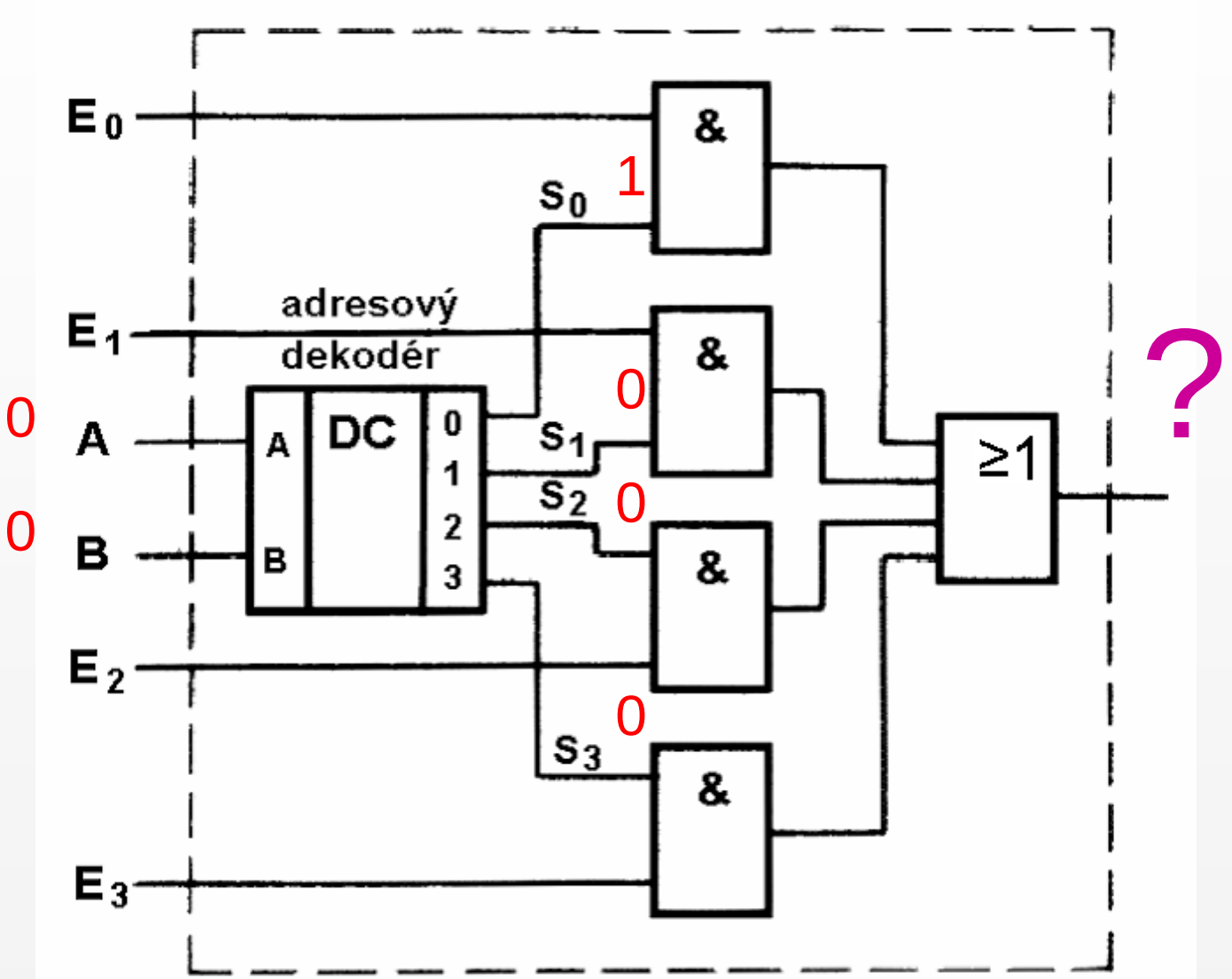
$$f = \overline{\overline{c}\,\overline{b}\,a} * \overline{\overline{c}\,b\,\overline{a}} * \overline{c\,\overline{b}\,\overline{a}} * \overline{c\,b\,\overline{a}}$$



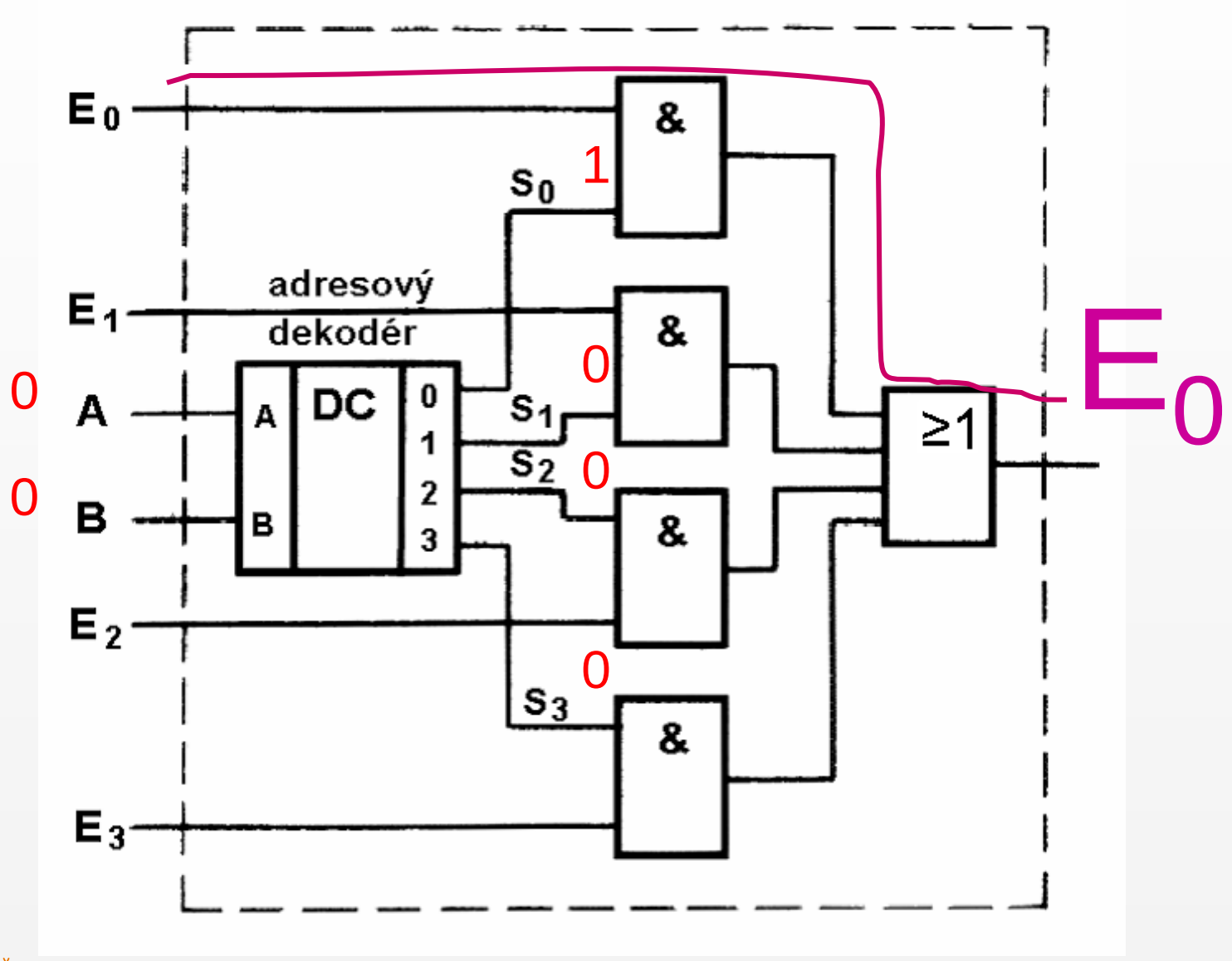
Multiplexer (digitální)



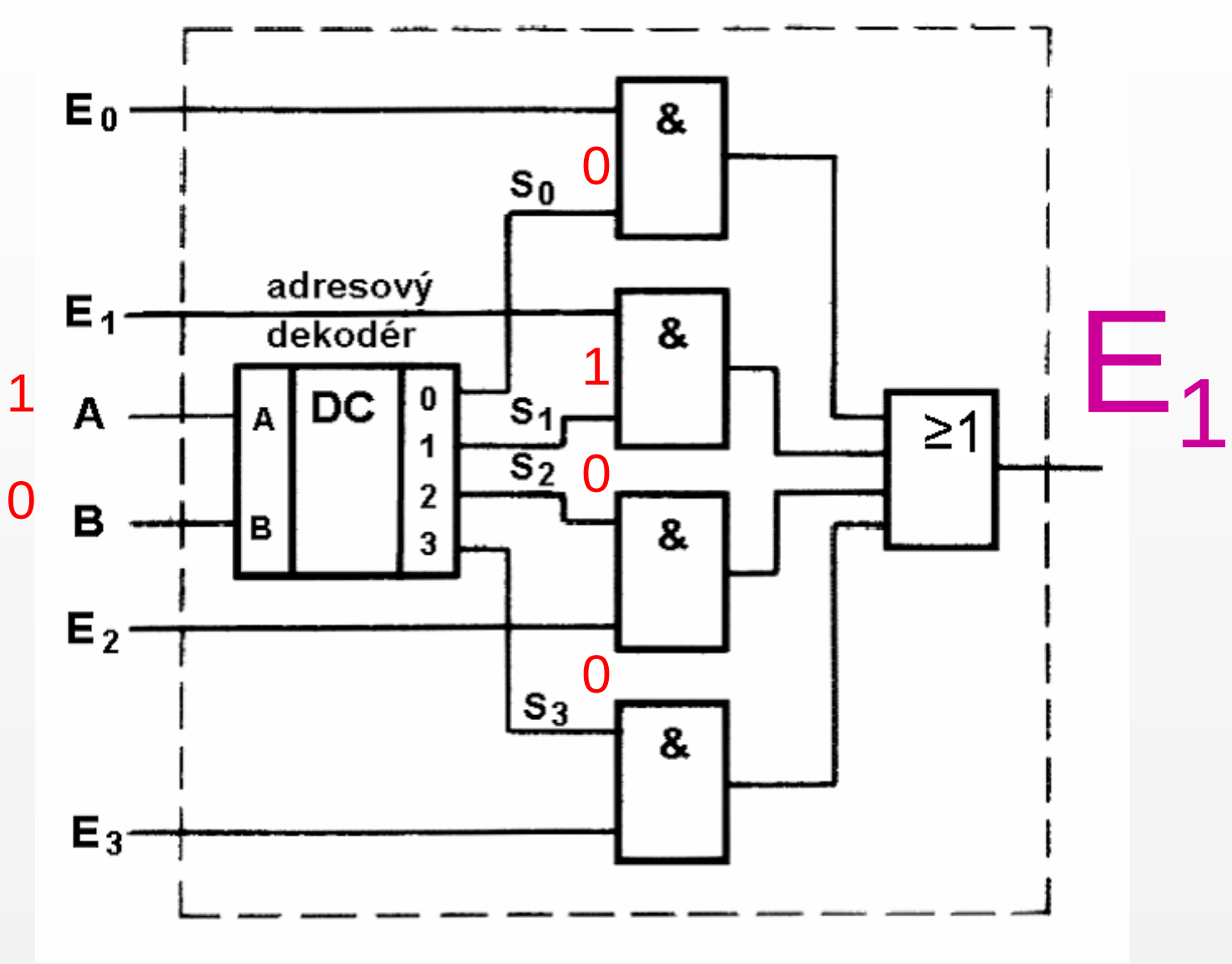
Multiplexer



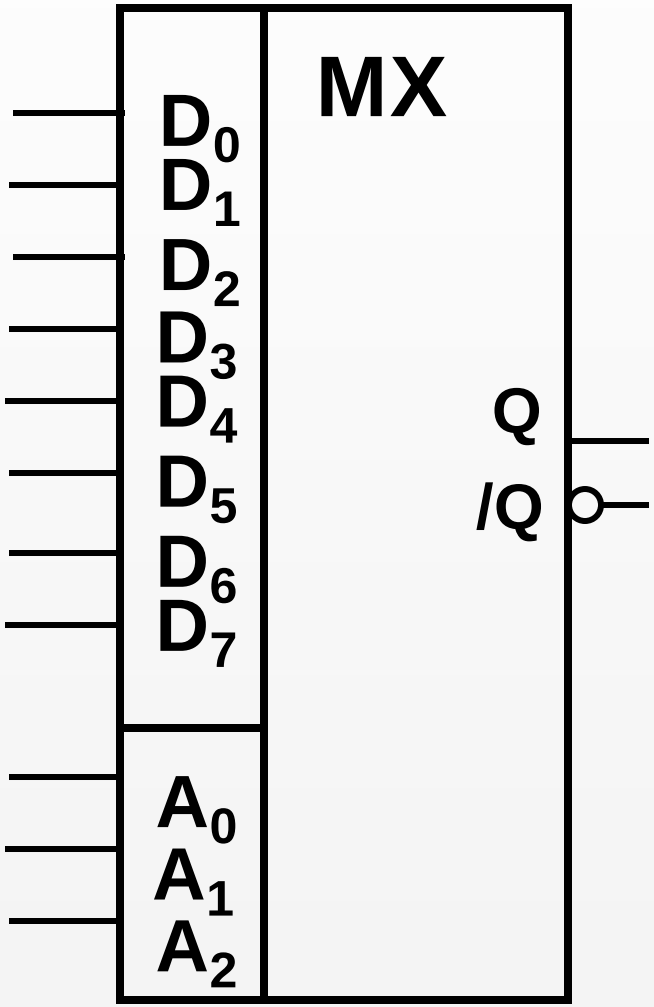
Multiplexer

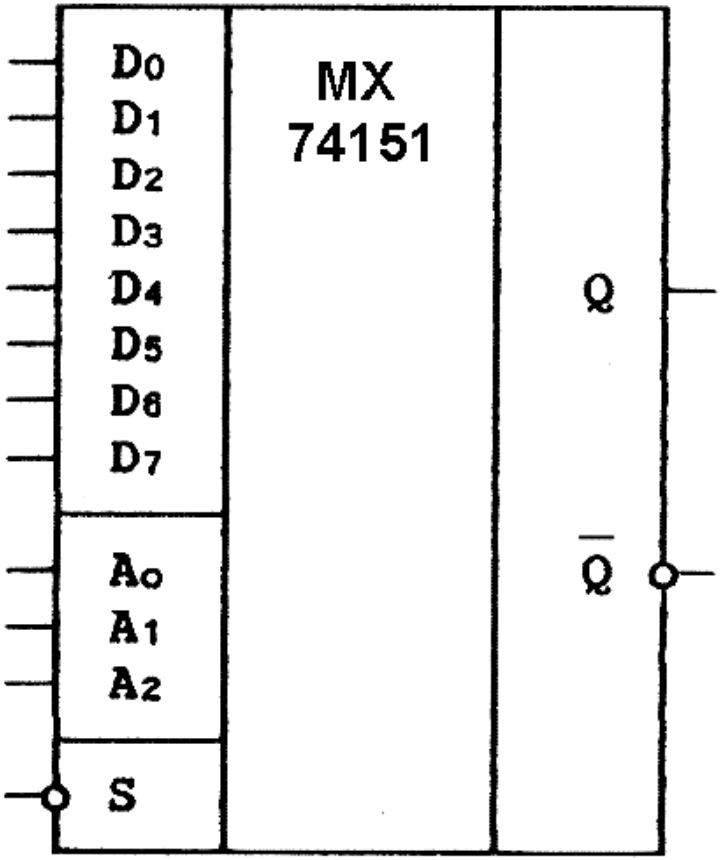


Multiplexer



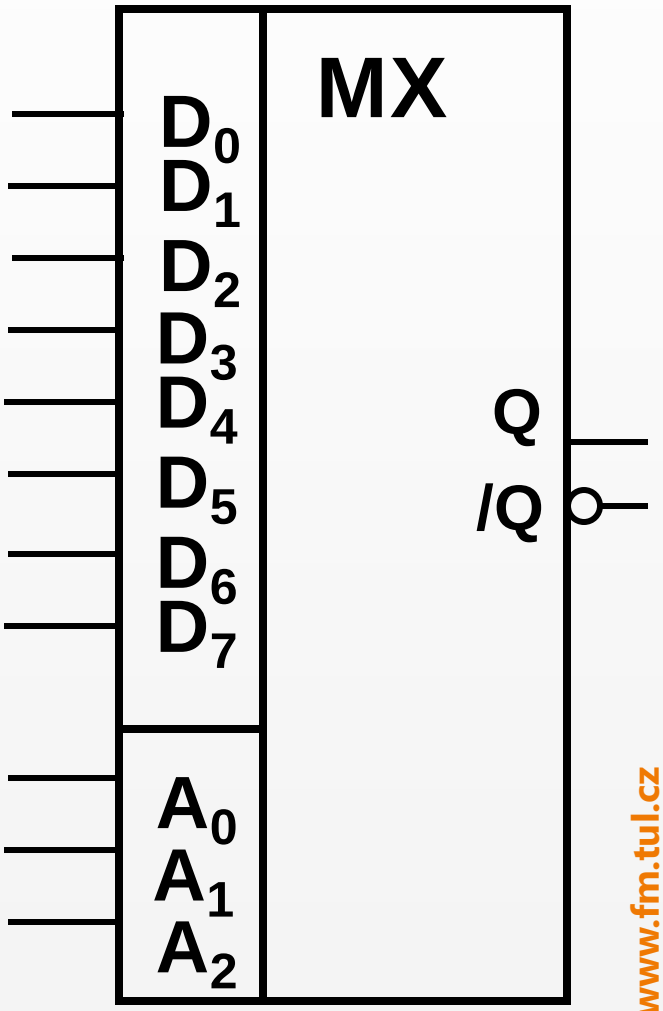
Multiplexer v IO





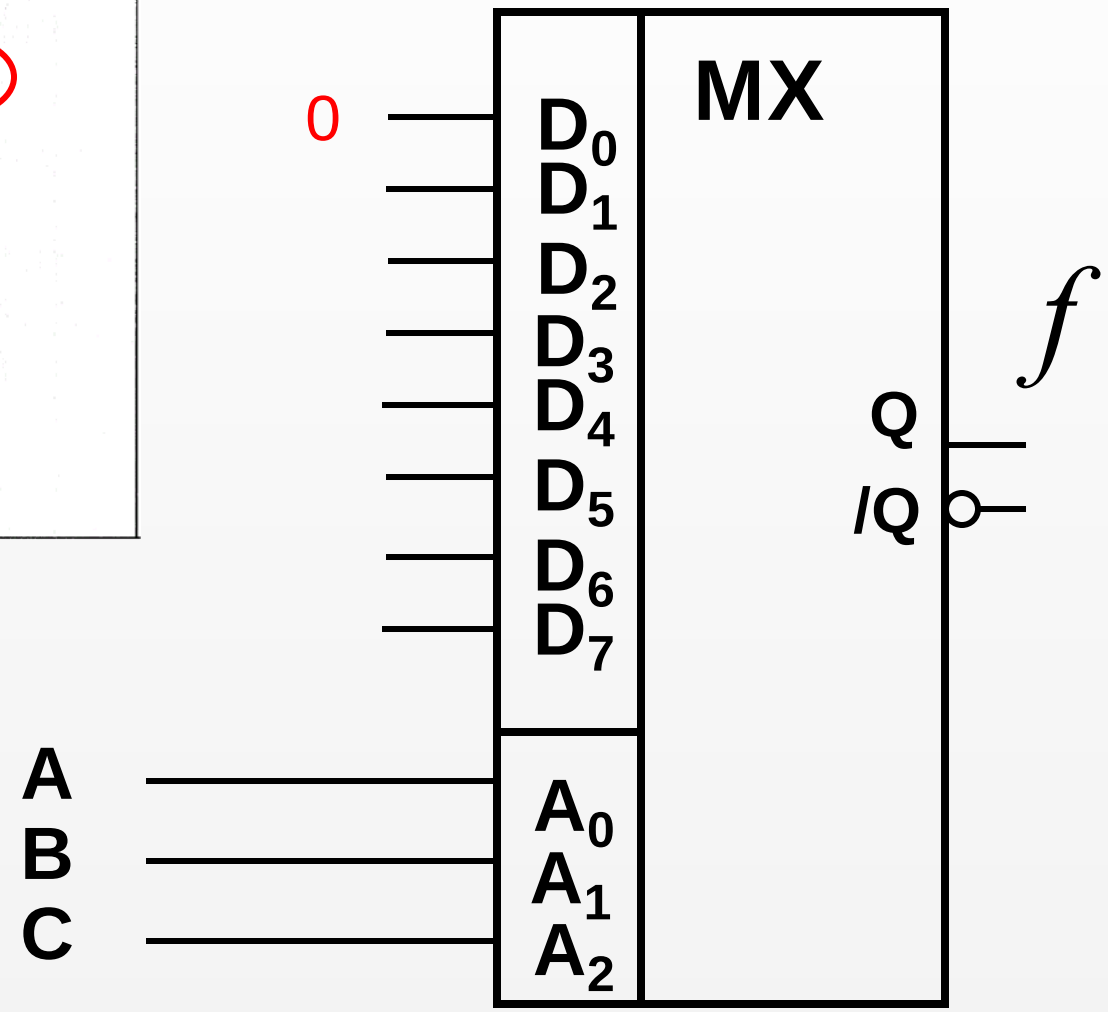
Multiplexer

stavový index s	c b a	funkční hodnota $f(c,b,a)$	minterm P_s
0	0 0 0	0	$\bar{c}\bar{b}\bar{a}$
1	0 0 1	1	$\bar{c}\bar{b}a$
2	0 1 0	1	$\bar{c}b\bar{a}$
3	0 1 1	0	$\bar{c}ba$
4	1 0 0	1	$c\bar{b}\bar{a}$
5	1 0 1	0	$c\bar{b}a$
6	1 1 0	1	$cb\bar{a}$
7	1 1 1	0	cba



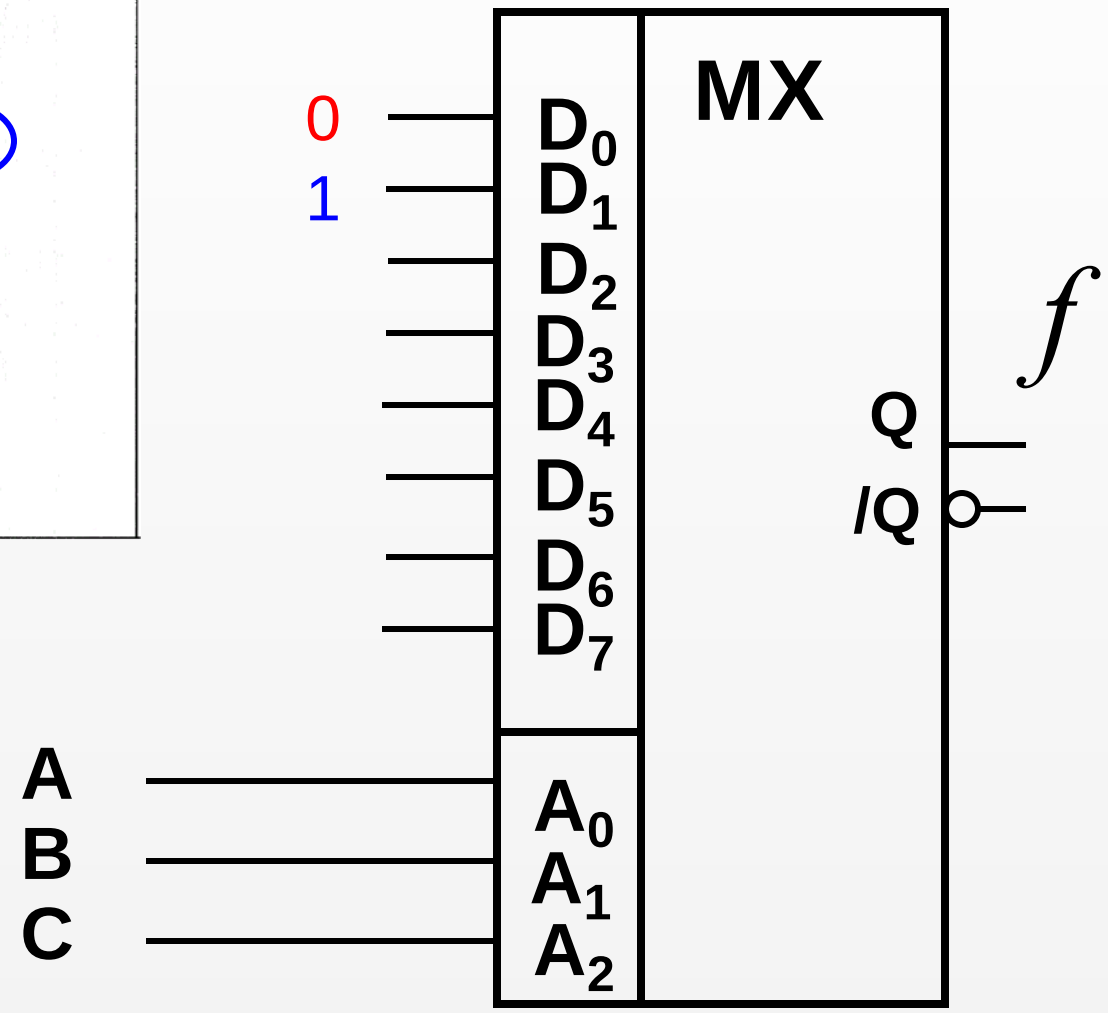
Multiplexer

stavový index s	c b a	funkční hodnota $f(c,b,a)$
0	0 0 0	0
1	0 0 1	1
2	0 1 0	1
3	0 1 1	0
4	1 0 0	1
5	1 0 1	0
6	1 1 0	1
7	1 1 1	0



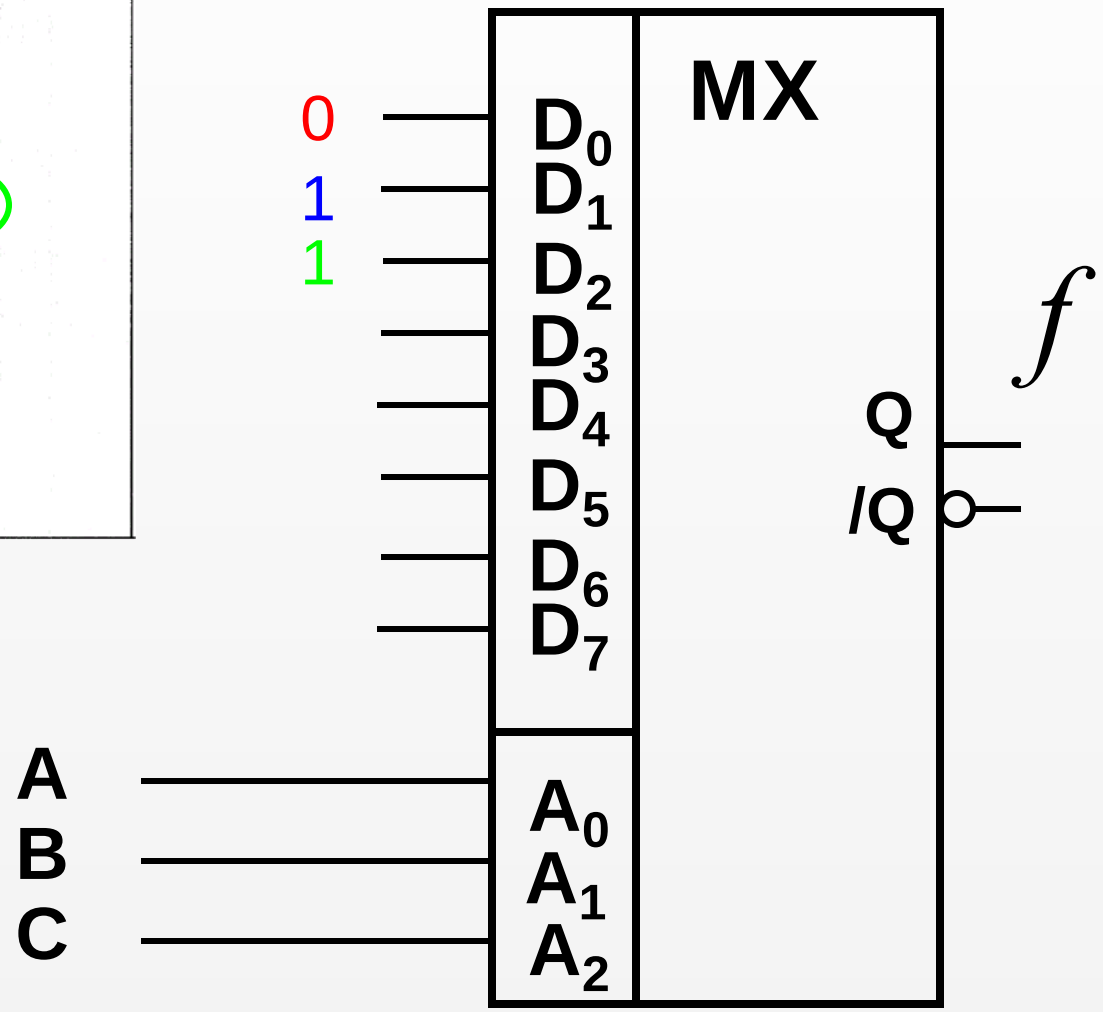
Multiplexer

stavový index s	c b a	funkční hodnota $f(c,b,a)$
0	0 0 0	0
1	0 0 1	1
2	0 1 0	1
3	0 1 1	0
4	1 0 0	1
5	1 0 1	0
6	1 1 0	1
7	1 1 1	0



Multiplexer

stavový index s	c b a	funkční hodnota $f(c,b,a)$
0	0 0 0	0
1	0 0 1	1
2	0 1 0	1
3	0 1 1	0
4	1 0 0	1
5	1 0 1	0
6	1 1 0	1
7	1 1 1	0



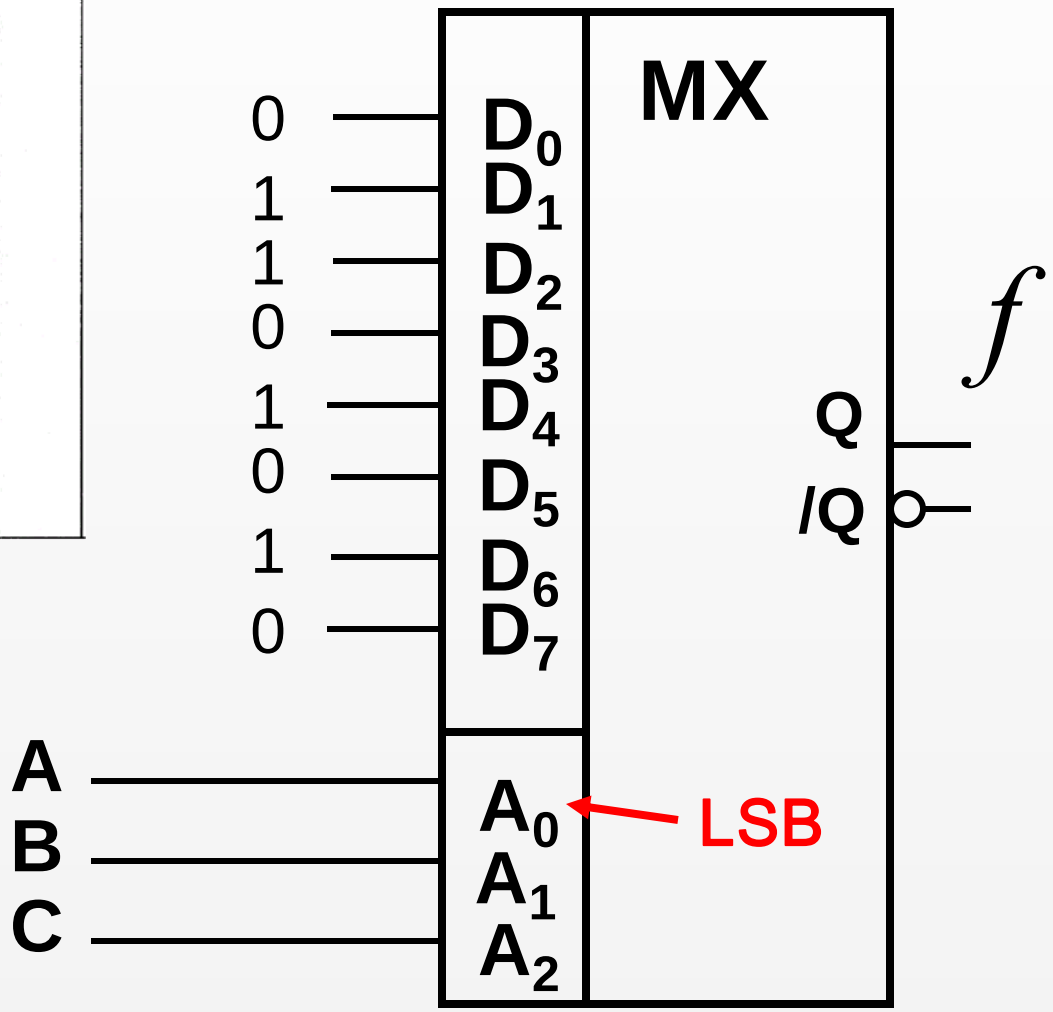
Multiplexer

stavový index s	c b a	funkční hodnota $f(c,b,a)$
0	0 0 0	0
1	0 0 1	1
2	0 1 0	1
3	0 1 1	0
4	1 0 0	1
5	1 0 1	0
6	1 1 0	1
7	1 1 1	0

MSB

LSB

3-vstupy

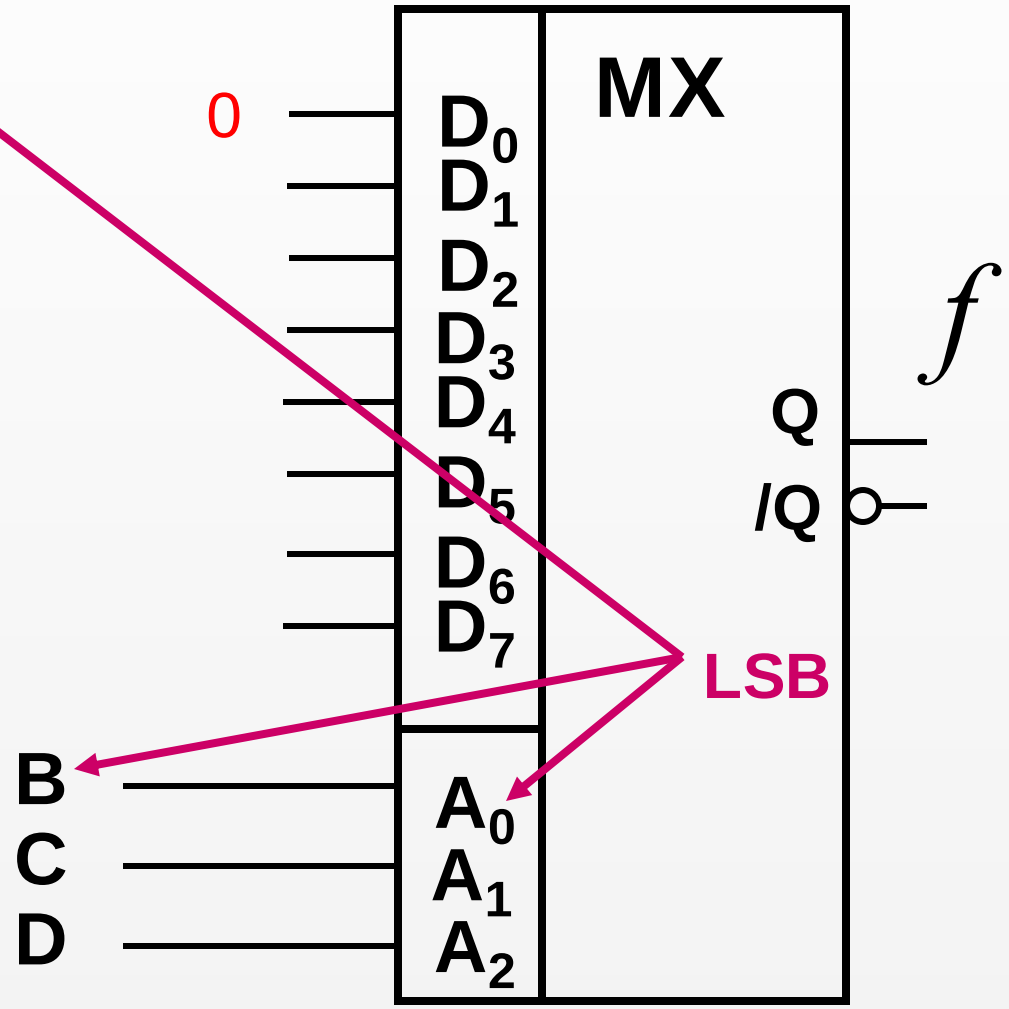


MSB (Most Significant Bit)
LSB (Least Significant Bit)

Multiplexer

D	C	B	A	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

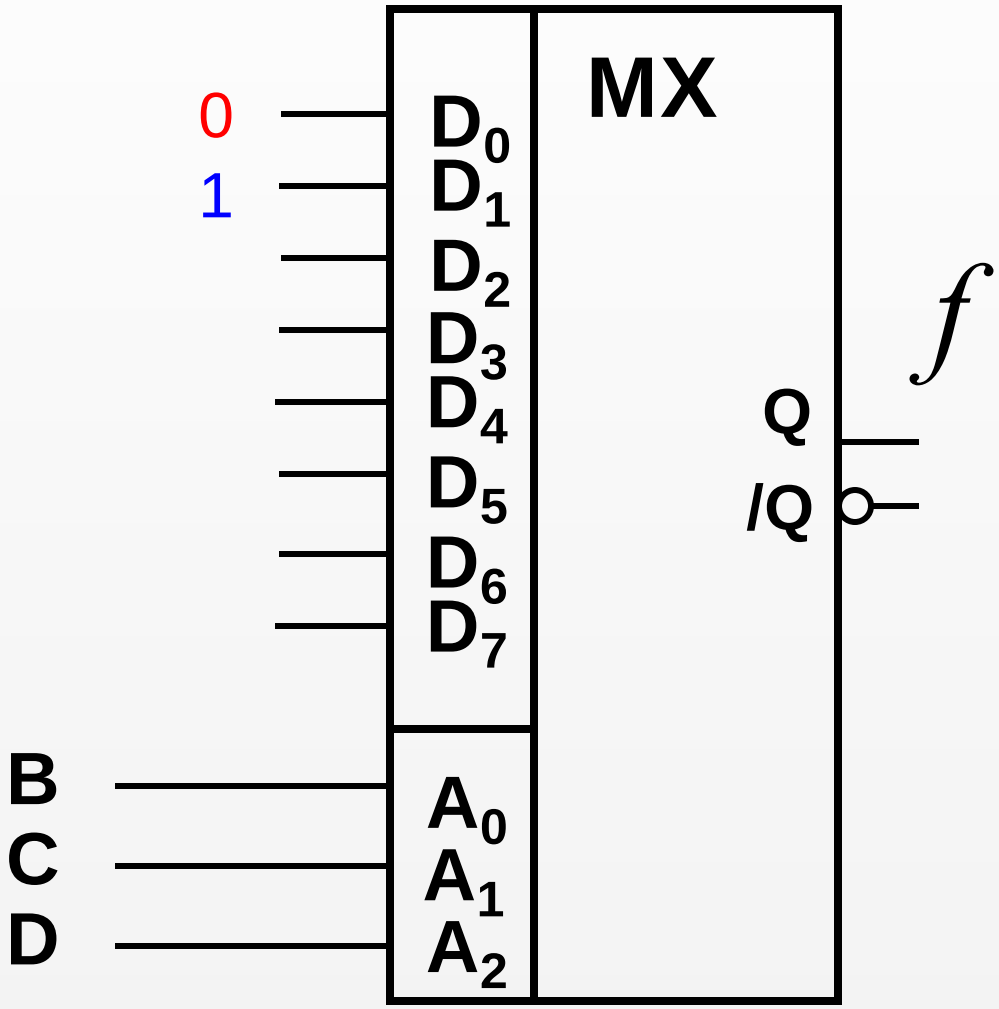
4-vstupy



Multiplexer

D	C	B	A	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

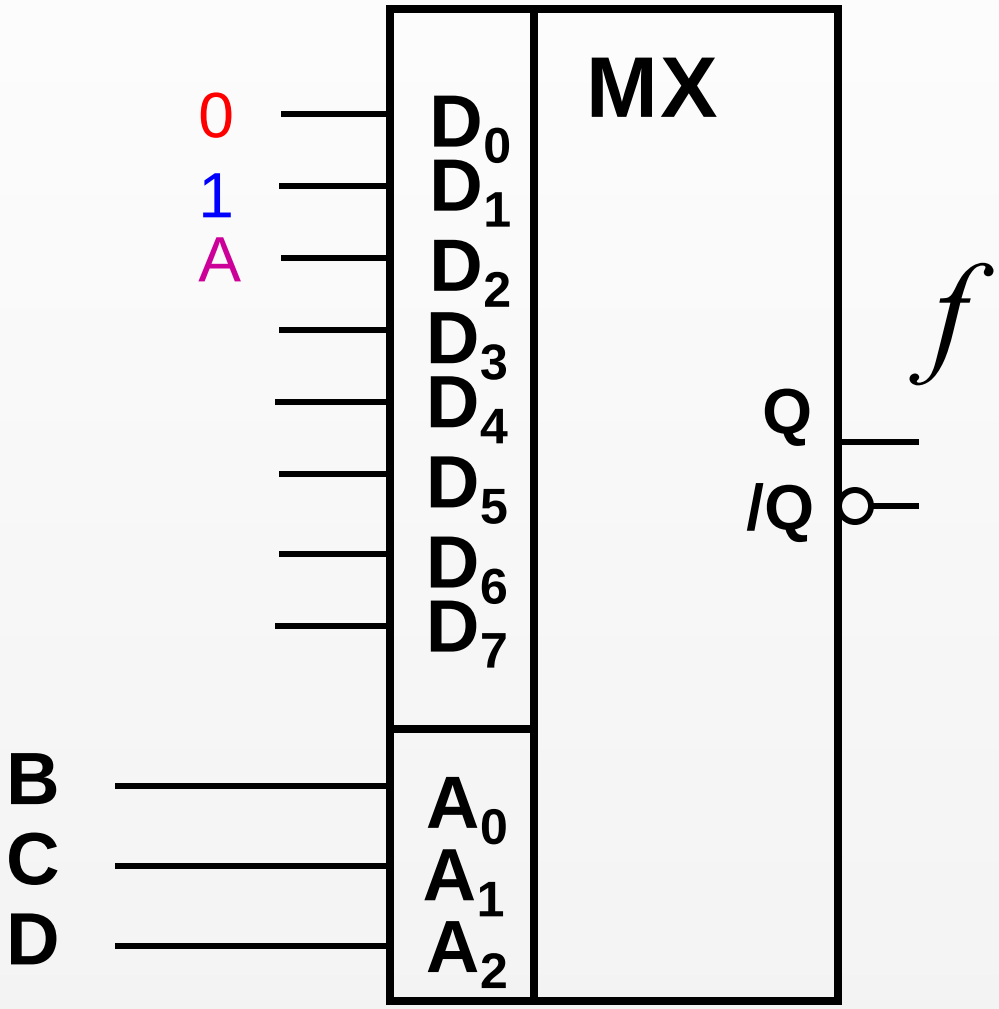
4-vstupy



Multiplexer

D	C	B	A	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

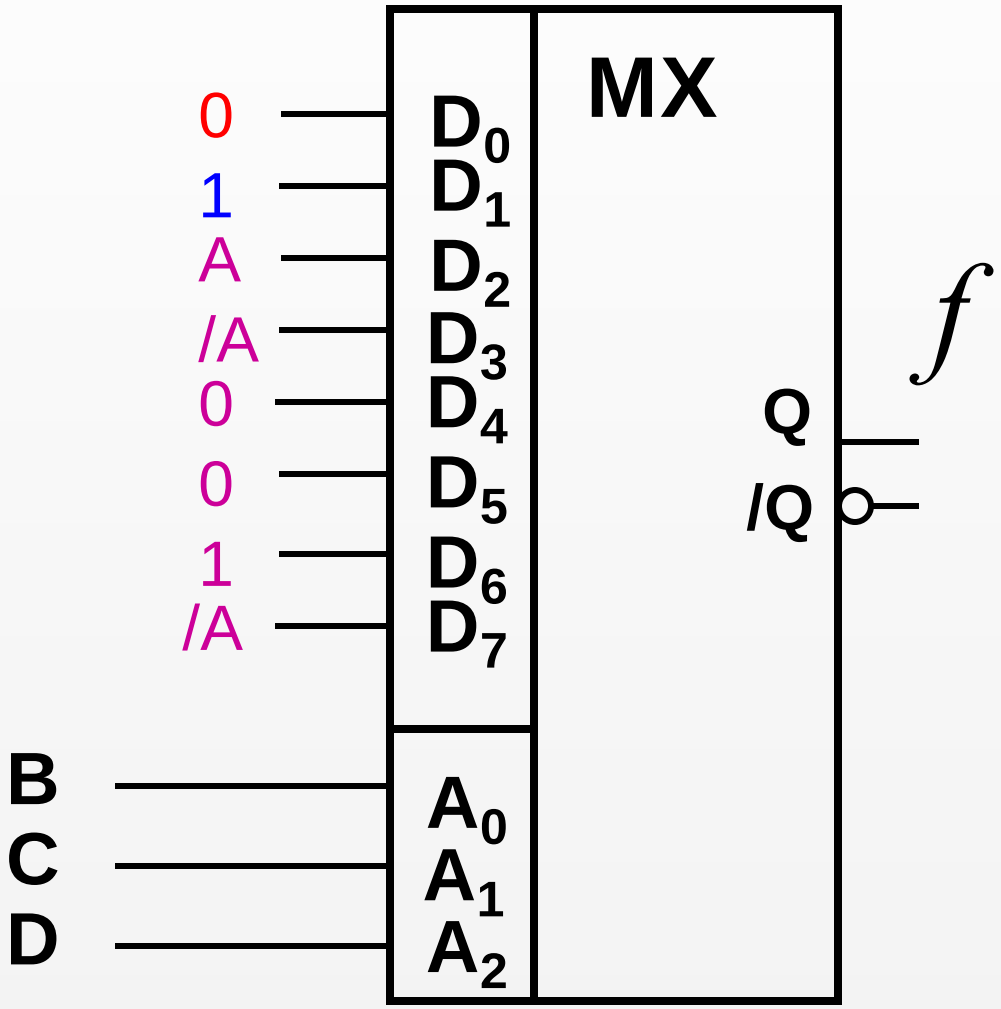
4-vstupy



Multiplexer

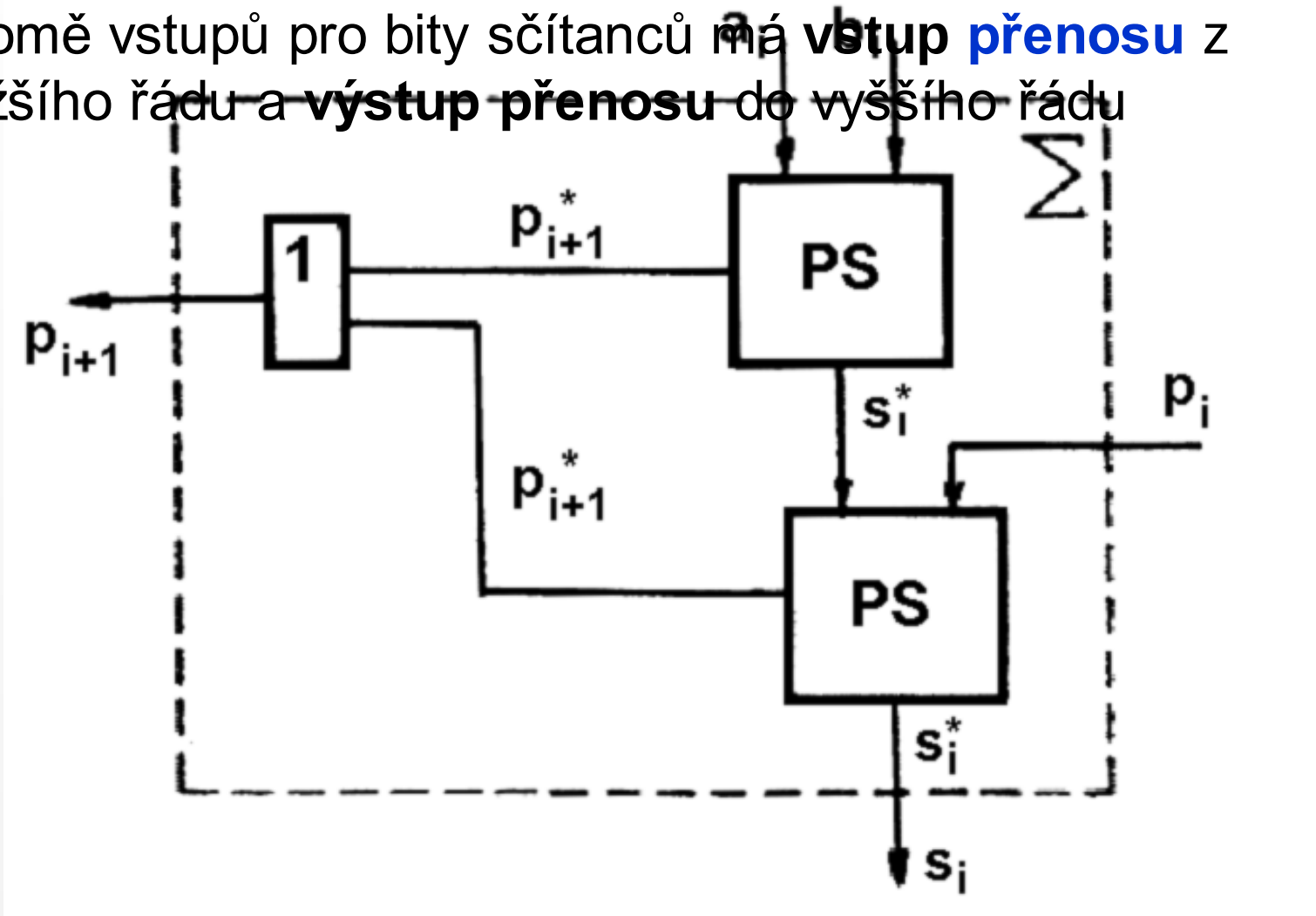
D	C	B	A	*f*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

4-vstupy



Úplná sčítačka

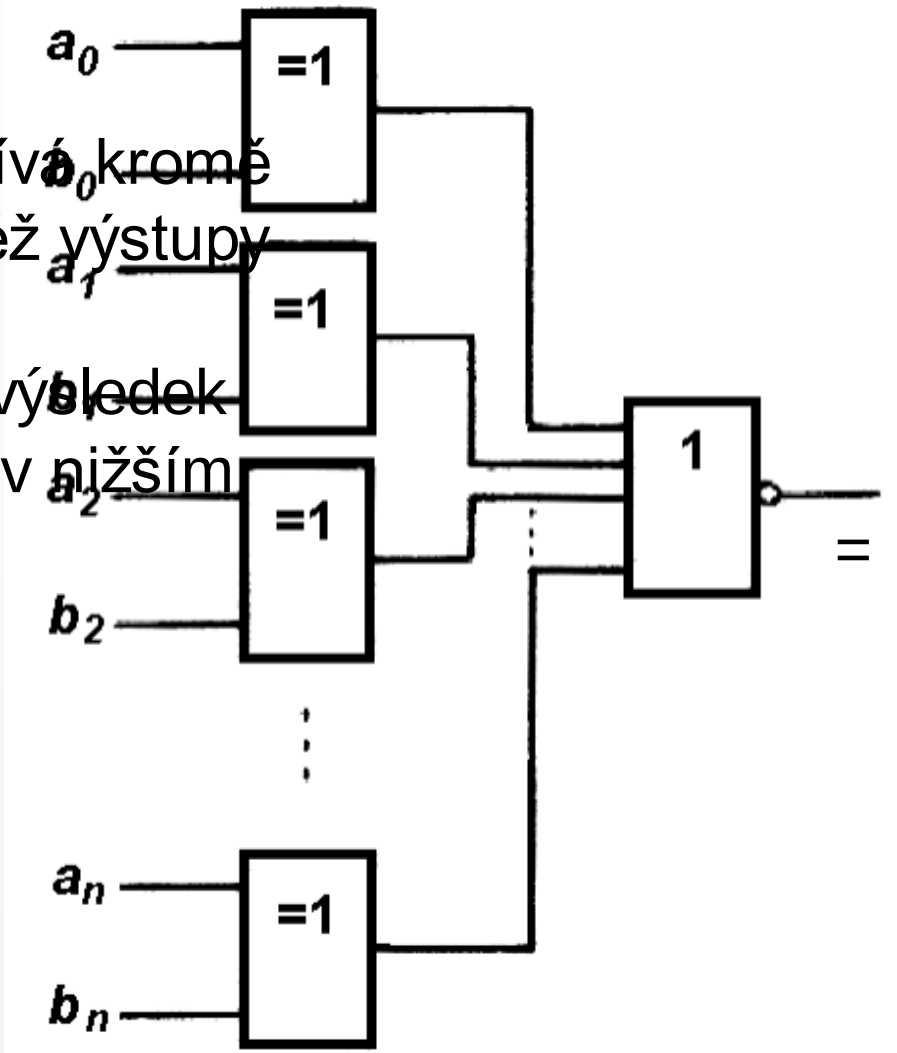
- zpravidla složena ze 2 **sčítaček poloviční**
- kromě vstupů pro bity sčítanců má **vstup přenosu** z nižšího řádu a **výstup přenosu** do vyššího řádu



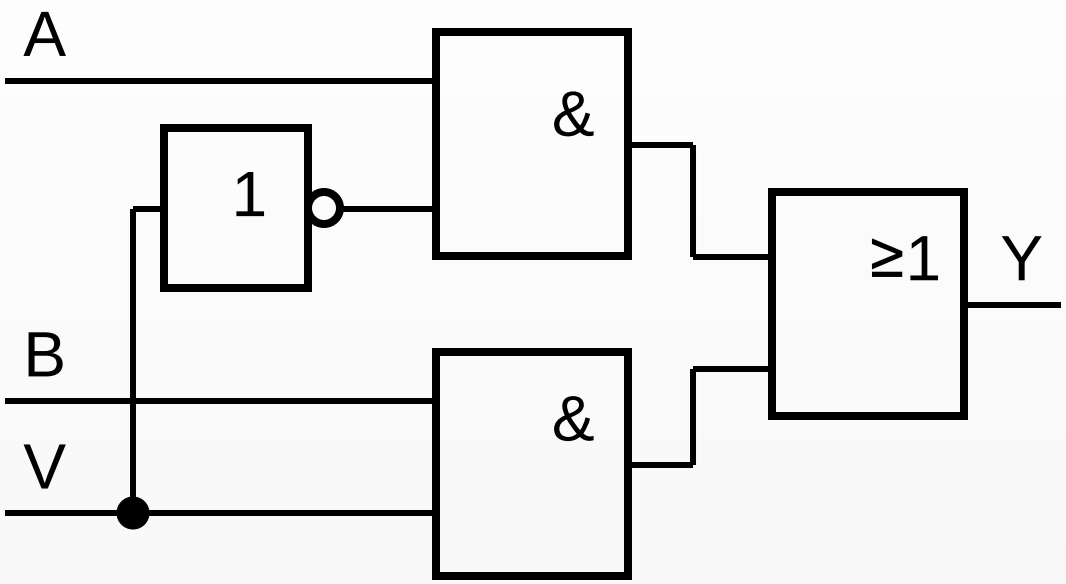
Komparátor (číslicový)

Komparátor dvou čísel A a B

- integrován mívá kromě výstupu „=“ též výstupy „<“ a „>“ a vstupy pro výsledek vyhodnocení v nižším řádu

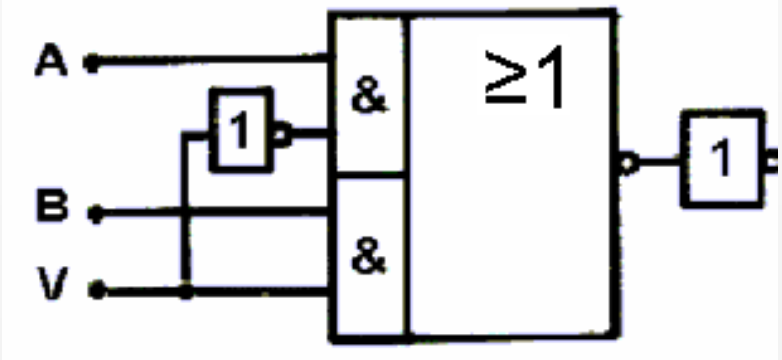


Jednobitová výhybka



V	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

řešení pomocí AND-OR-INVERT



Kódy

- číselné soustavy spolu s pravidly, která říkají, jaké informace (obraz) jsou přiřazeny jednotlivým číslům.

Kódy pro detekci a opravy chyb

- **Paritní kód**
  lichá/sudá parita
- **Kontrolní součet**
  integrita bloku dat,
  např. počet „1“ v bloku nebo součet bajtů modulo 2^{16}
  CRC kódy
- **Samoopravné kódy**
  Hammingovy kódy, kódová vzdálenost

BCD kód

BCD kód - pro práci s čísly desítkové soustavy.

Slouží ke kódování desítkových číslic, tj. 0 … 9.

znak	obraz			
0	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**1**
2	**0**	**0**	**1**	**0**
3	**0**	**0**	**1**	**1**
4	**0**	**1**	**0**	**0**
5	**0**	**1**	**0**	**1**
6	**0**	**1**	**1**	**0**
7	**0**	**1**	**1**	**1**
8	**1**	**0**	**0**	**0**
9	**1**	**0**	**0**	**1**

Grayův kód

Gray kód -
sousední slova se liší pouze v jedné číslici

znak	obraz			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Kód 1 z N

1 z N kód

N-1 nul

jediná jednička

znak	obraz			
0	**0**	**0**	**0**	**1**
1	**0**	**0**	**1**	**0**
2	**0**	**1**	**0**	**0**
3	**1**	**0**	**0**	**0**

Použití u dekodéru…

Alfanumerické kódy - ASCII

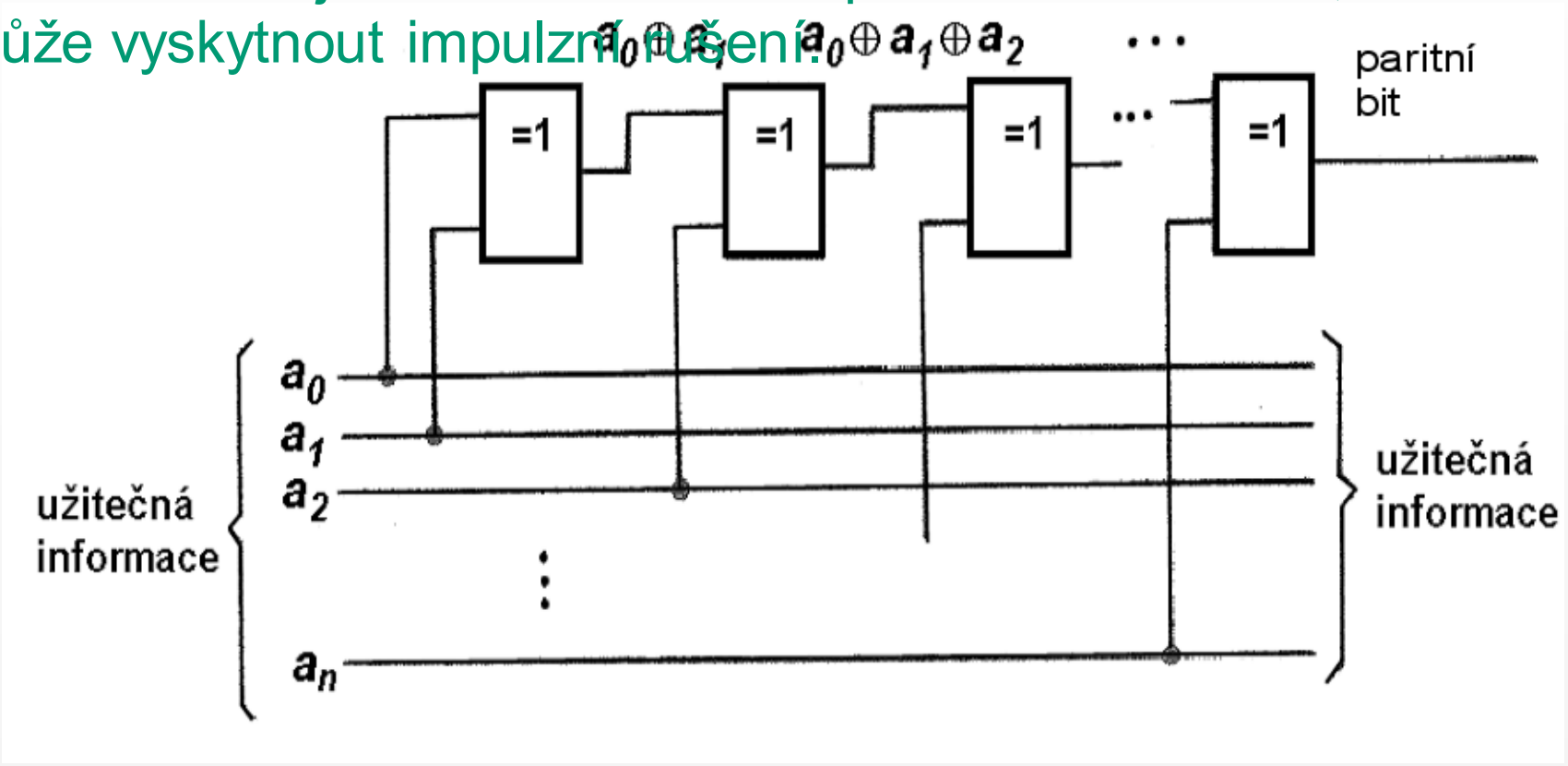
Dec	Hex	IBM	Dec	Hex	IBM	Dec	Hex	IBM	ISO	Win	Kam	Lat	Dec	Hex	IBM	ISO	Win	Kam	Lat
0	0		64	40	@	128	80	Ç			Č	Ç	192	C0	└	Ŕ	Ŕ	└	└
1	1	☺	65	41	A	129	81	ü			ü	ü	193	C1	┴	Á	Á	┴	┴
2	2	☻	66	42	B	130	82	é		,	é	é	194	C2	┬	Â	Â	┬	┬
3	3	♥	67	43	C	131	83	â			ď	â	195	C3	├	Ă	Ă	├	├
4	4	♦	68	44	D	132	84	ä		„	ä	ä	196	C4	─	Ä	Ä	─	─
5	5	♣	69	45	E	133	85	à		…	Ď	ů	197	C5	┼	Ĺ	Ĺ	┼	┼
6	6	♠	70	46	F	134	86	å		†	Ť	ć	198	C6	╞	Ć	Ć	╞	Ă
7	7	•	71	47	G	135	87	ç		‡	č	ç	199	C7	╟	Ç	Ç	╟	ă
8	8	◘	72	48	H	136	88	ê		ˆ	ě	ł	200	C8	╚	Č	Č	╚	╚
9	9	○	73	49	I	137	89	ë		‰	Ě	ë	201	C9	╔	É	É	╔	╔
10	A	◙	74	4A	J	138	8A	è		Š	Ĺ	Ő	202	CA	╩	Ę	Ę	╩	╩
11	B	♂	75	4B	K	139	8B	ï		<	Í	ő	203	CB	╦	Ë	Ë	╦	╦
12	C	♀	76	4C	L	140	8C	î		Ś	Ľ	î	204	CC	╠	Ě	Ě	╠	╠
13	D	♪	77	4D	M	141	8D	ì		Ť	Í	Ź	205	CD	═	Í	Í	═	═
14	E	♫	78	4E	N	142	8E	Ä		Ž	Ä	Ä	206	CE	╬	Î	Î	╬	╬
15	F	☼	79	4F	O	143	8F	Å		Ź	Á	Ć	207	CF	╧	Đ	Đ	╧	¤

Toto je pouze část ASCII

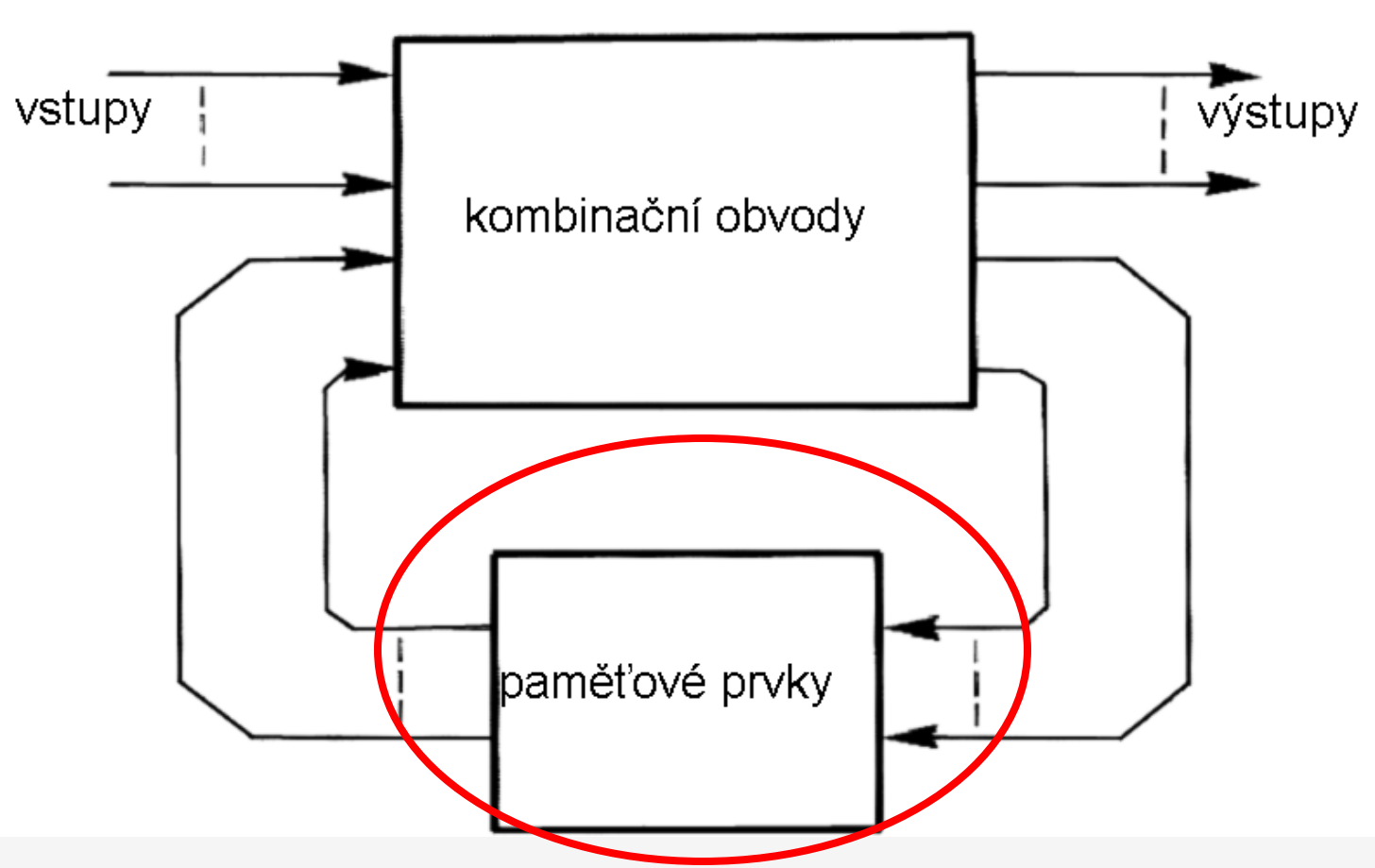
Parita

Počet jedniček ve slově doplněn paritním bitem na **sudý** (obr.) / **lichý** počet.

Používá se zejména u **sériového** přenosu informace, kde se může vyskytnout impulzní rušení.



Sekvenční obvody

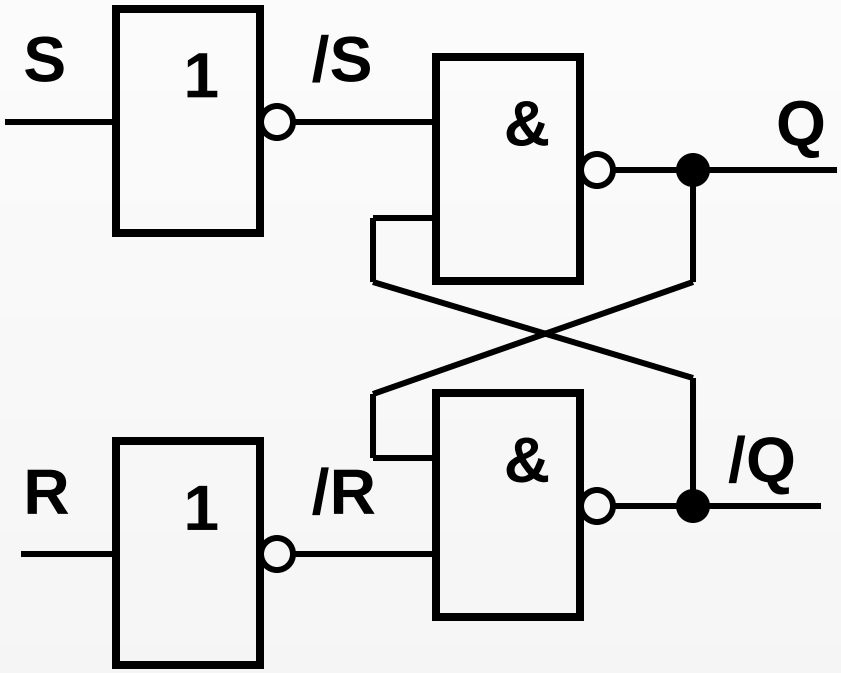


výstupy jsou určeny hodnotou **vstupů** a **vnitřním** (předcházejícím) stavem

Sekvenční obvody

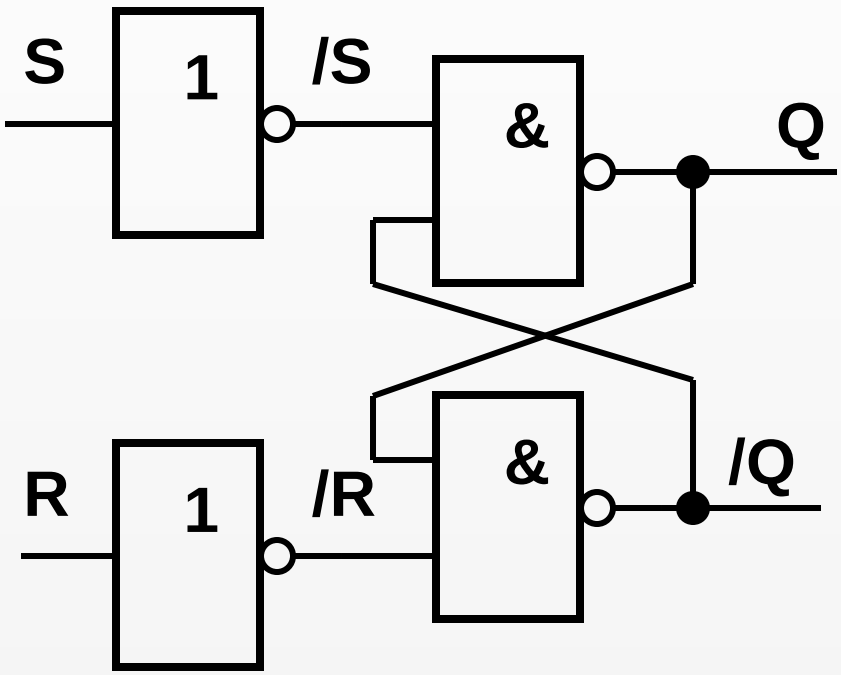
- Rozdělení sekvenčních obvodů
- RS klopný obvod
- D-klopný obvod
- JK-klopný obvod
- Konstrukce Master-Slave obvodů
- Čítače
- Registry
- Realizace sekvenčních úloh

R-S klopný obvod



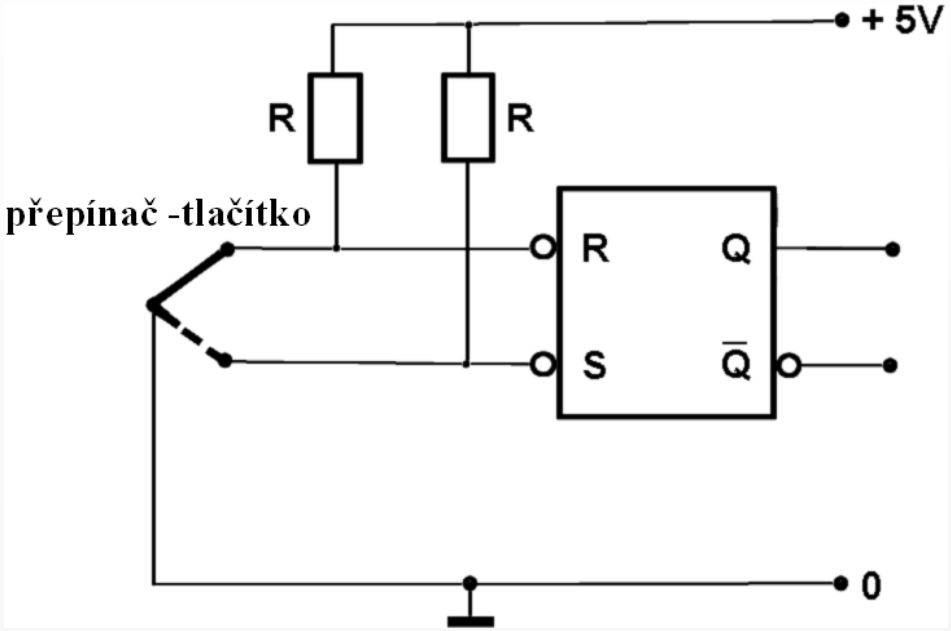
R	S	Q_t
0	0	Q_{t-1}
0	1	1
1	0	0
1	1	X

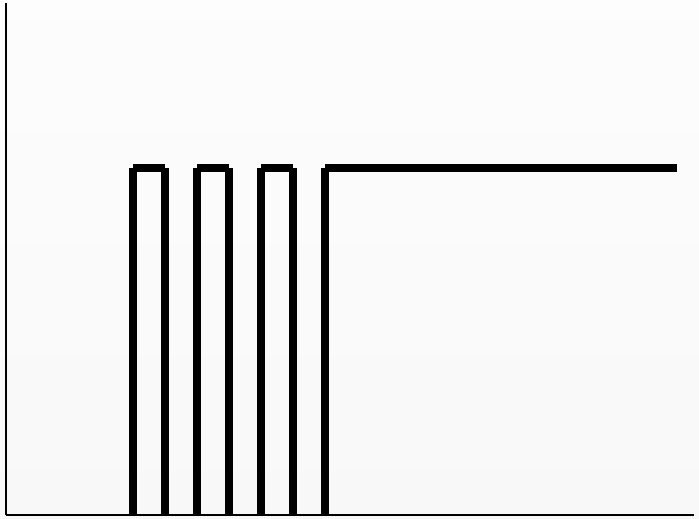
R-S klopný obvod



R	S	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	X (1)
1	1	1	X (1)

Použití RS obvodu





- Ošetření odskoku kontaktu tlačítka
- Asynchronní nastavování výstupu obvodu
- atd.

Rozdělení klopných obvodů

- **Monostabilní** - upravují zachycené impulsy na impulsy s předem stanovenou délkou
- **Astabilní** - slouží ke generování period. signálu
- **Bistabilní** - mají dva stabilní stavy

Rozdělení bistabilních klopných obvodů:

- **Asynchronní** - Asynchronní obvody reagují na všechny změny vstupního signálu (jednodušší než synchronní)
- **Synchronní** - Synchronní obvody reagují na vstupní signály pouze v okamžicích, kdy je aktivní hodinový signál, povolující změnu stavu

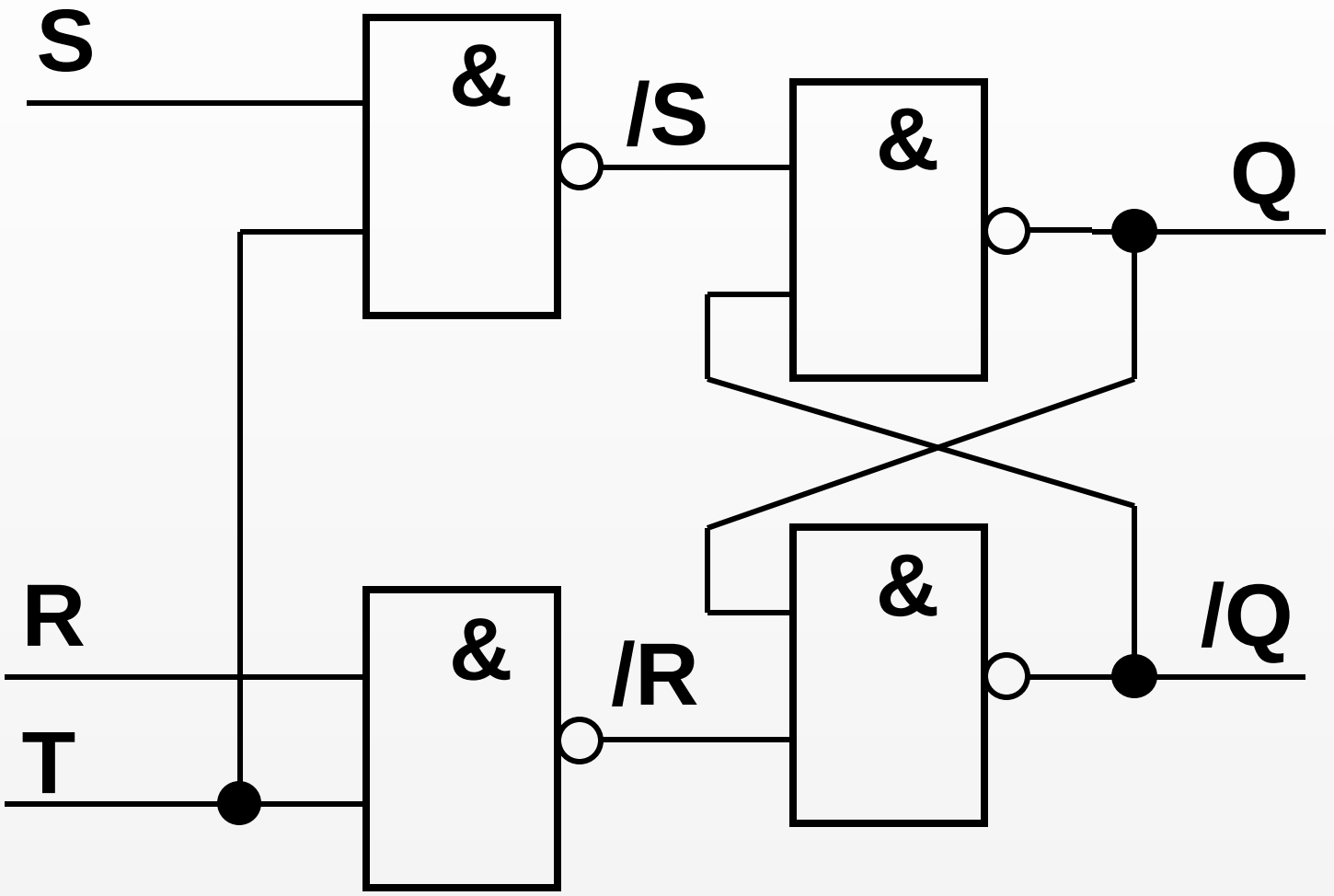
Vlastnosti Asynchronních obvodů

- **Reagují okamžitě** po změně vstupních proměnných → obtíže při návrhu sekvenčních obvodů (obvody připojené na vstupu klopného obvodu mohou vlivem hazardů způsobit nechtěné překlopení obvodu).
- Při návrhu asynchronních obvodů je tedy třeba důsledně **hazardy odstranit a zajistit stabilitu** jednotlivých stavů.

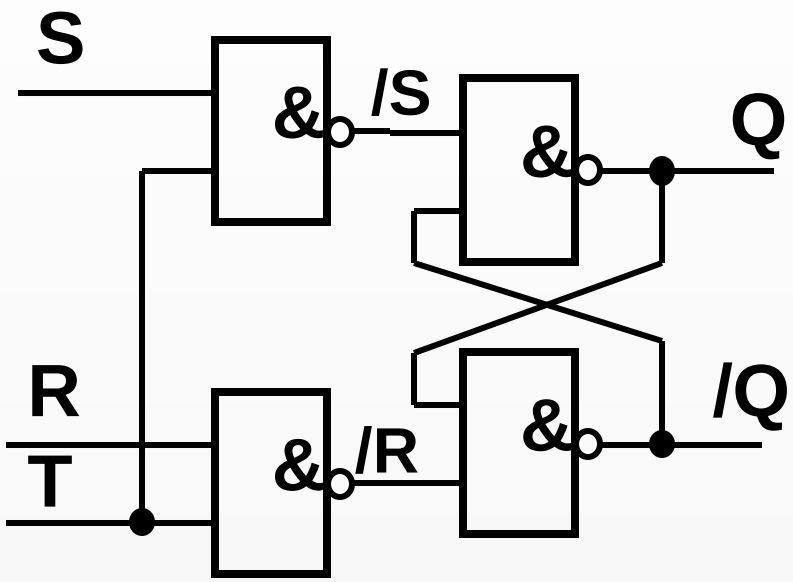
Vlastnosti Asynchronních obvodů

- Takovýto způsob návrhu je obtížný, pro složitější obvody je prakticky neřešitelný a proto se více než asynchronních obvodů v praxi využívá obvodů **synchronních**, které reagují na vstupní signály pouze v těch okamžicích, kdy jsou všechny logické hodnoty na vstupu ustáleny.

Synchronní RS klopný obvod



Synchronní obvody

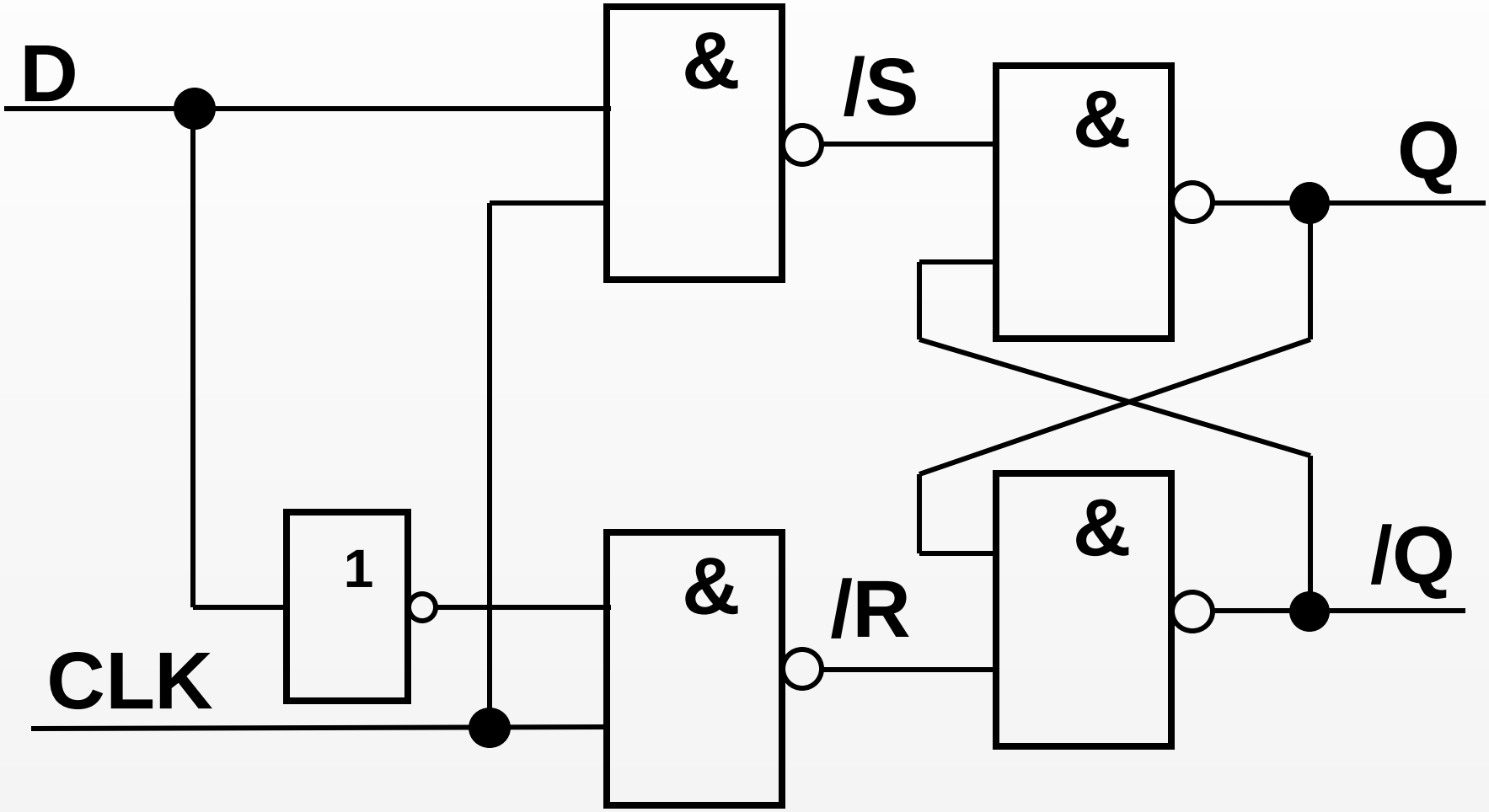


Synchronní obvody mají navíc další synchronizační vstup, který umožňuje znecitlivět ostatní vstupy až do doby, kdy jsou zajištěny podmínky pro správnou funkci obvodu (bezhazardnost).

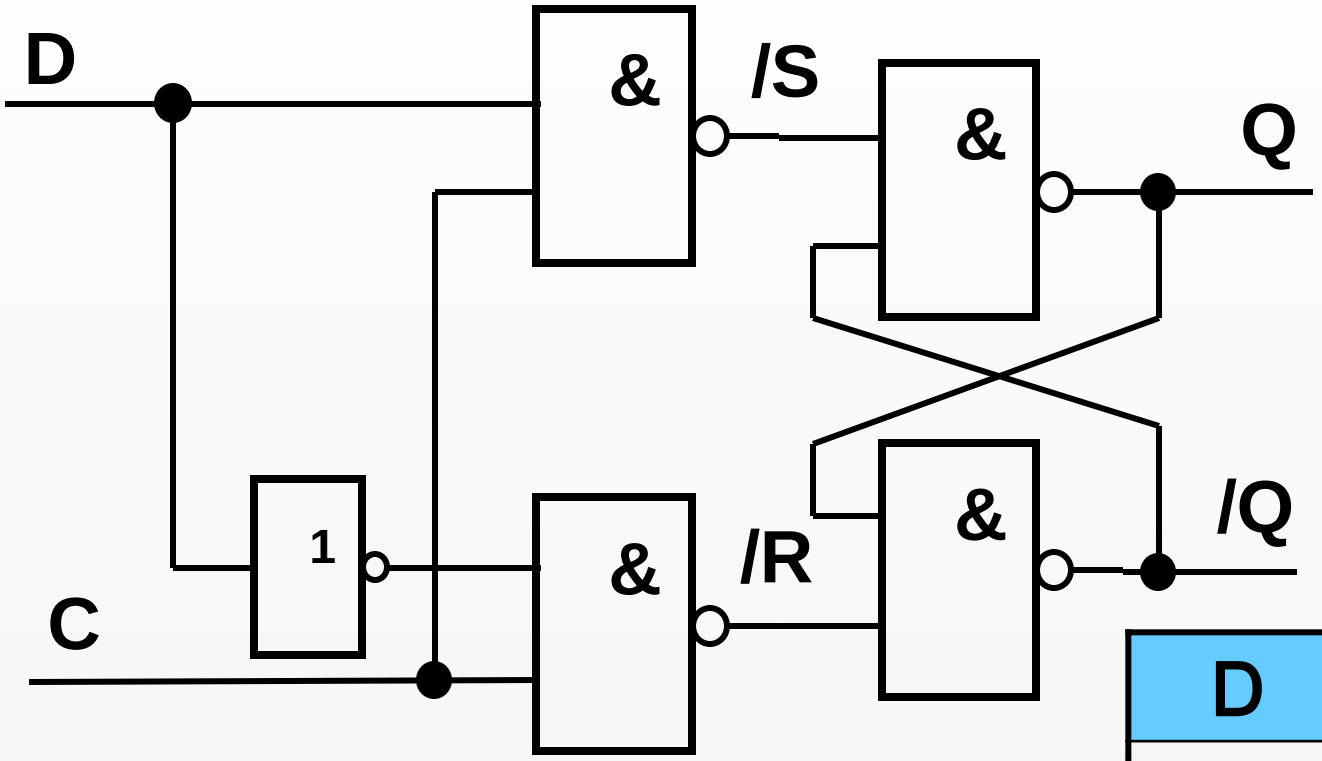
Tento vstup bývá označen zpravidla zkratkou T, C nebo CLK - hodinový vstup.

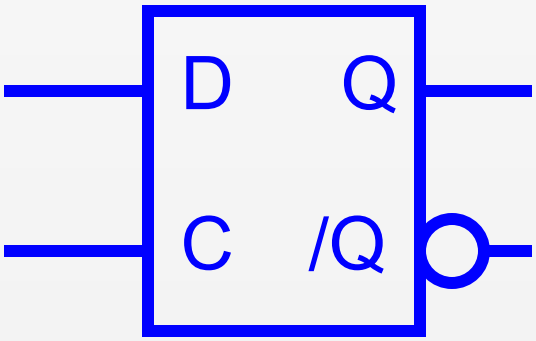
Hodinové vstupy bývají buzeny zpravidla periodickým signálem, tvořeným krátkými impulsy a nazývaným hodinový signál.

D-klopný obvod – hladinový



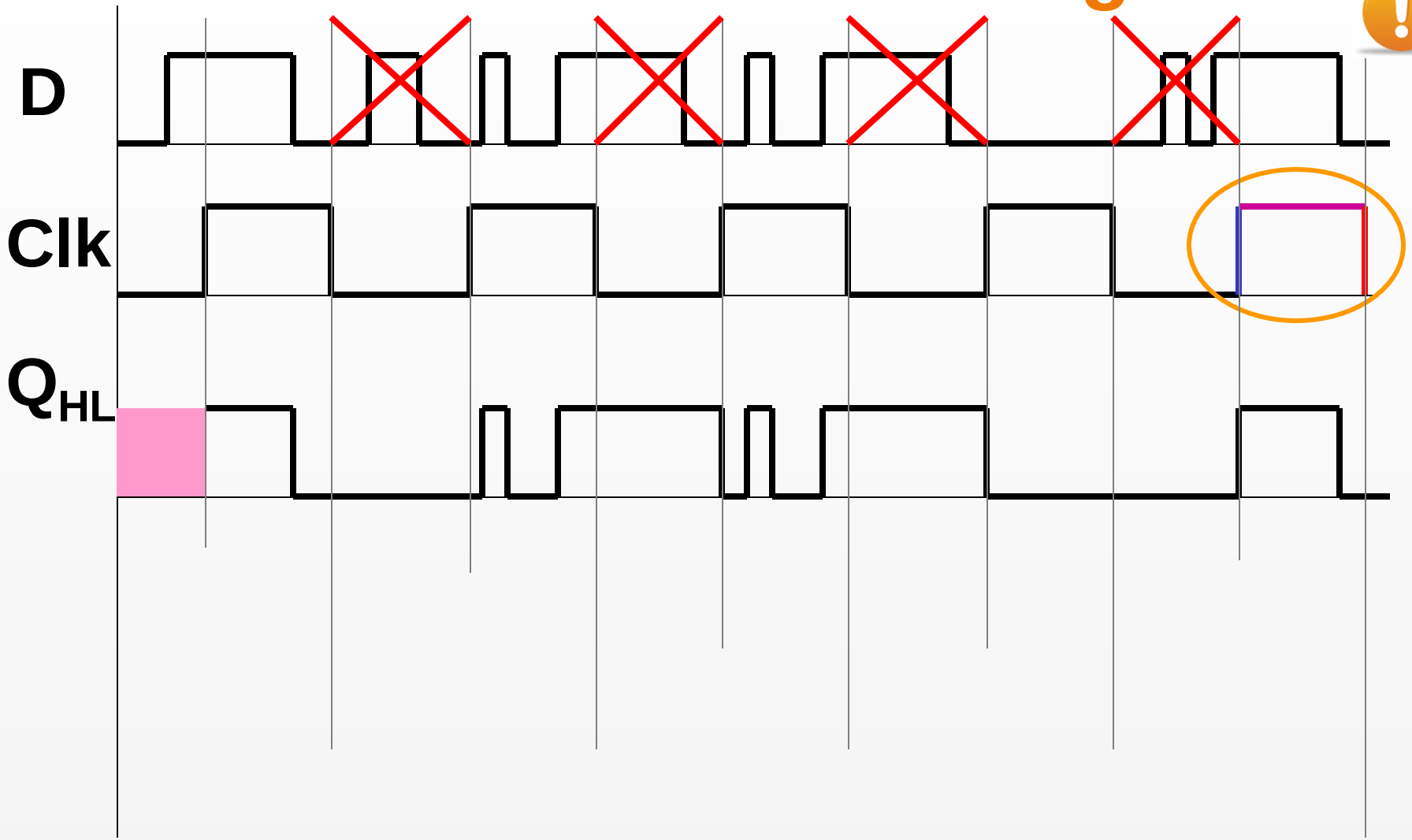
D-klopný obvod – hladinový





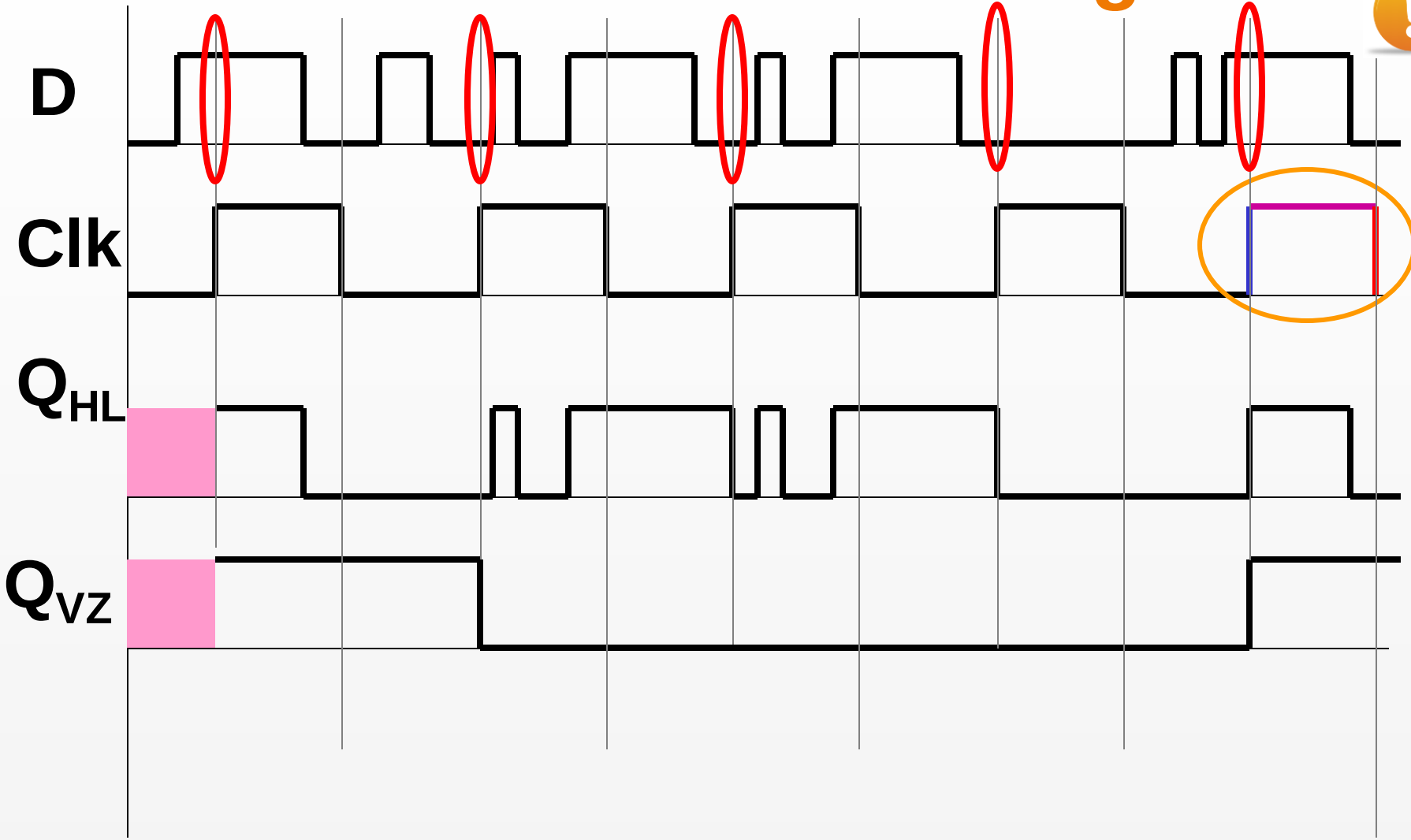
D	C	Q_t
0	0	Q_{t-1}
1	0	Q_{t-1}
0	1	0
1	1	1

Reakce D-KO na hod. signál



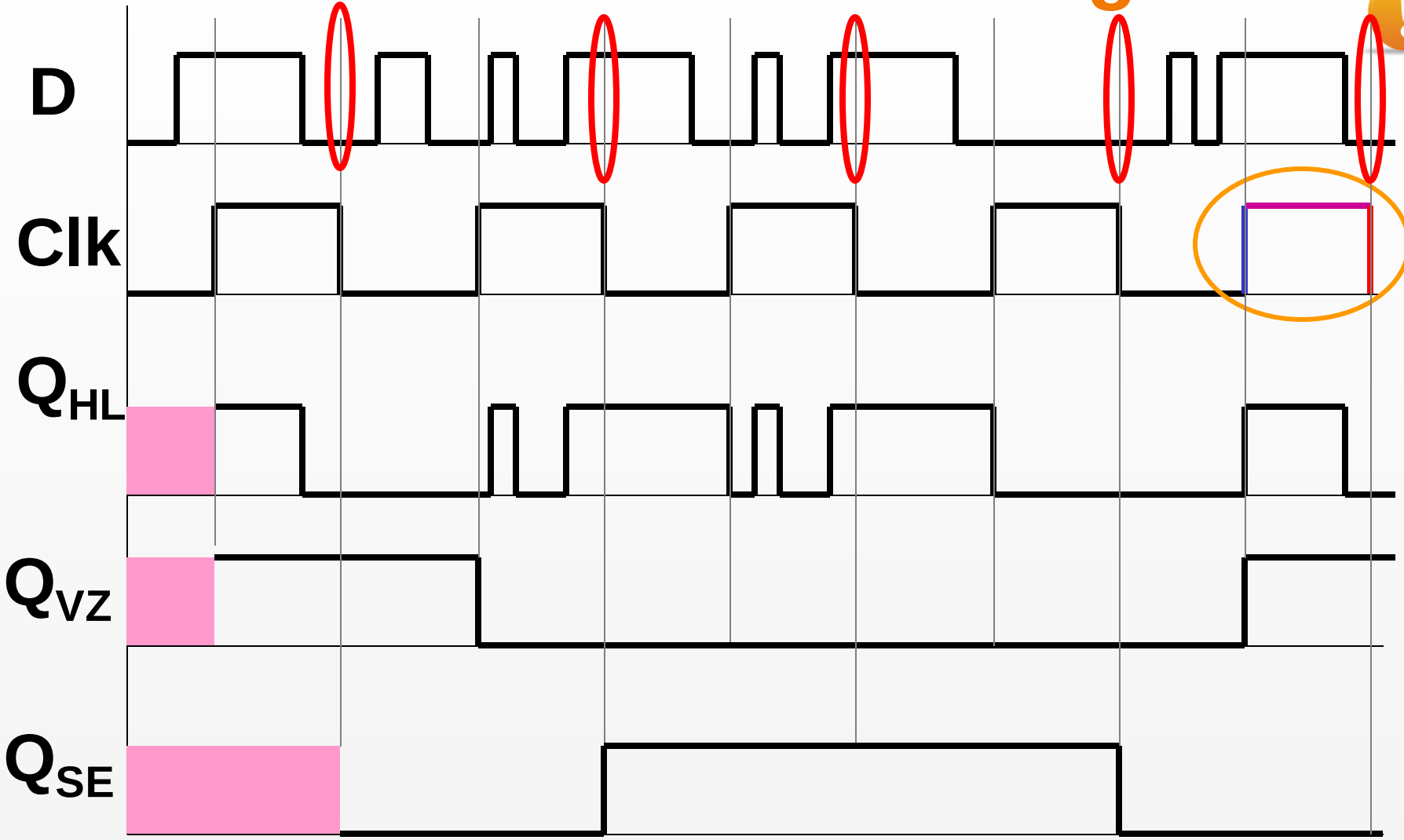
Q_{HL}-hladinový

Reakce D-KO na hod. signál



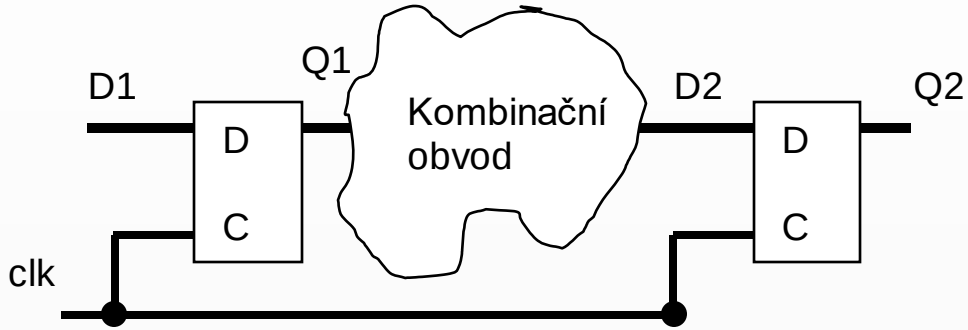
Q_{HL}-hladinový Q_{VZ}-vzestupná

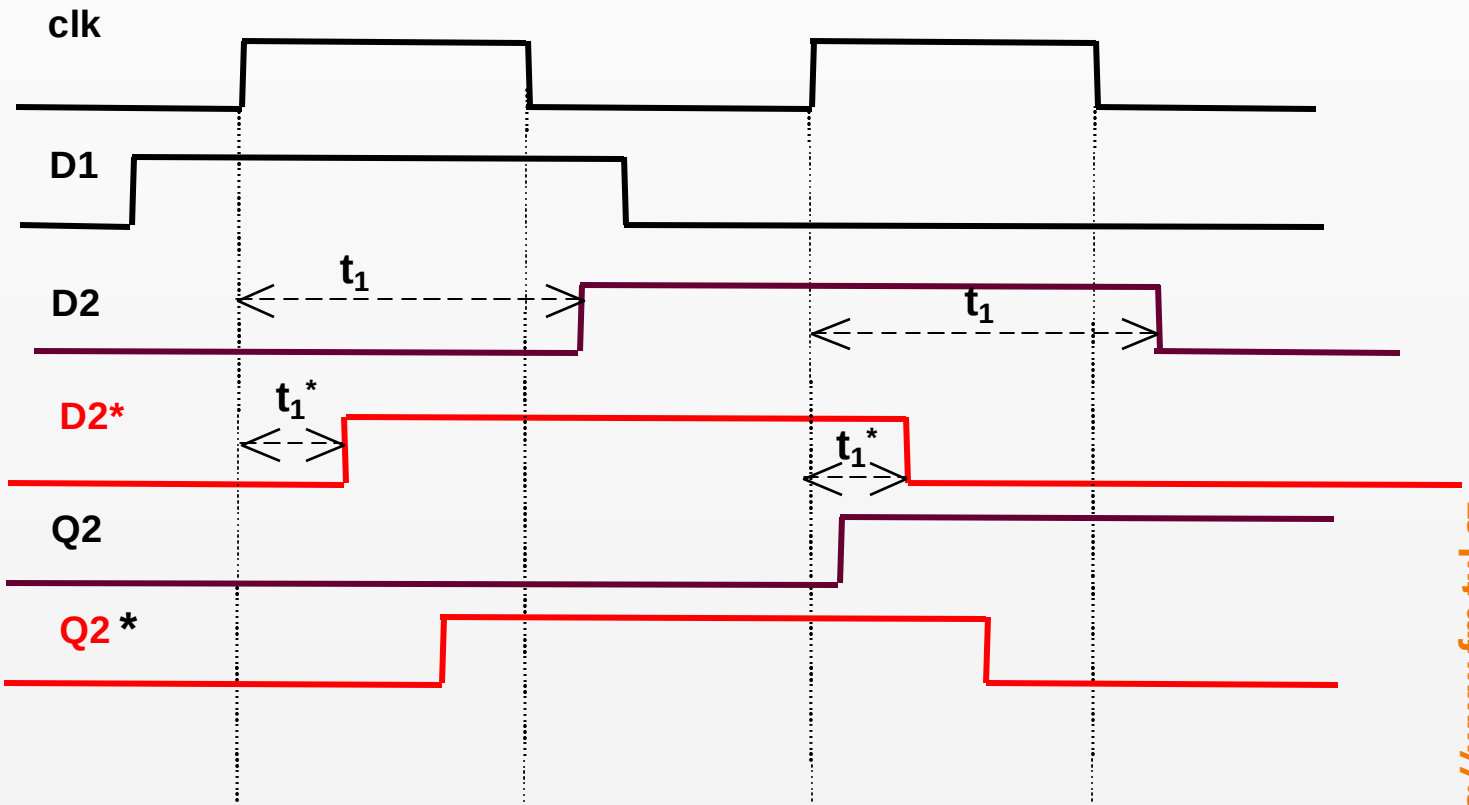
Reakce D-KO na hod. signál



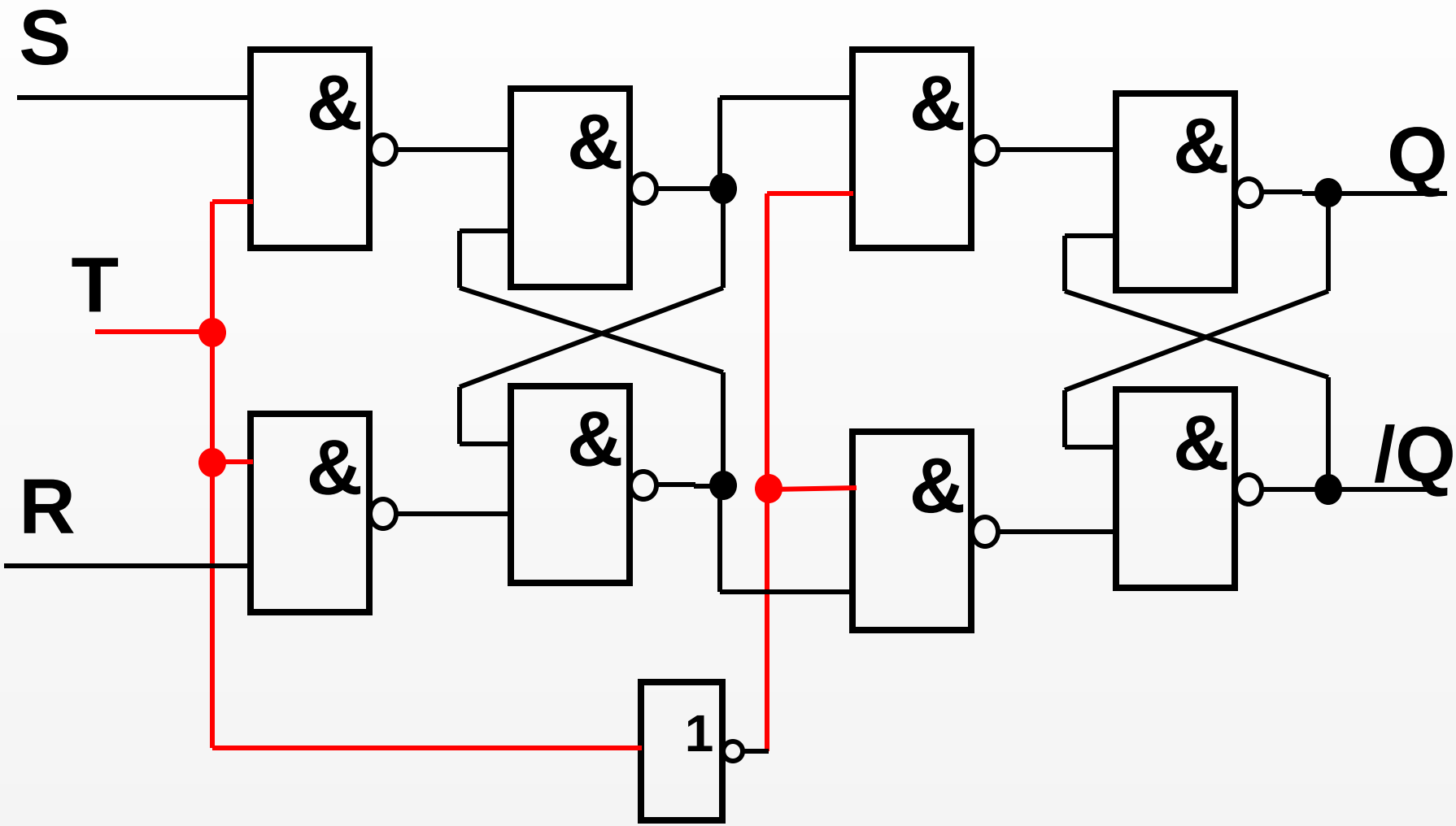
Q_{HL}-hladinový Q_{VZ}-vzestupná Q_{SE}-sestupná

Hazard v sekvenčních obvodech

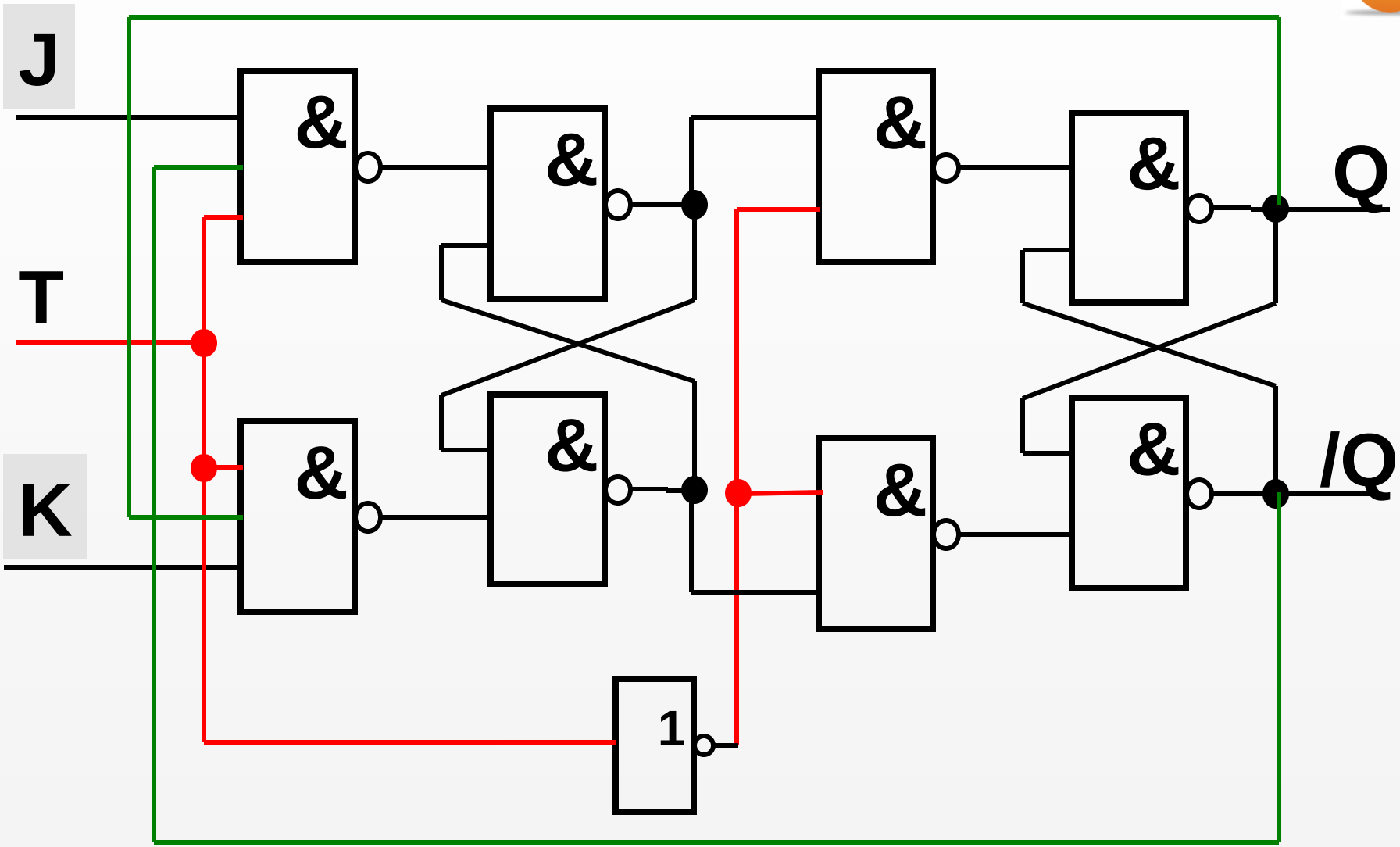




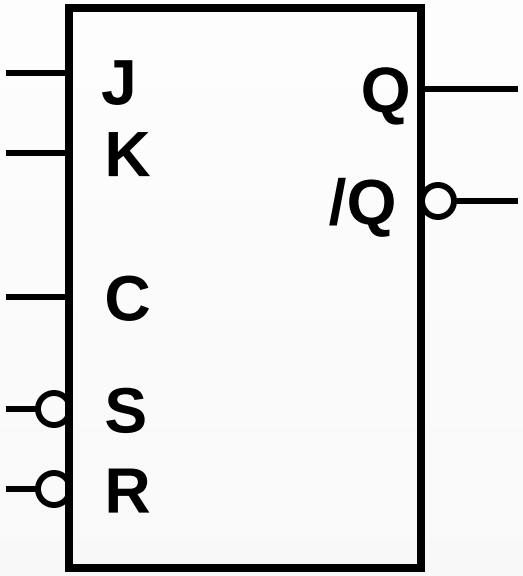
Master-Slave RS-KO (ideové schéma)



Master-Slave JK-KO (ideové schéma)



JK-klopný obvod



JK-klopný obvod

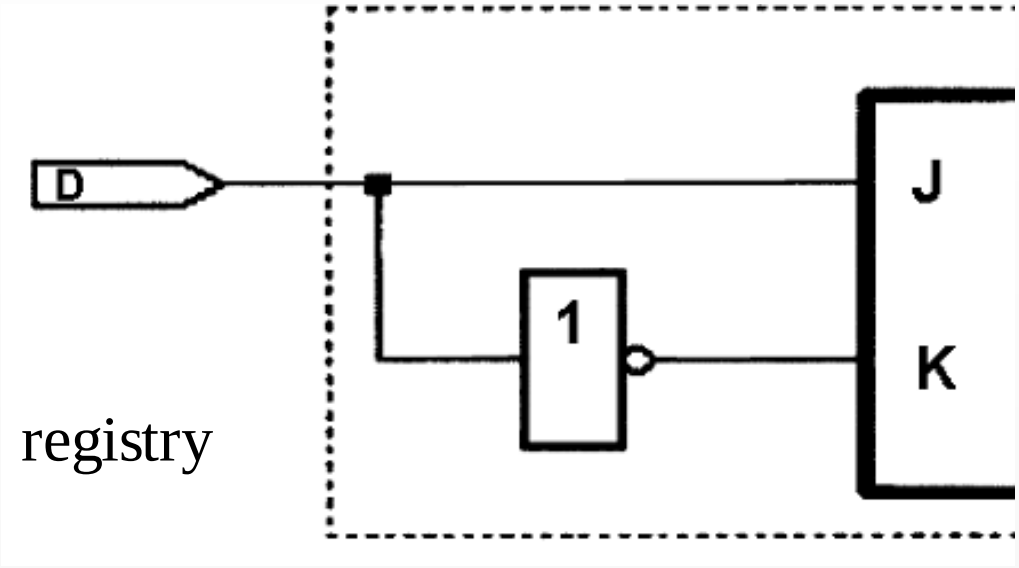
	Vstupy					Výstupy		Poznámky
	S	R	C	J	K	Q^+	$\bar{Q}^+$	
Asynchr.	0	1	x	x	x	1	0	nastavení do 1
	1	0	x	x	x	0	1	nastavení do 0
	0	0	x	x	x	1*	1*	nestabilní stav
Synchr.	1	1	⊓	0	0	Q^-	$\bar{Q}^-$	beze změny
	1	1	⊓	1	0	1	0	nastavení do 1
	1	1	⊓	0	1	0	1	nastavení do 0
	1	1	⊓	1	1	$\bar{Q}^-$	Q^-	změna stavu
	1	1	0	x	x	Q	$\bar{Q}$	beze změny

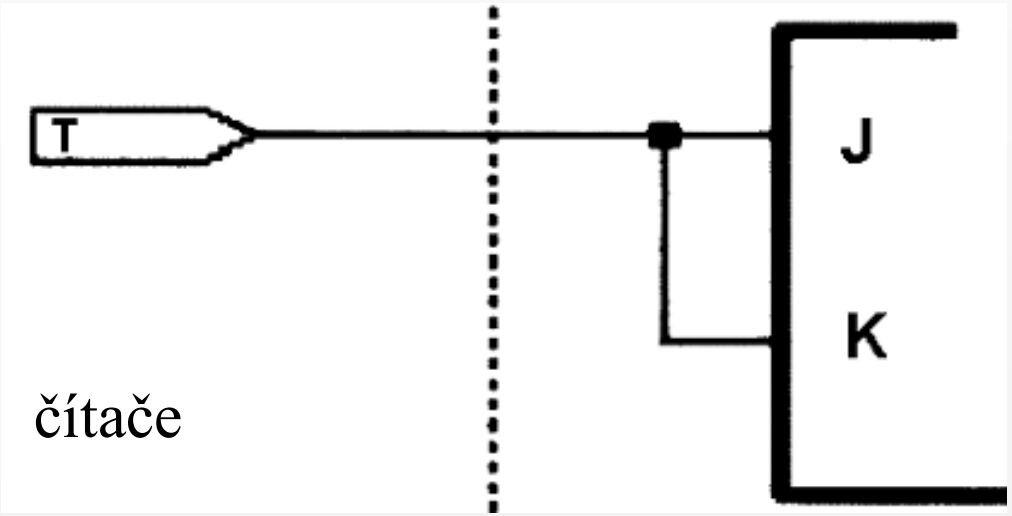
* Nestabilní stavy (obdobně jako u dvoustupňového klopného obvodu RS)

Úpravy JK-KO

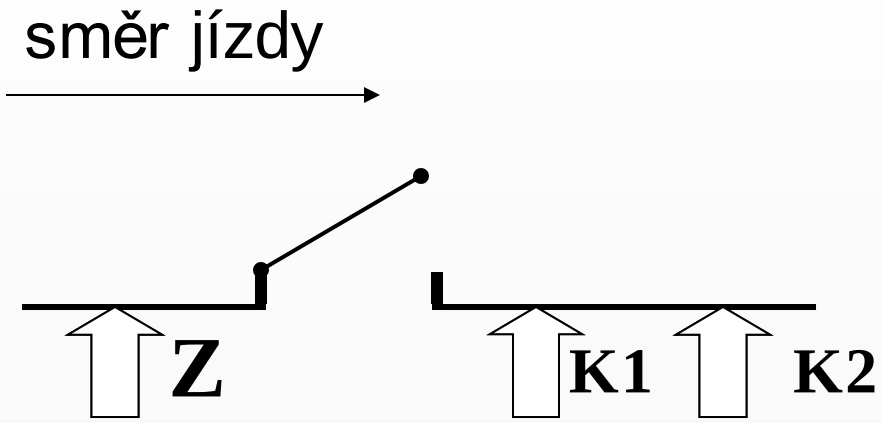
Realizace **D-KO** a **T-KO** pomocí **JK-KO**

J	K	Q
0	0	Q_{n-1}
0	1	**0**
1	0	**1**
1	1	$/Q_{n-1}$



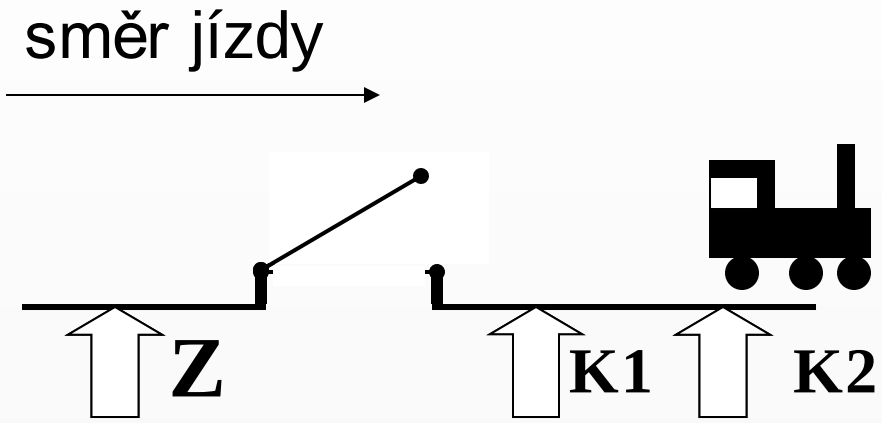


Příklad sekvenční úlohy



Přejede-li lokomotiva bod Z, závory se spustí, je-li poslední vagón za bodem K1, závory se zvednou. Sestavte obvod, který ovládá signál pro spuštění závor.

Příklad sekvenční úlohy



Přejede-li lokomotiva bod Z, závory se spustí, je-li poslední vagón za bodem K1, závory se zvednou. Sestavte obvod, který ovládá signál pro spuštění závor.

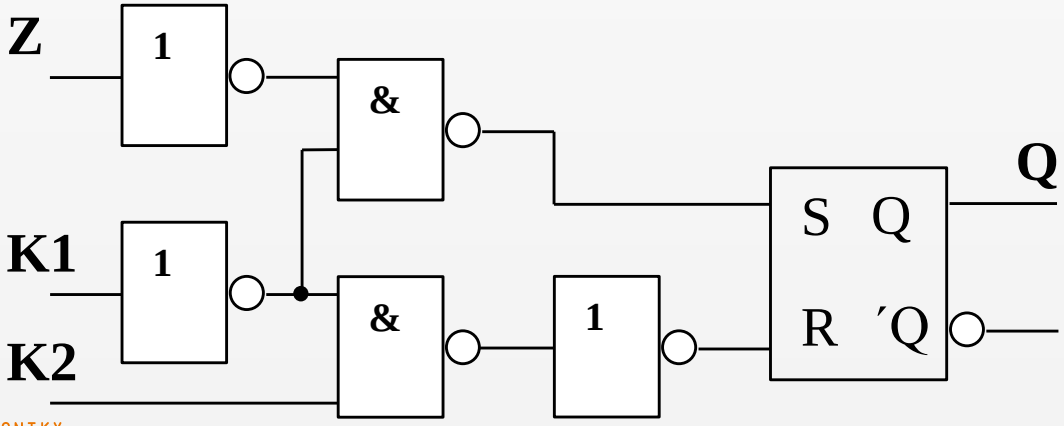
Z	K1	K2	Qt	Qt+1	Funkce RS
0	0	0	0	0	Pamatování
0	0	0	1	1	Pamatování
0	0	1	X	0	Nulování
0	1	X	X	1	Nastavení
1	X	X	X	1	Nastavení

Příklad – pokr.

vstup S (Spuštění závor) - log. 1 v případě, že Z nebo K1 jsou rovny 1. (S = Z +K1 …).

vstup R (Zvednutí závor) - log. 1 v případě, že platí současně, že K1 = 0 a K2 = 1.

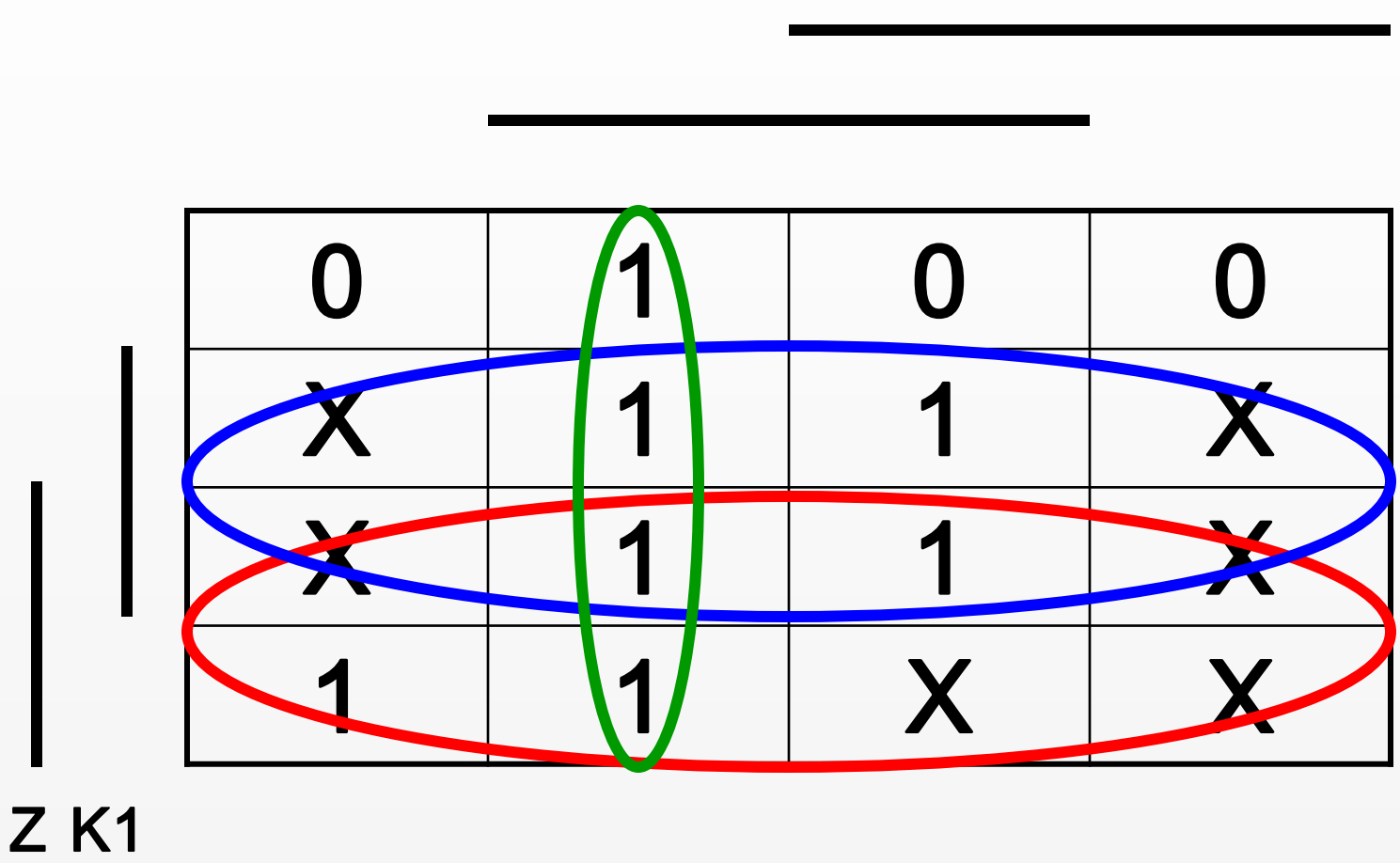
Případ, kdy současně oba vstupy S i R jsou rovny jedné nemůže nastat. Jestliže jsou všechny vstupní signály nulové, na S i R vstupy přivádíme 0. Při této hodnotě vstupů setrvávají závory v poloze dané předchozím nastavením



Metoda TF

Z	K1	K2	Q_{-1}	Q	Error
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	X	1
0	1	0	1	1	
0	1	1	0	X	1
0	1	1	1	1	
1	0	0	0	1	
1	0	0	1	1	
1	0	1	0	X	1
1	0	1	1	X	1
1	1	0	0	X	
1	1	0	1	1	
1	1	1	0	X	1
1	1	1	1	1	

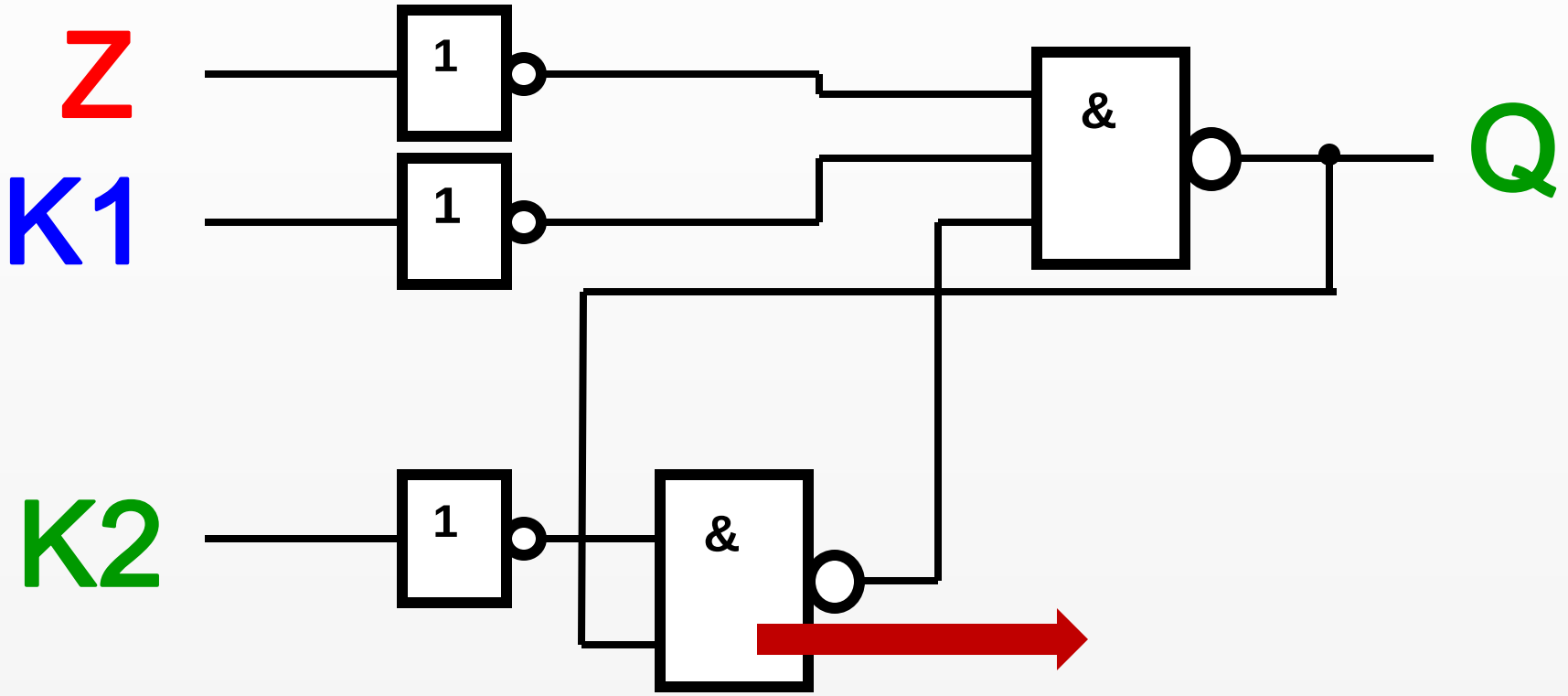
Příklad



Q
0
1
0
0
X
1
X
1
1
1
X
X
X
1
X
1

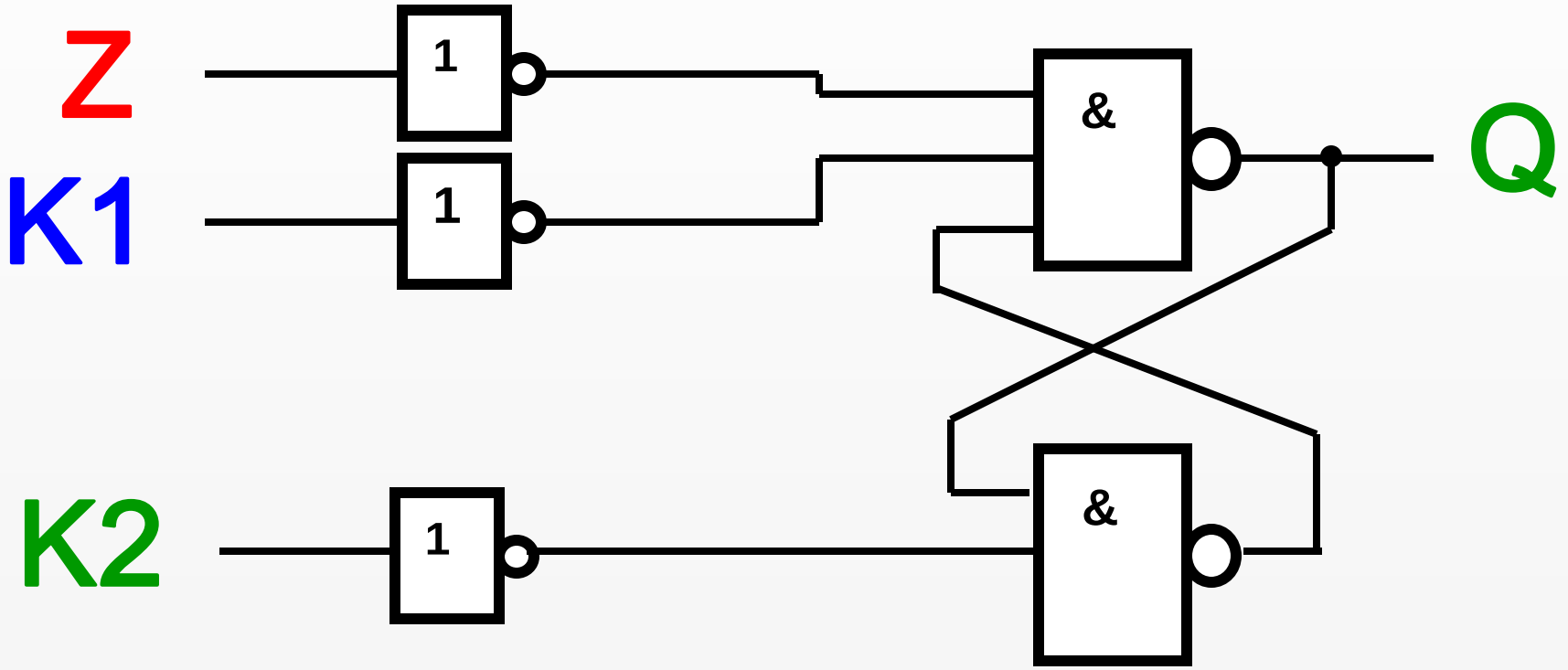
$$Q = Z + K1 + /K2 \bullet Q_{-1}$$

Příklad

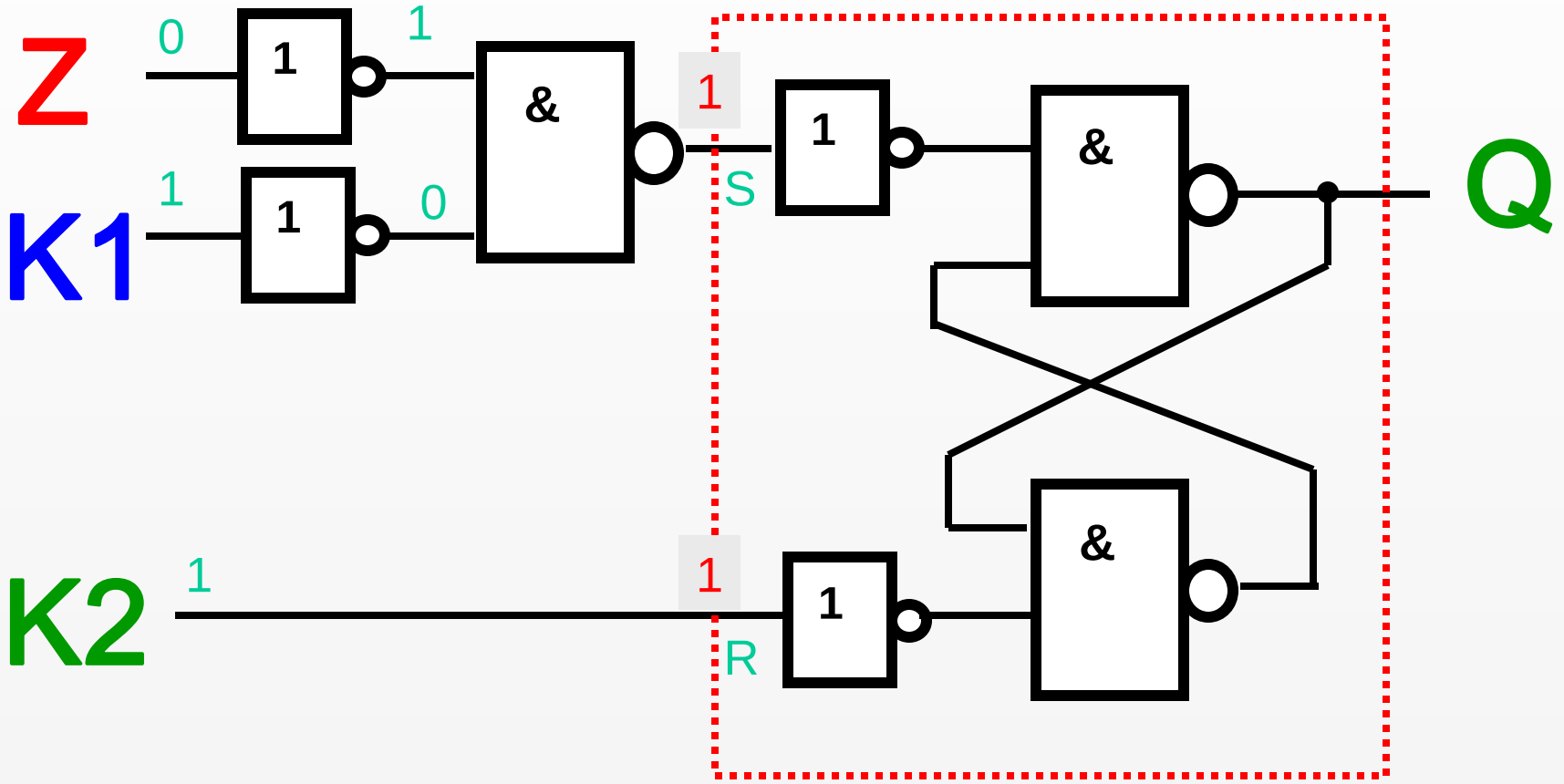


$$Q = Z + K1 + /K2 \bullet Q_{-1}$$

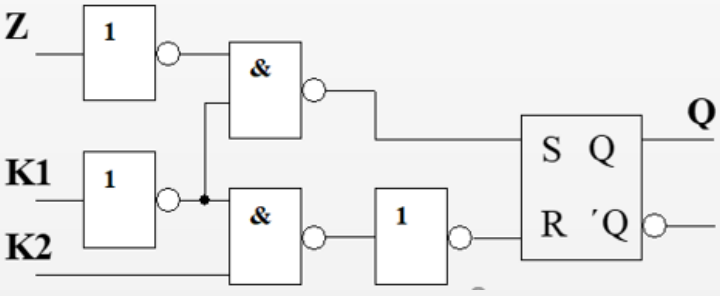
Příklad



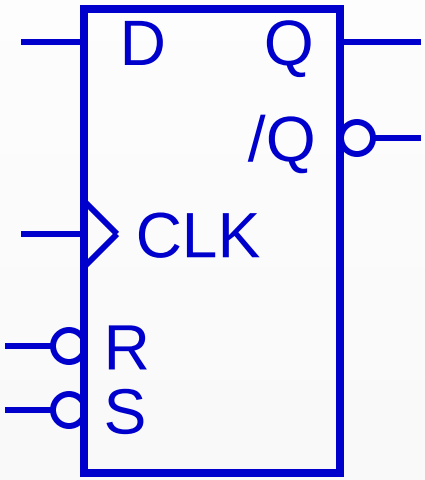
Příklad

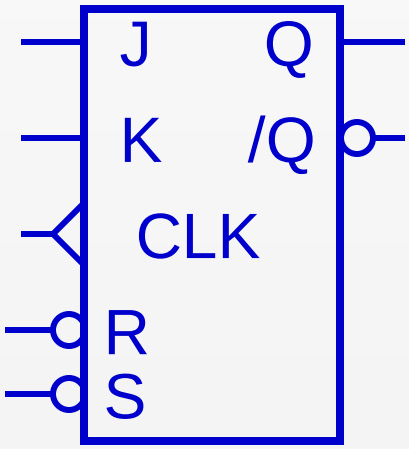


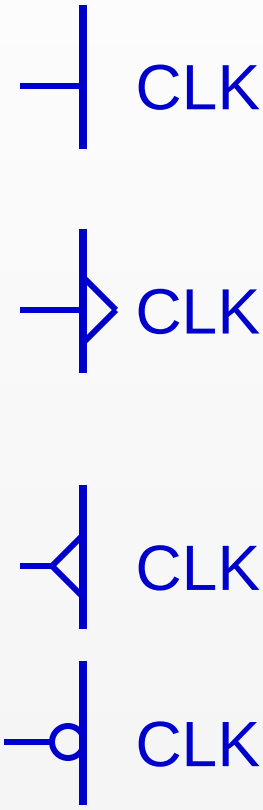
Příklad



Integrované obvody





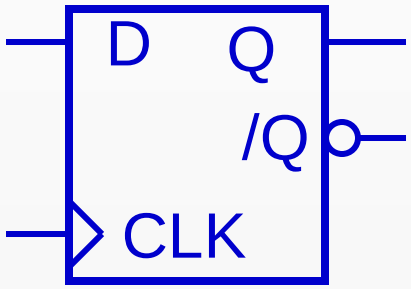


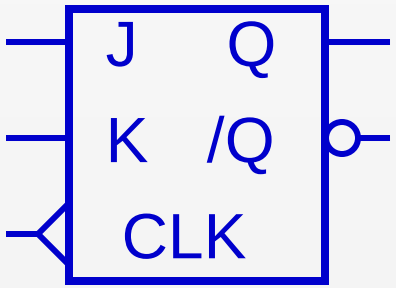
Vzestupná
Náběžná

Sestupná
Závěrná

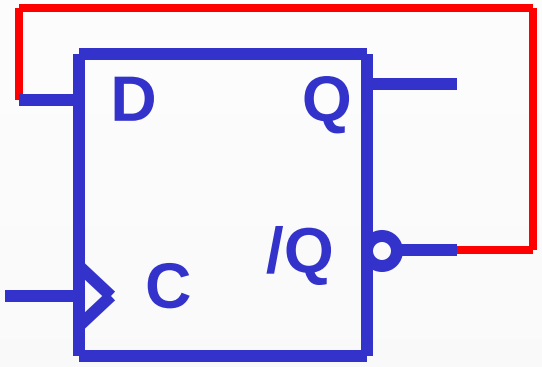
Čítače (asynchronní)

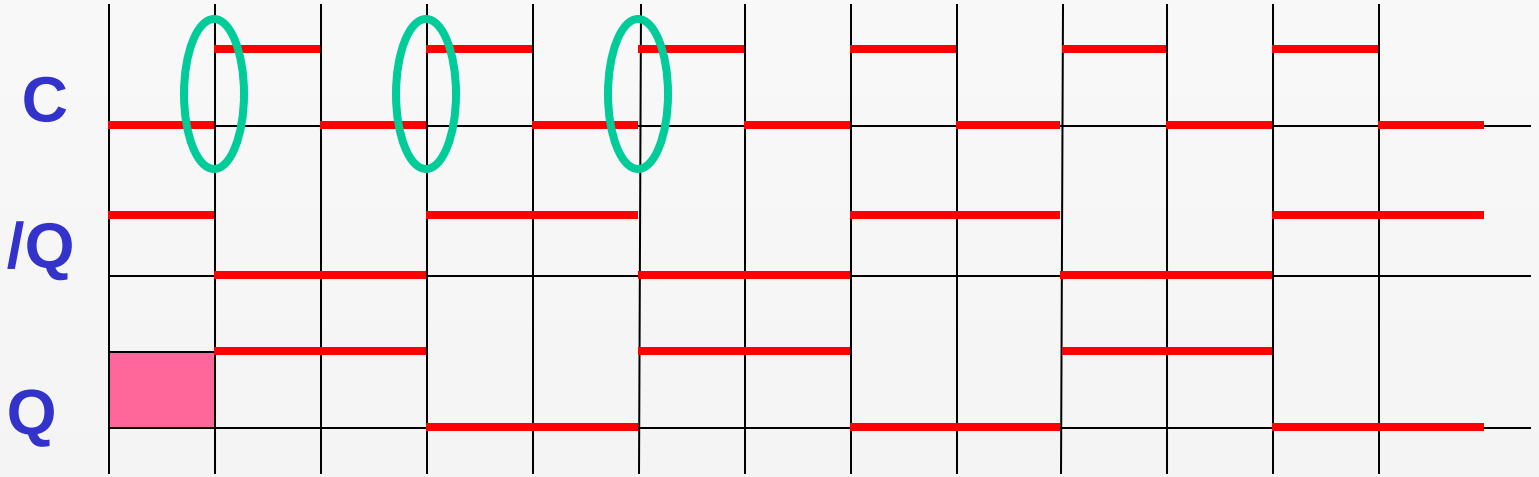
0	0	0	0
1.	0	0	1
2.	0	1	0
3.	0	1	1
4.	1	0	0
5.	1	0	1
6.	1	1	0
7.	1	1	1
atd.	Q3	Q2	Q1





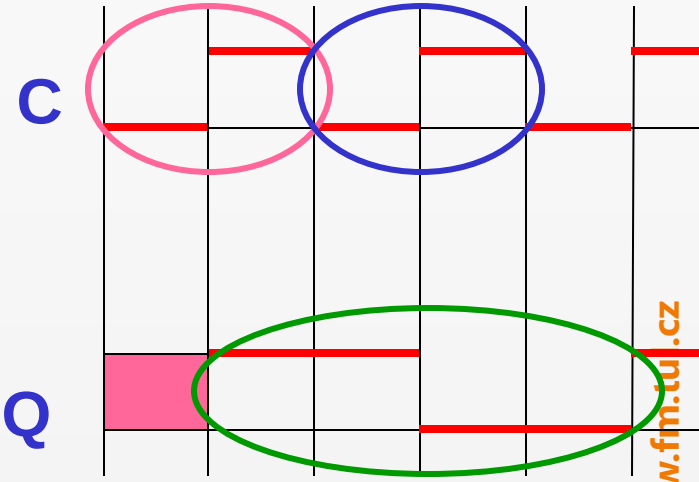
Čítače (asynchronní)



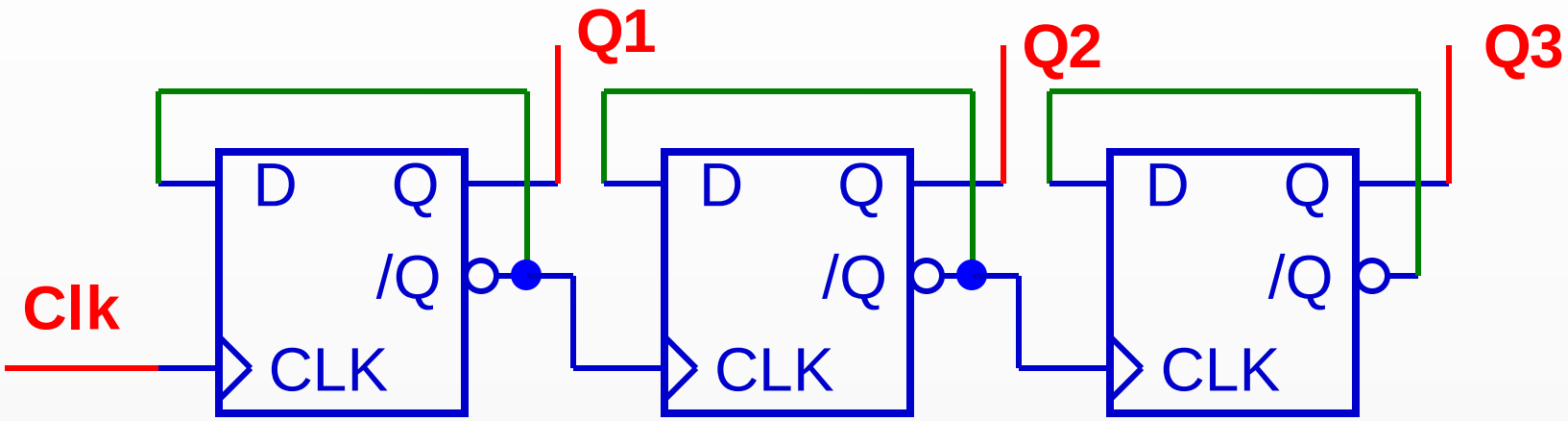


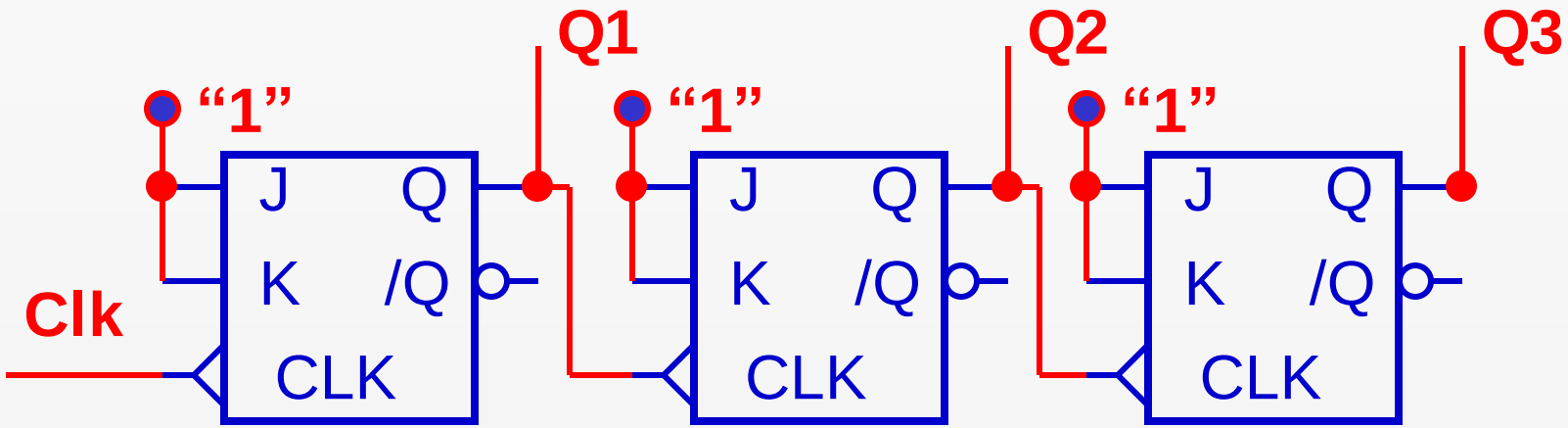
Čítače (asynchronní)

0	0	0	0
1.	0	0	1
2.	0	1	0
3.	0	1	1
4.	1	0	0
5.	1	0	1
6.	1	1	0
7.	1	1	1
atd.	Q3	Q2	Q1



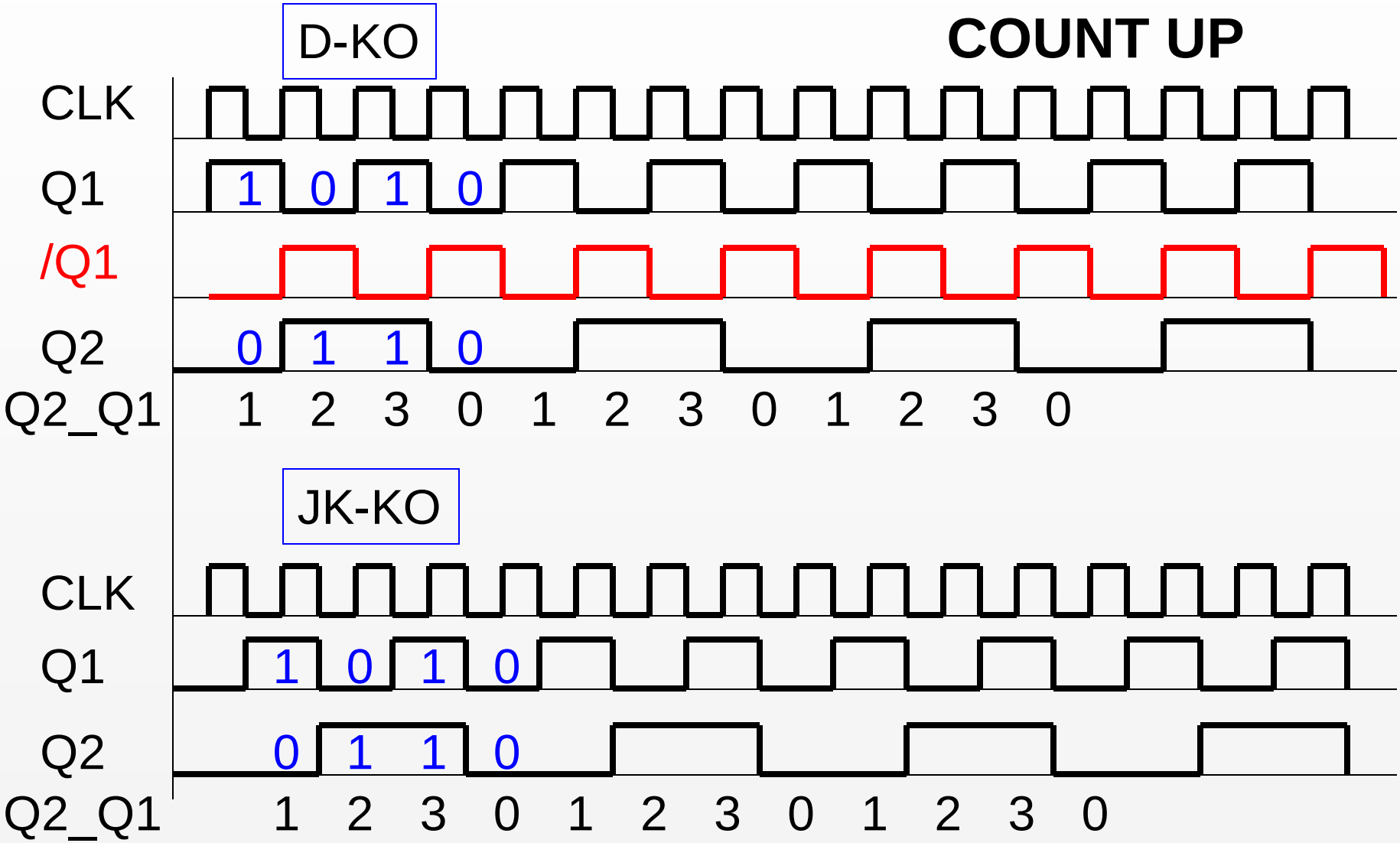
Asynchronní čítače



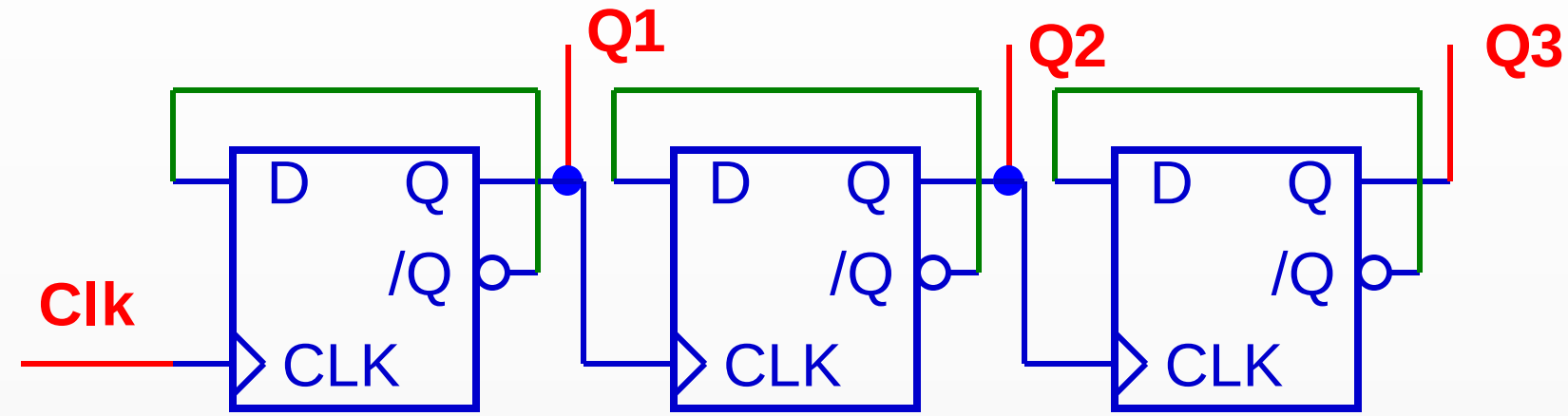


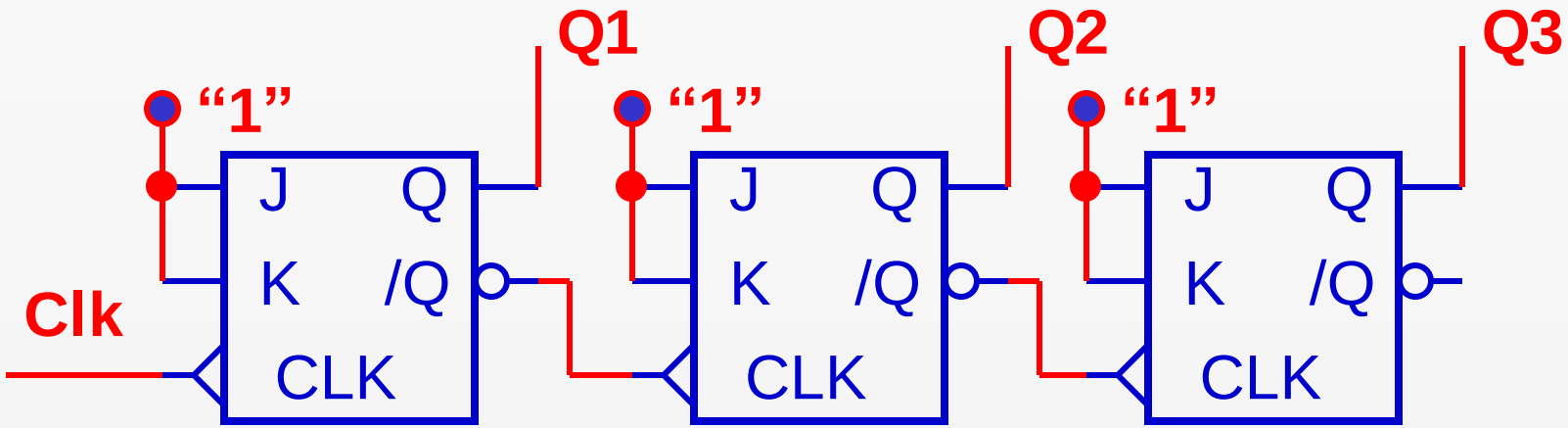
COUNT UP

Asynchronní čítače



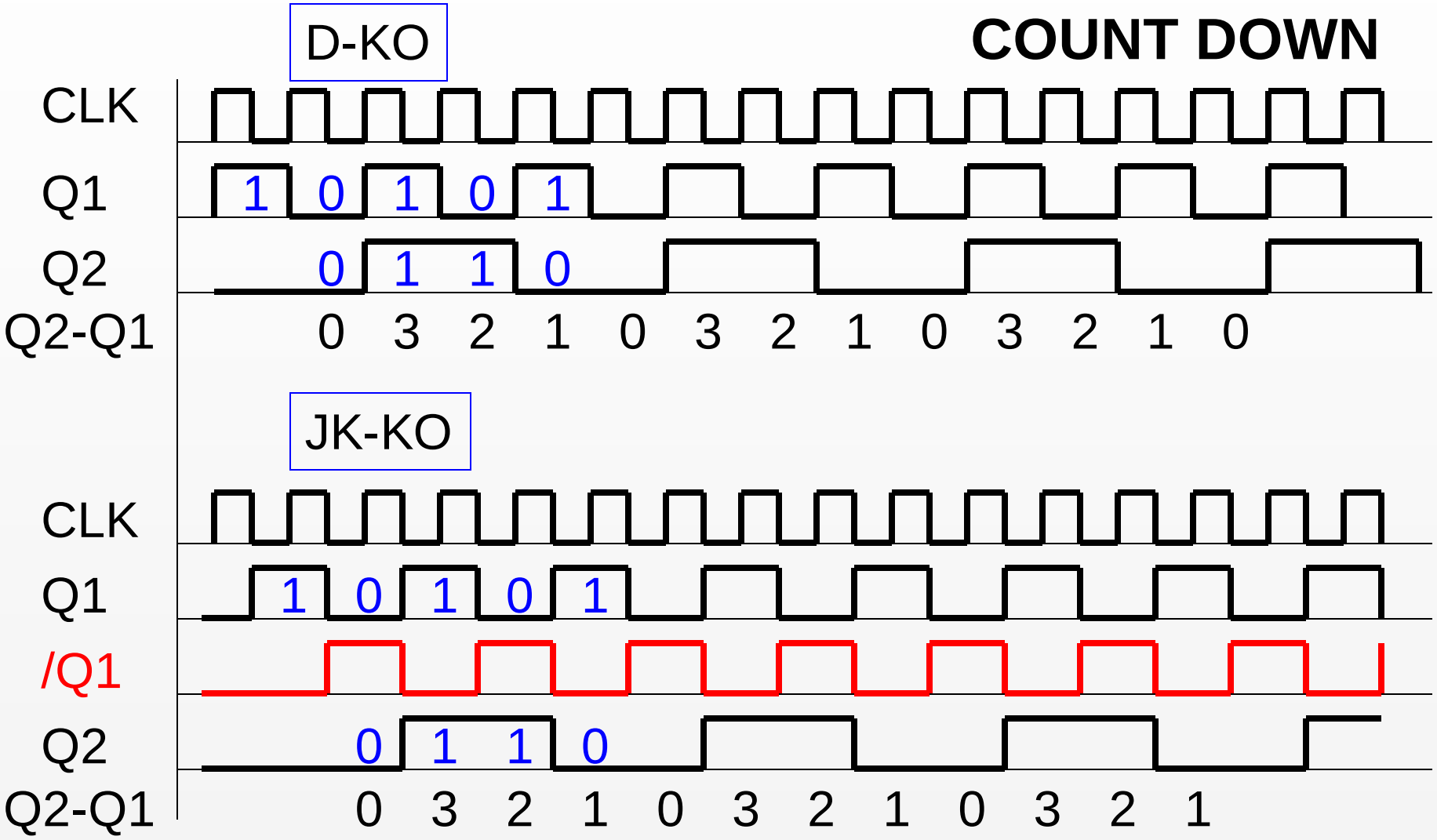
Asynchronní čítače



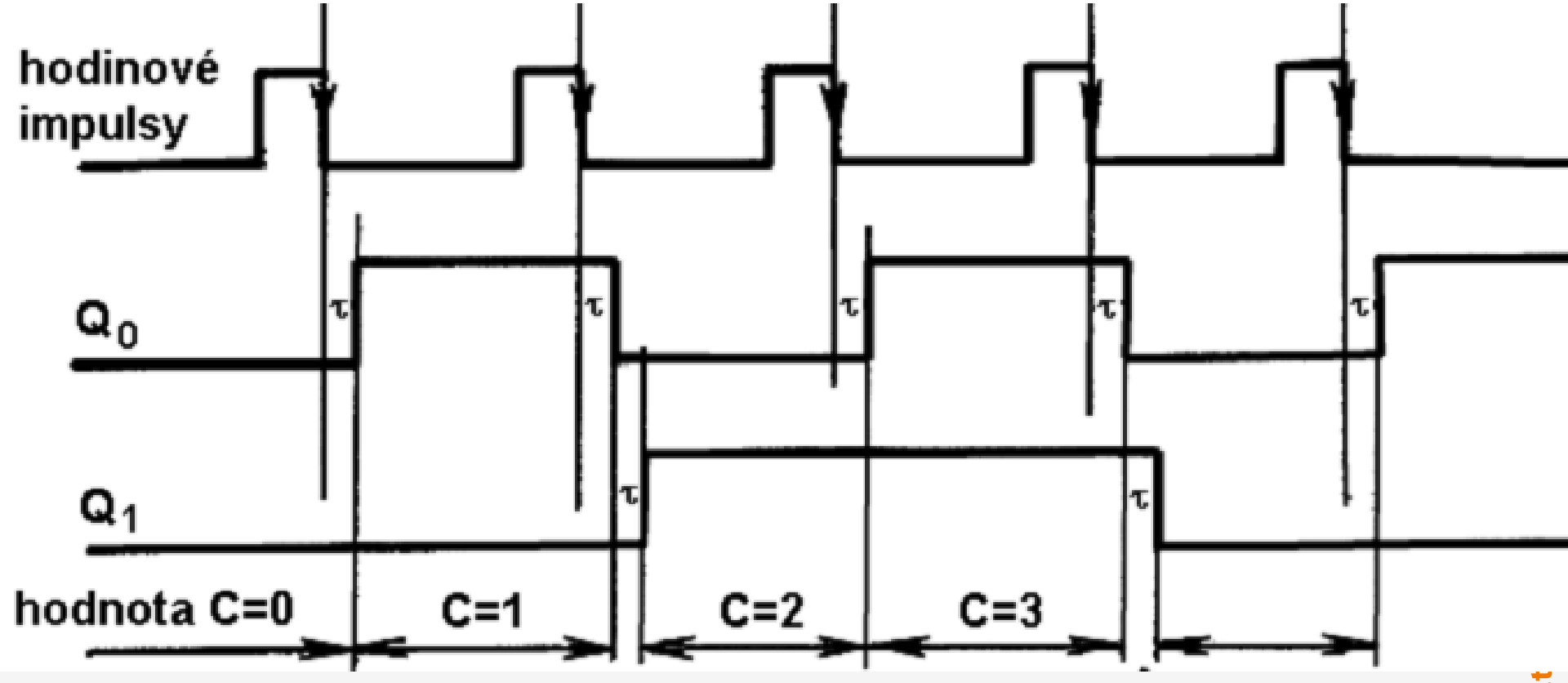


COUNT DOWN

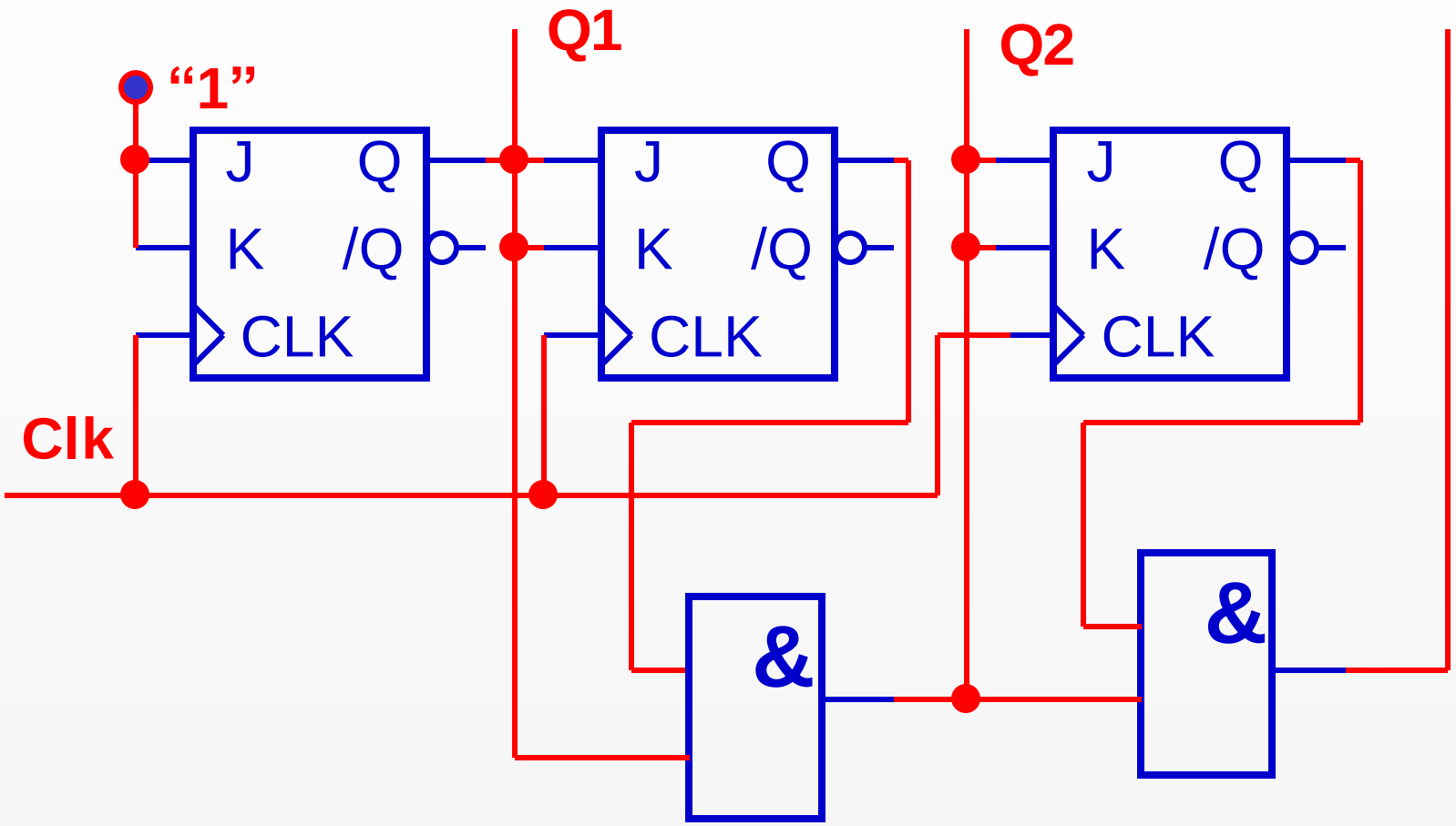
Asynchronní čítače



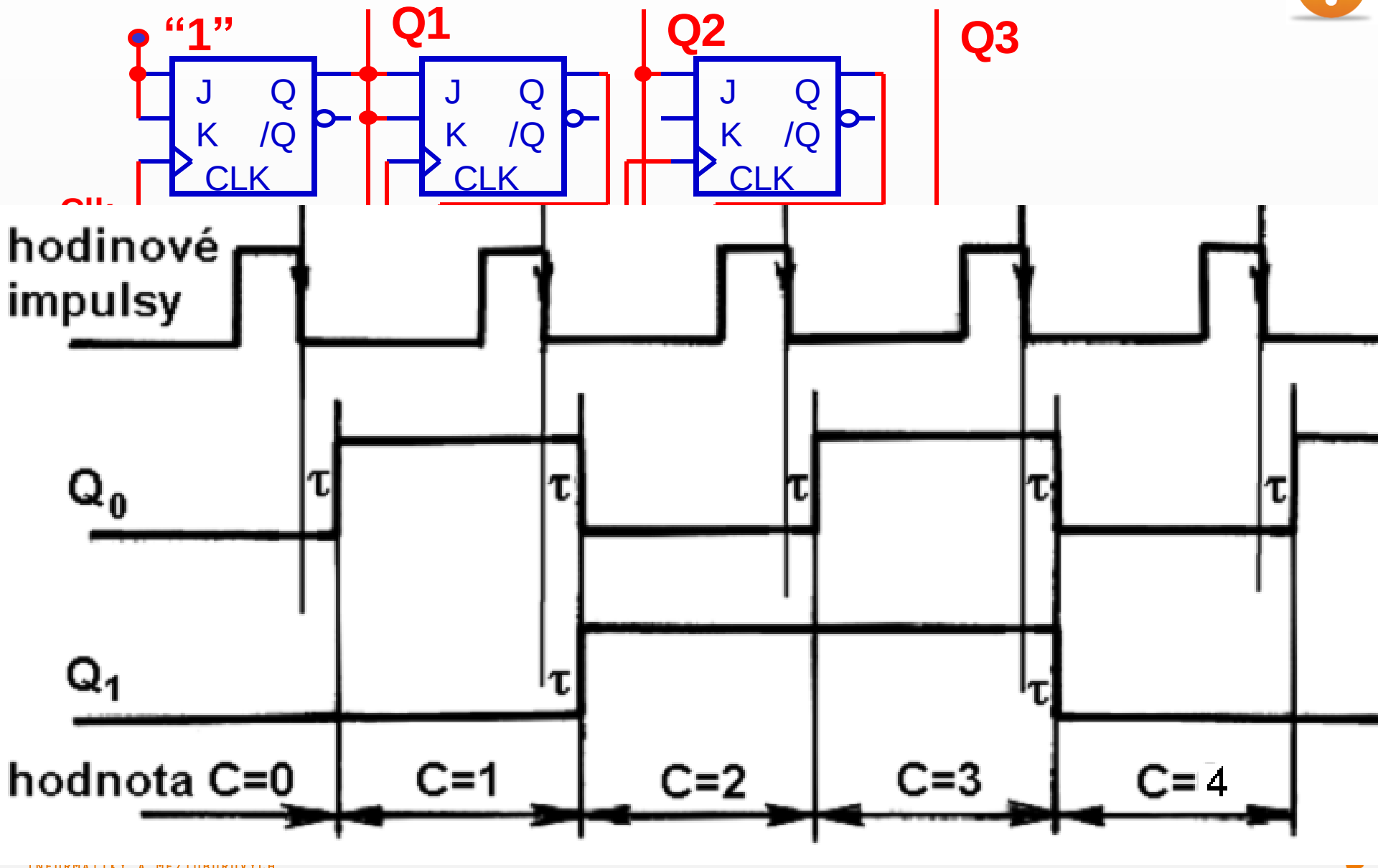
Asynchronní čítače



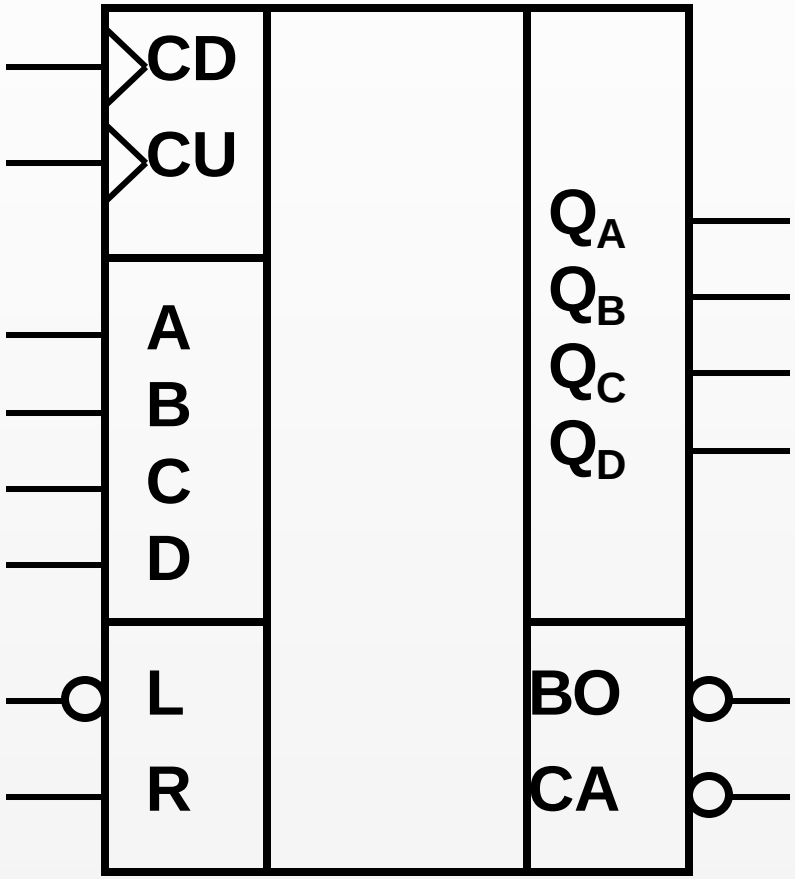
Synchronní čítače



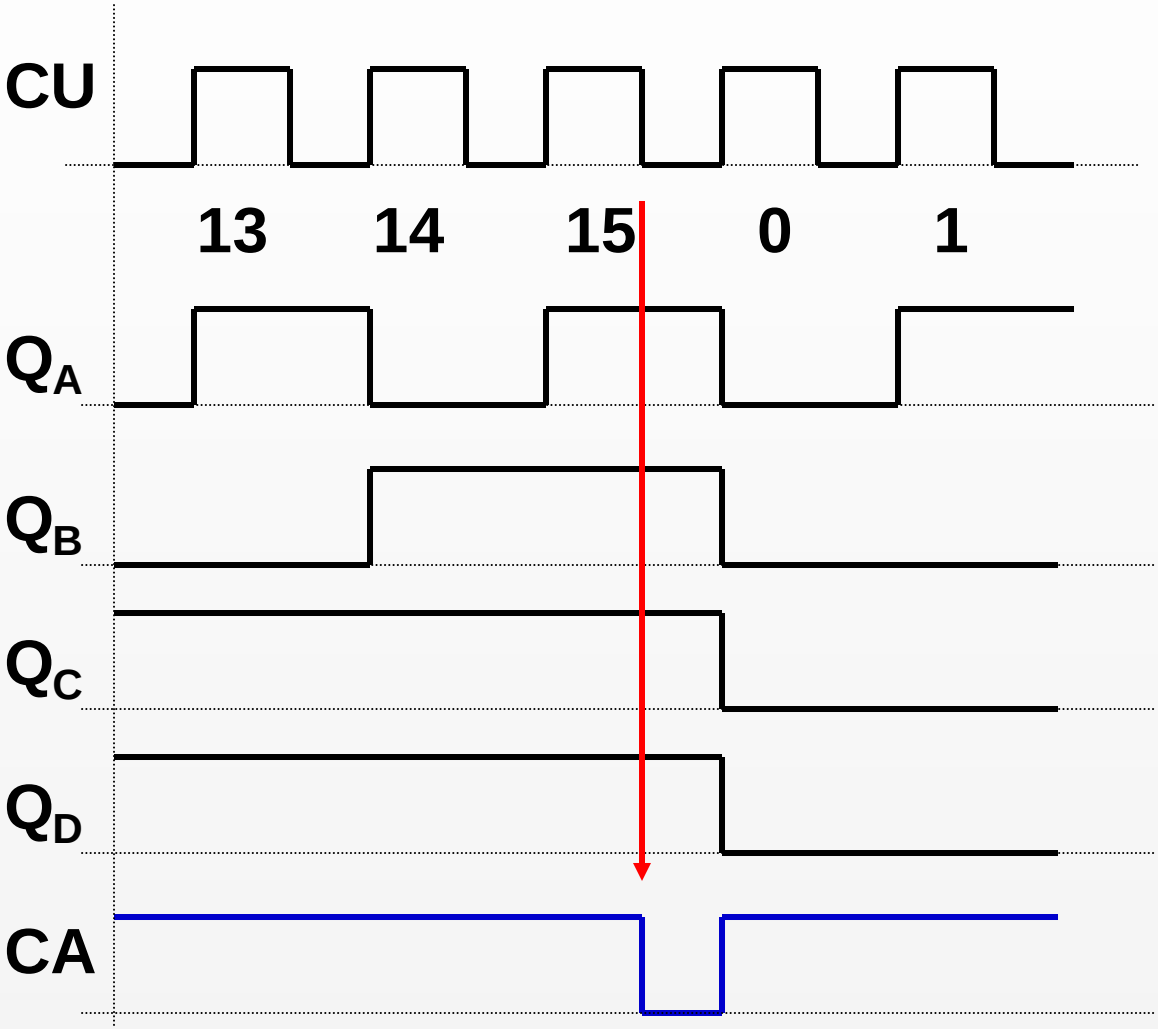
Synchronní čítače

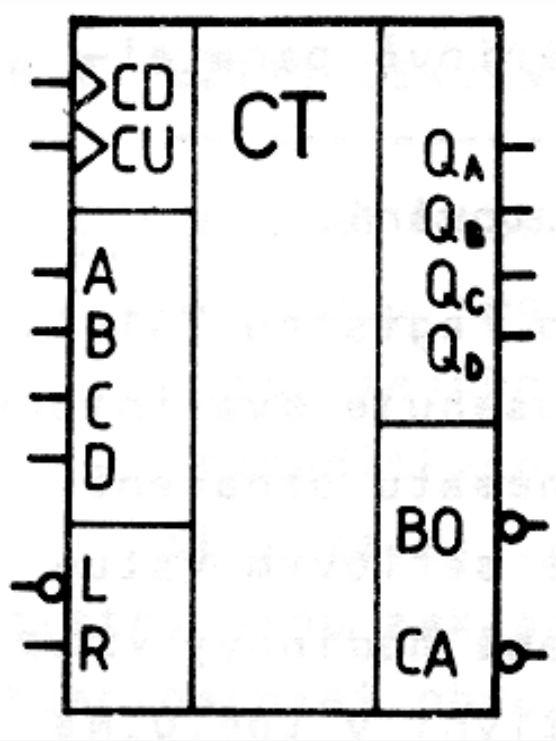


Integrovaný čítač



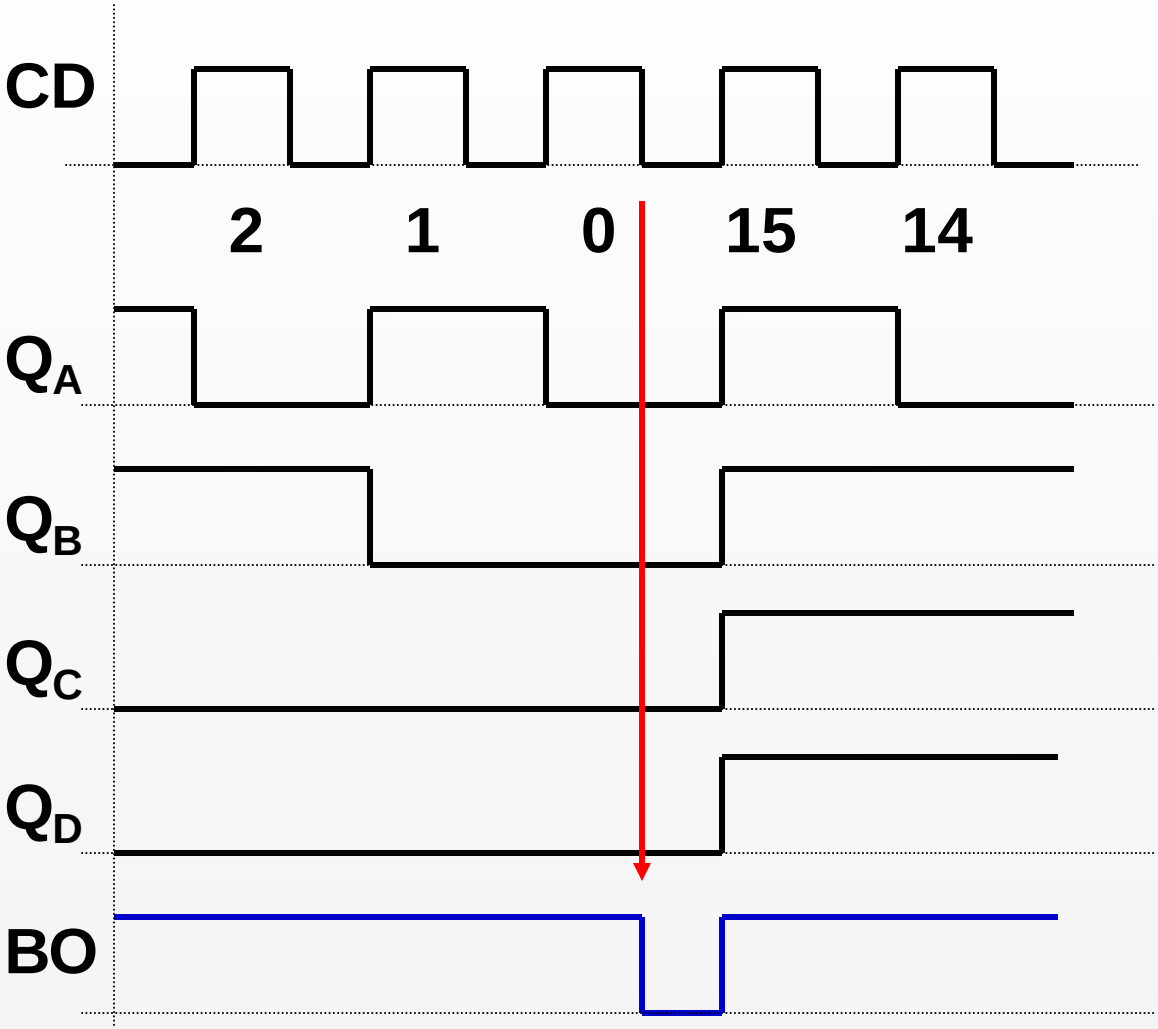
Časový diagram čítače

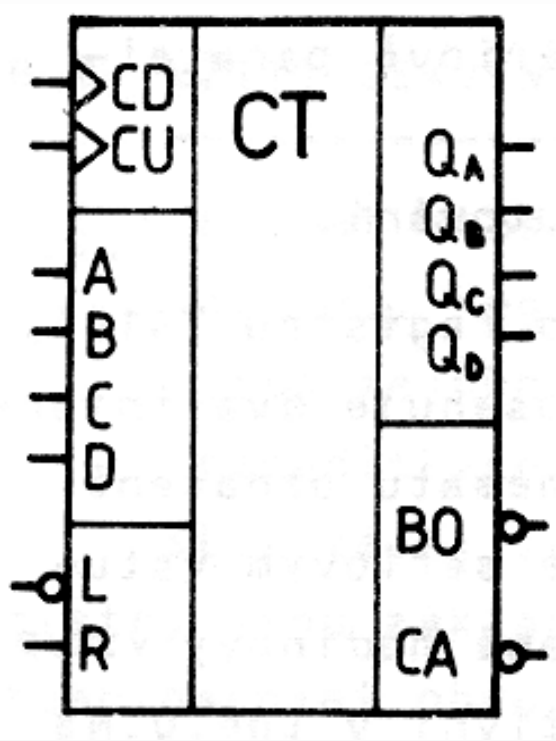




Count UP

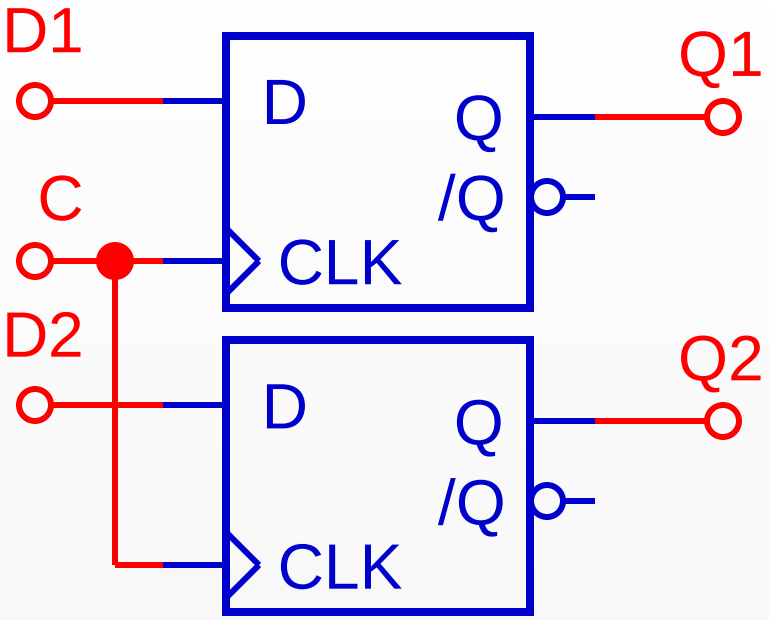
Časový diagram čítače



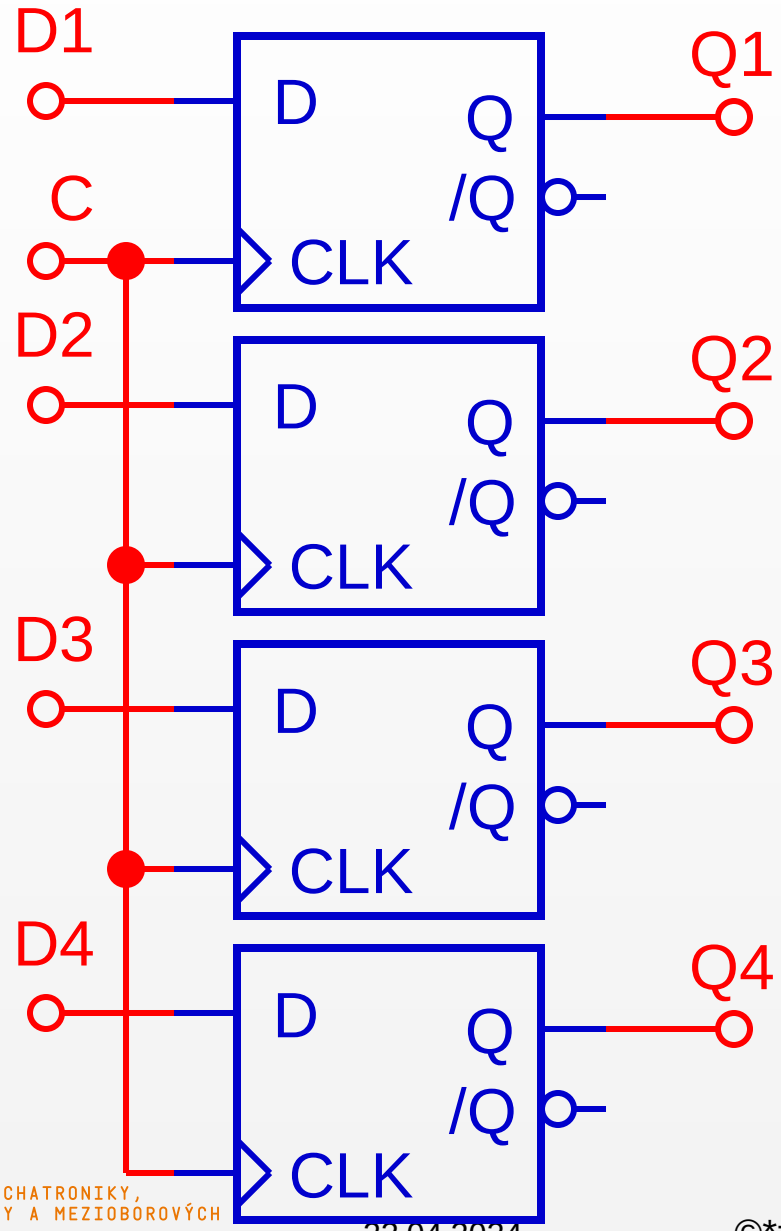


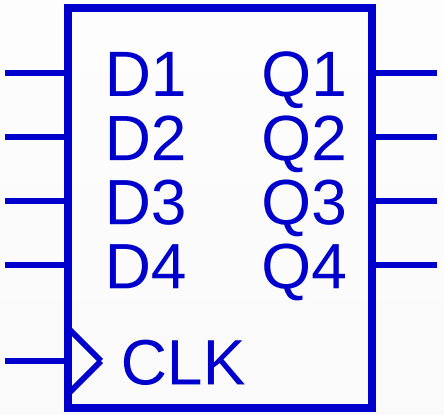
Count DOWN

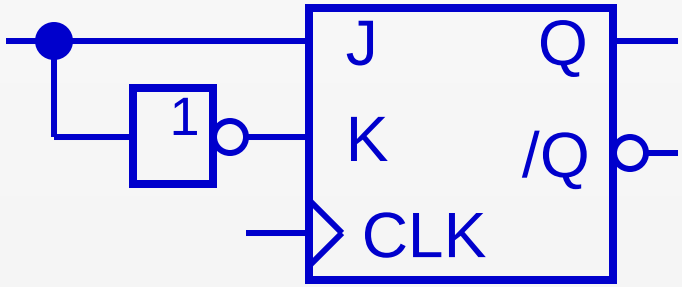
Registr



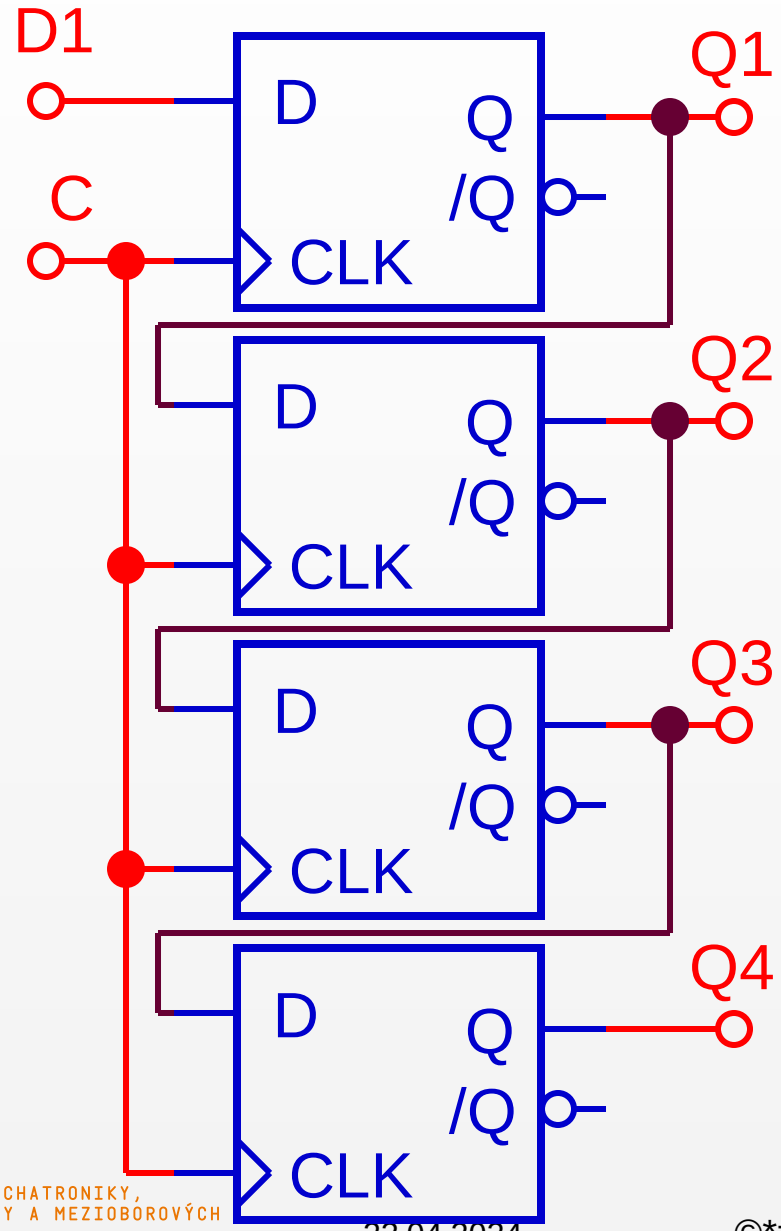
Registr

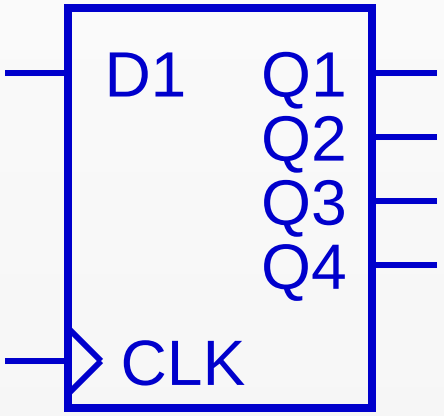

D1
C
D2
D3
D4
D
Q
/Q
CLK
Q1
Q2
Q3
Q4


D1
D2
D3
D4
Q1
Q2
Q3
Q4
CLK

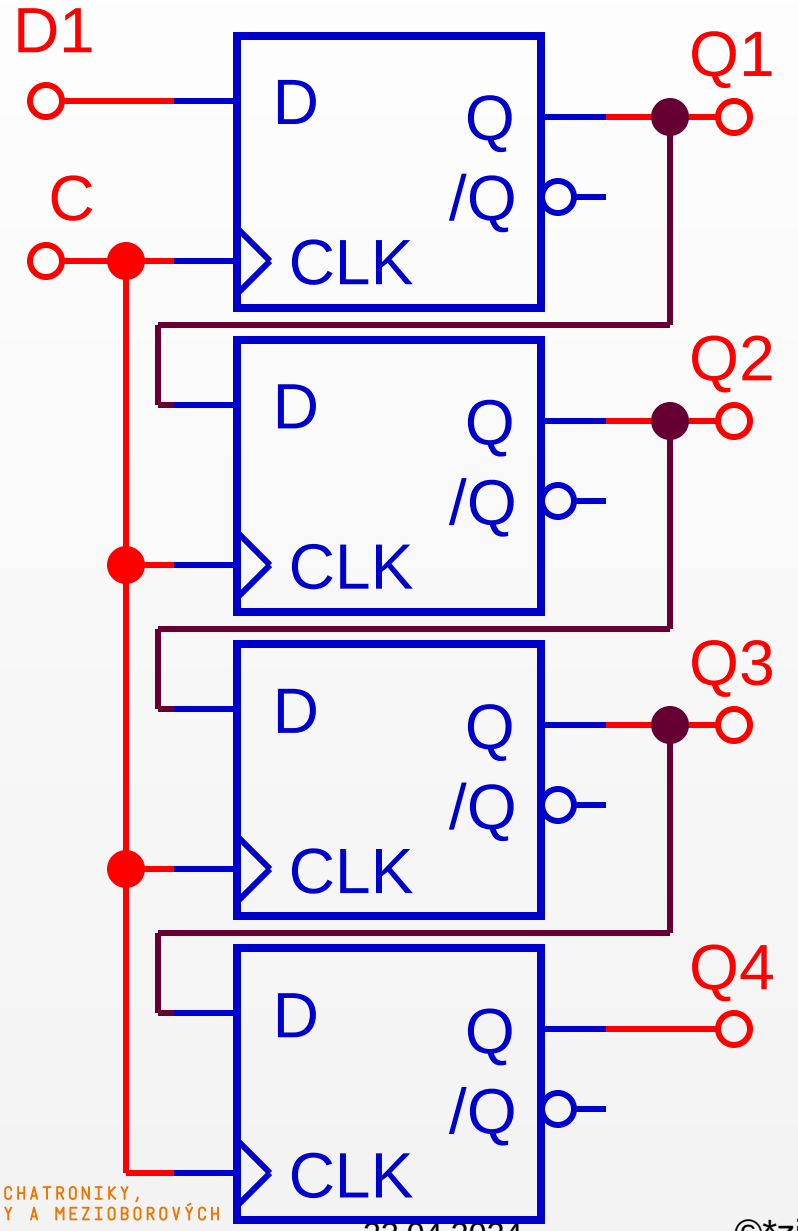

1
J
K
Q
/Q
CLK

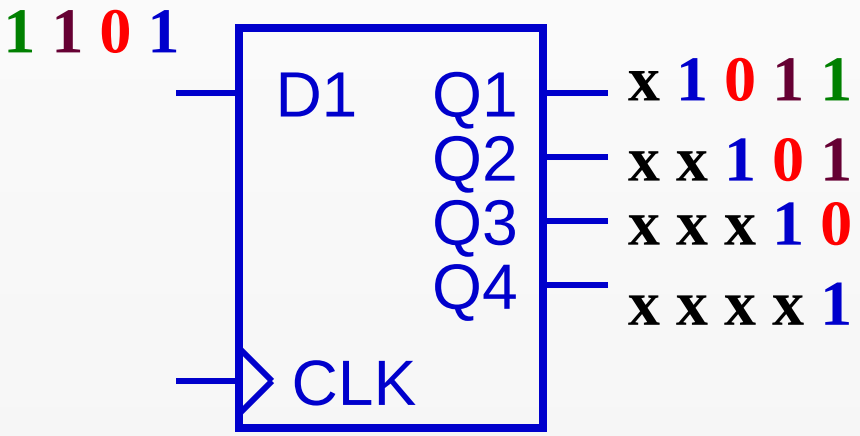
Posuvný registr



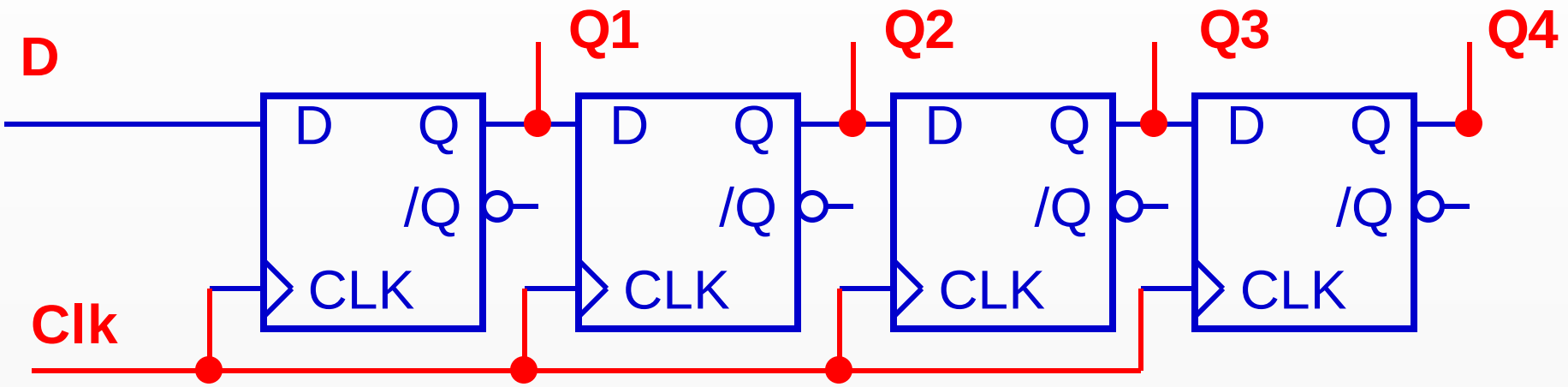


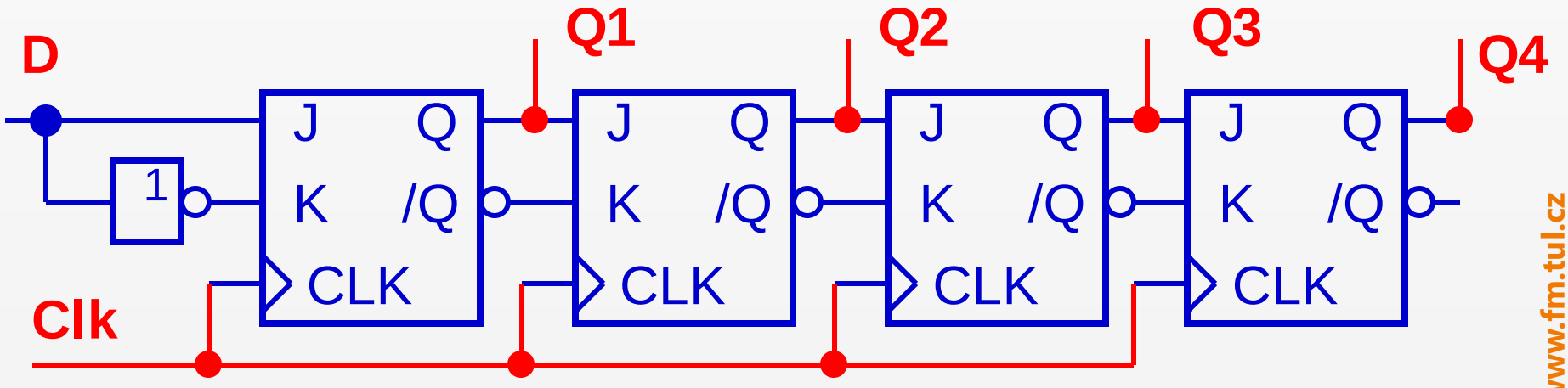
Posuvný registr



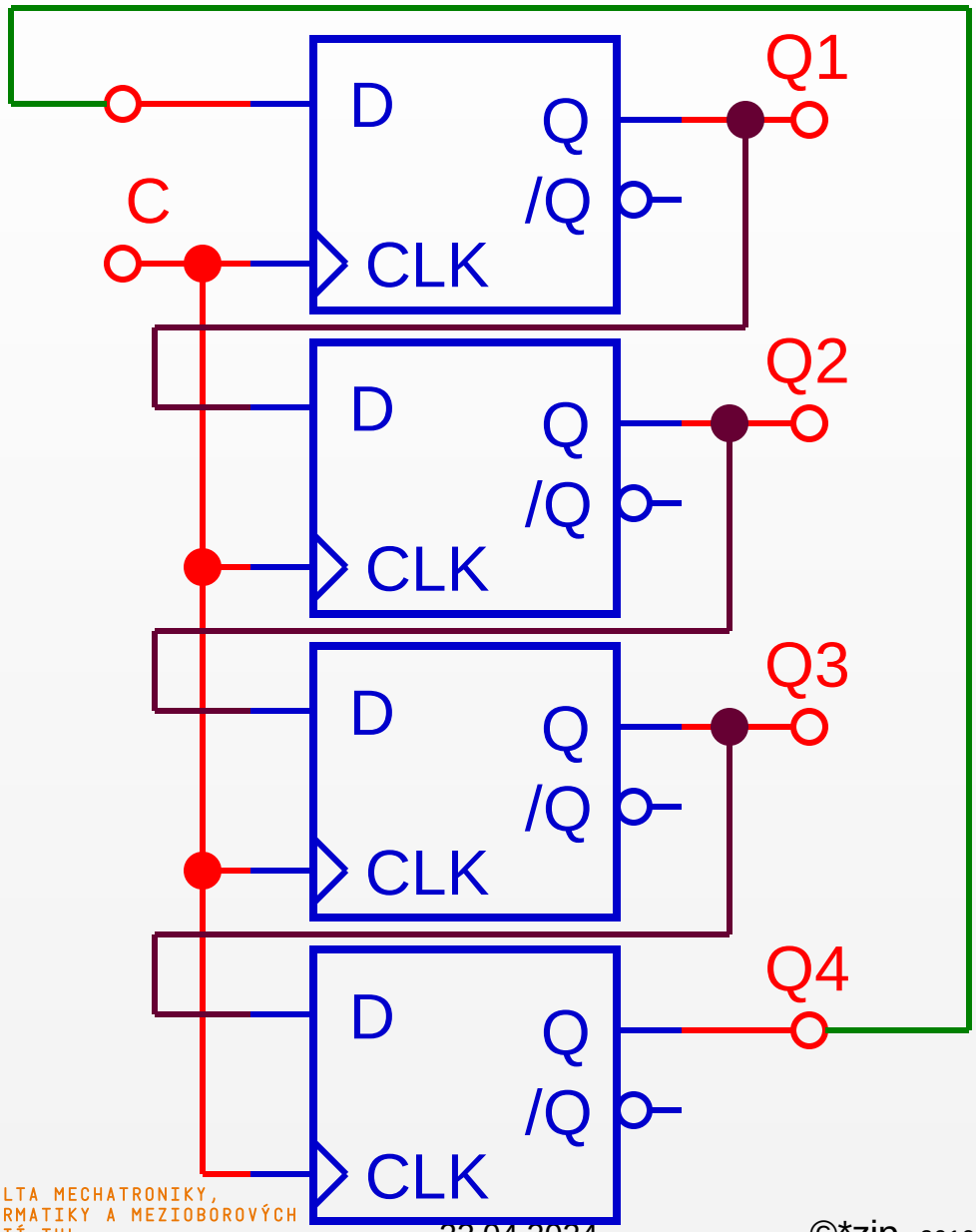


Posuvný registr

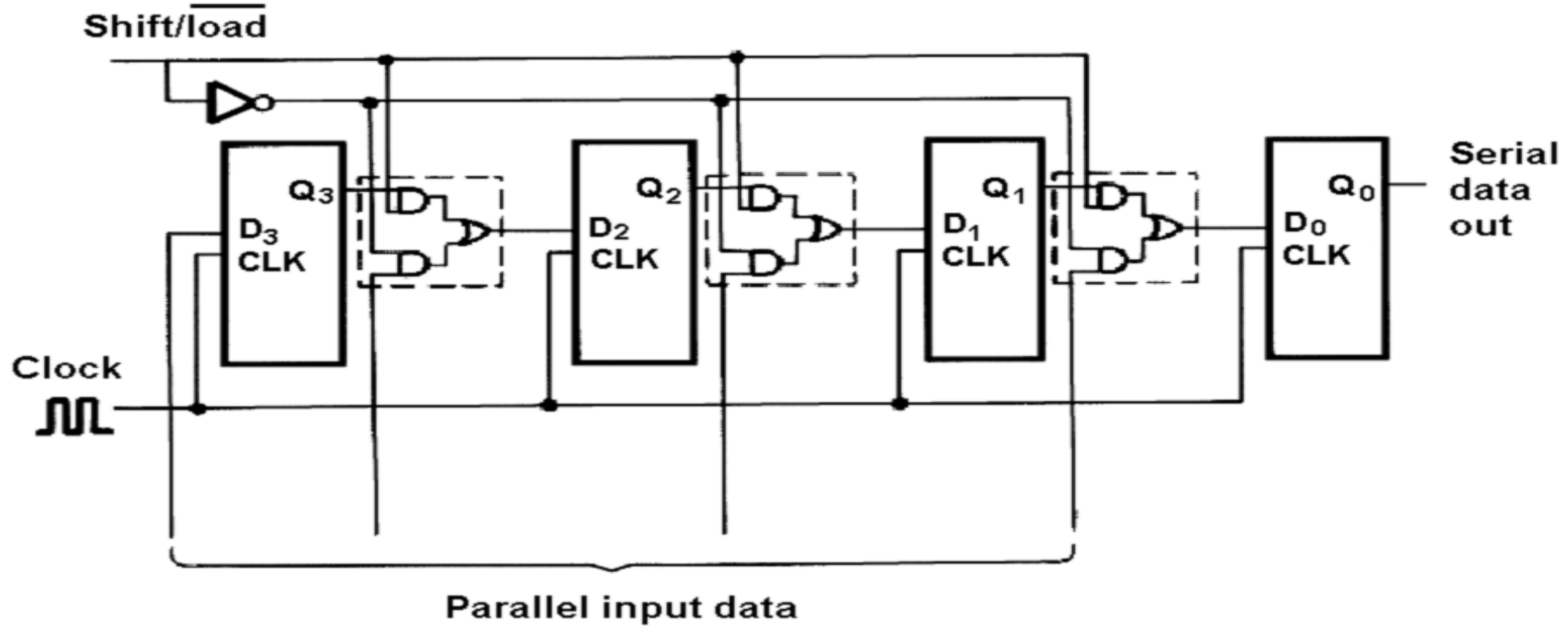




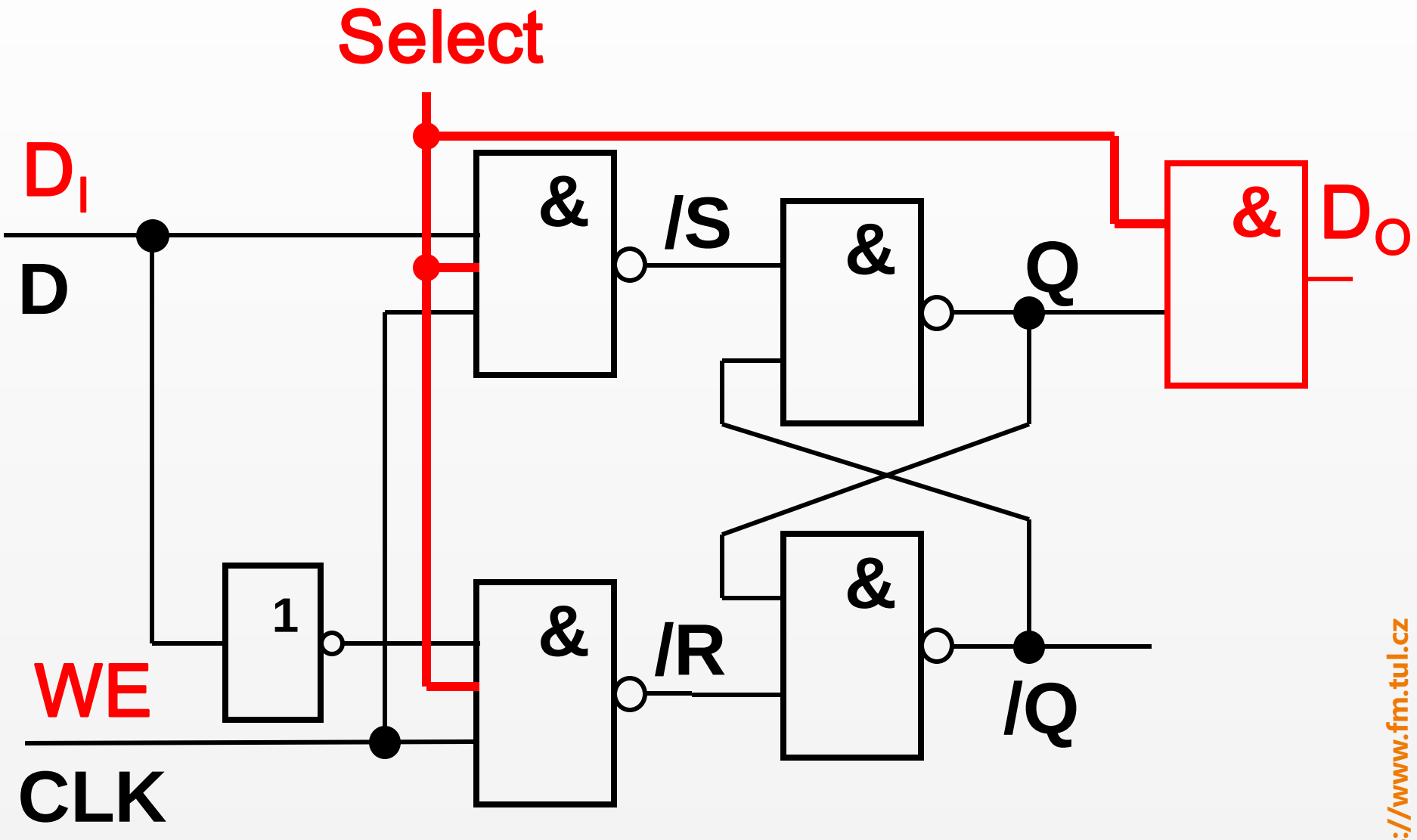
Posuvný registr



Posuvný registr



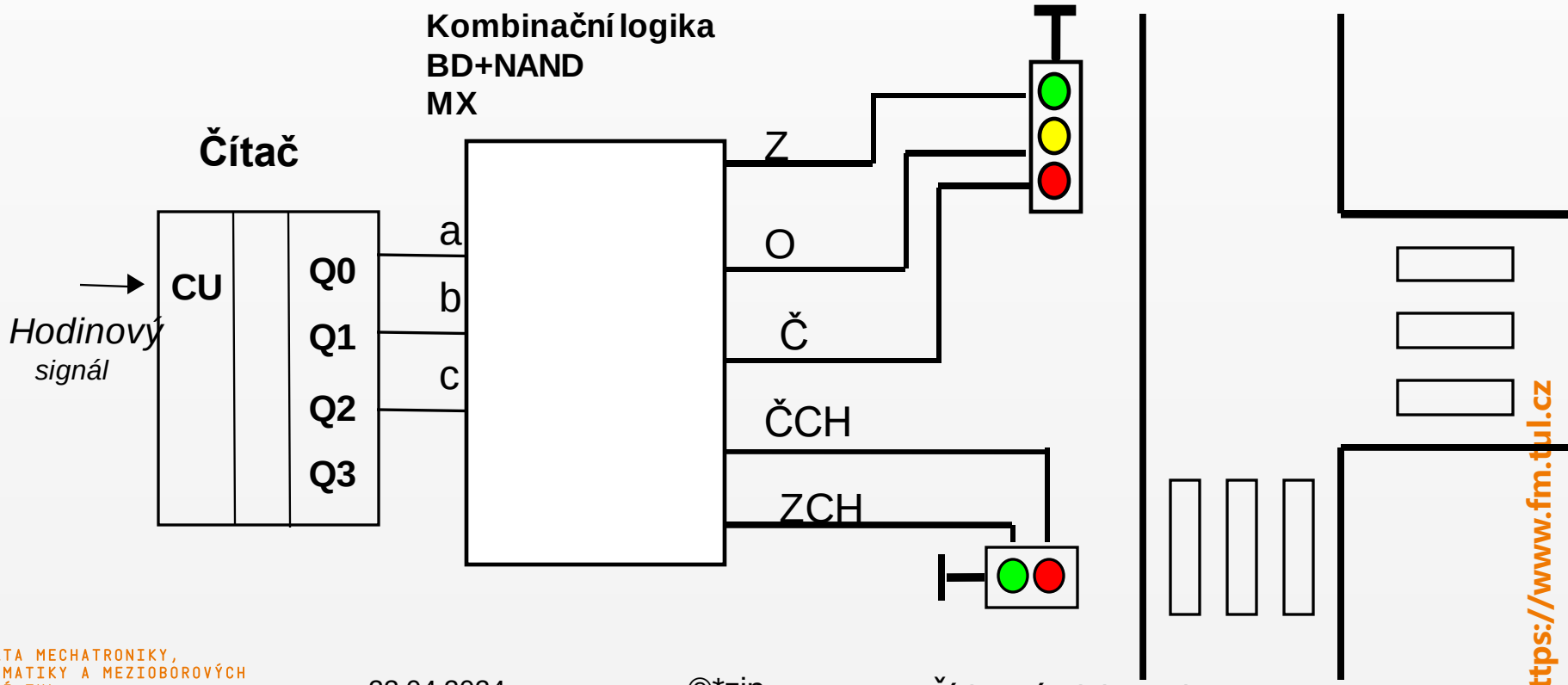
Statická paměťová buňka



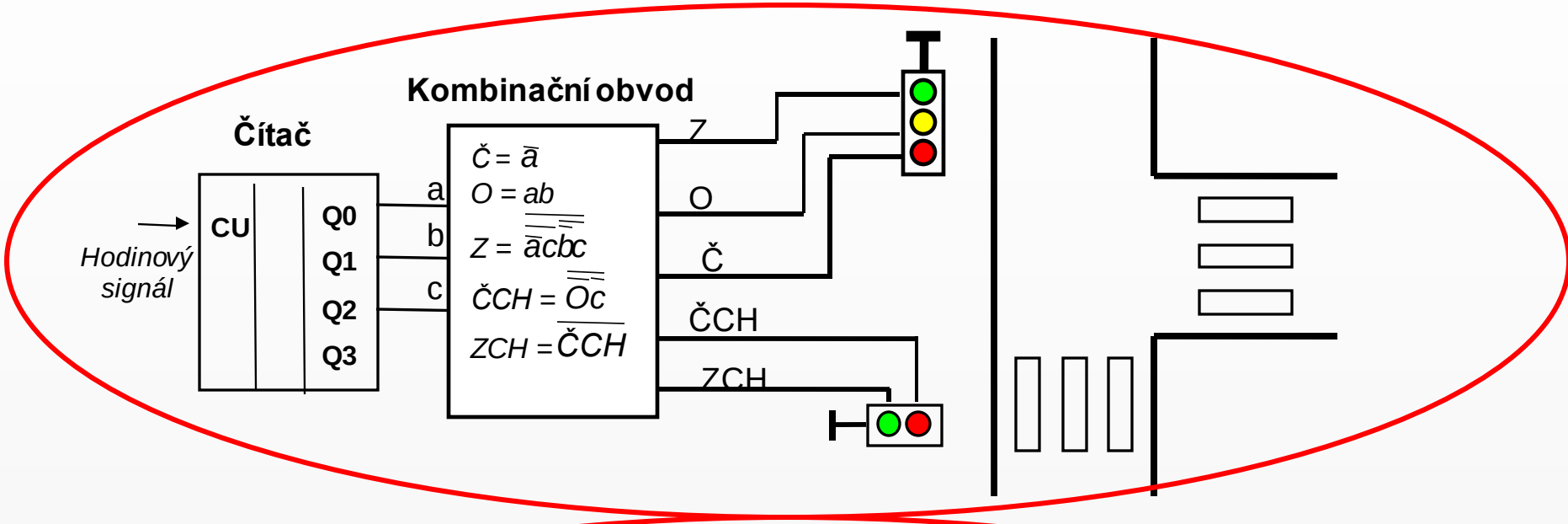
Návrh sekvenčních obvodů

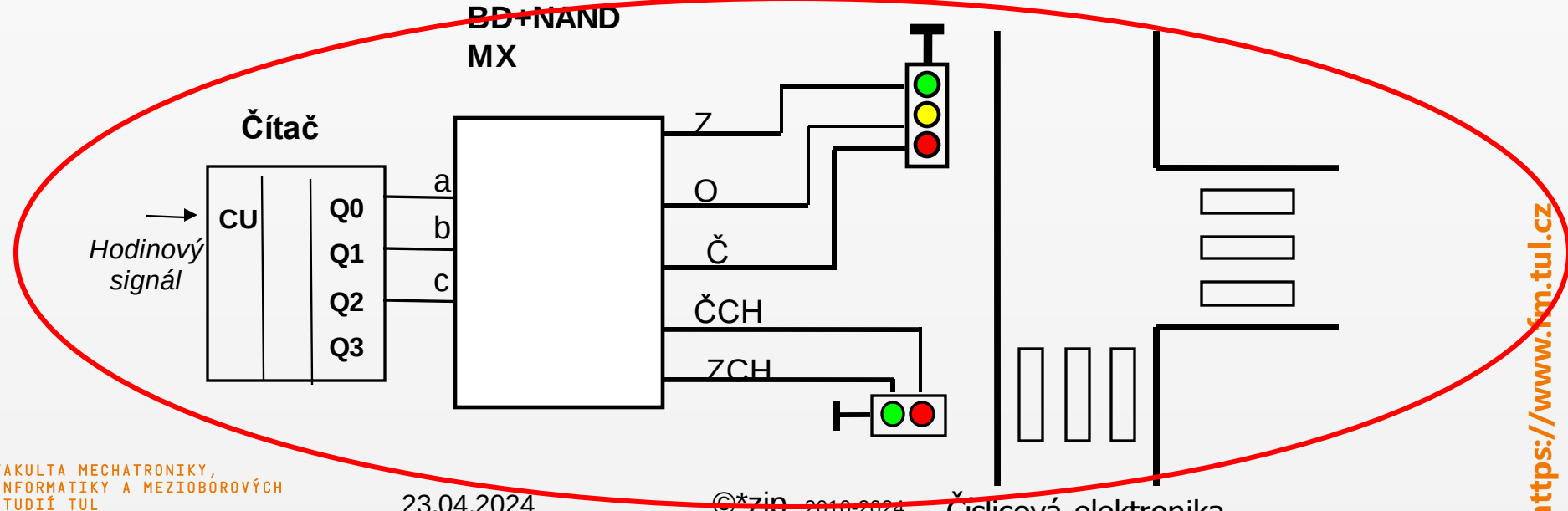
Automat typu MOORE

- jednotlivé stavy „kódovány“ čítačem
- každému stavu odpovídá nějaká kombinace výstupních proměnných

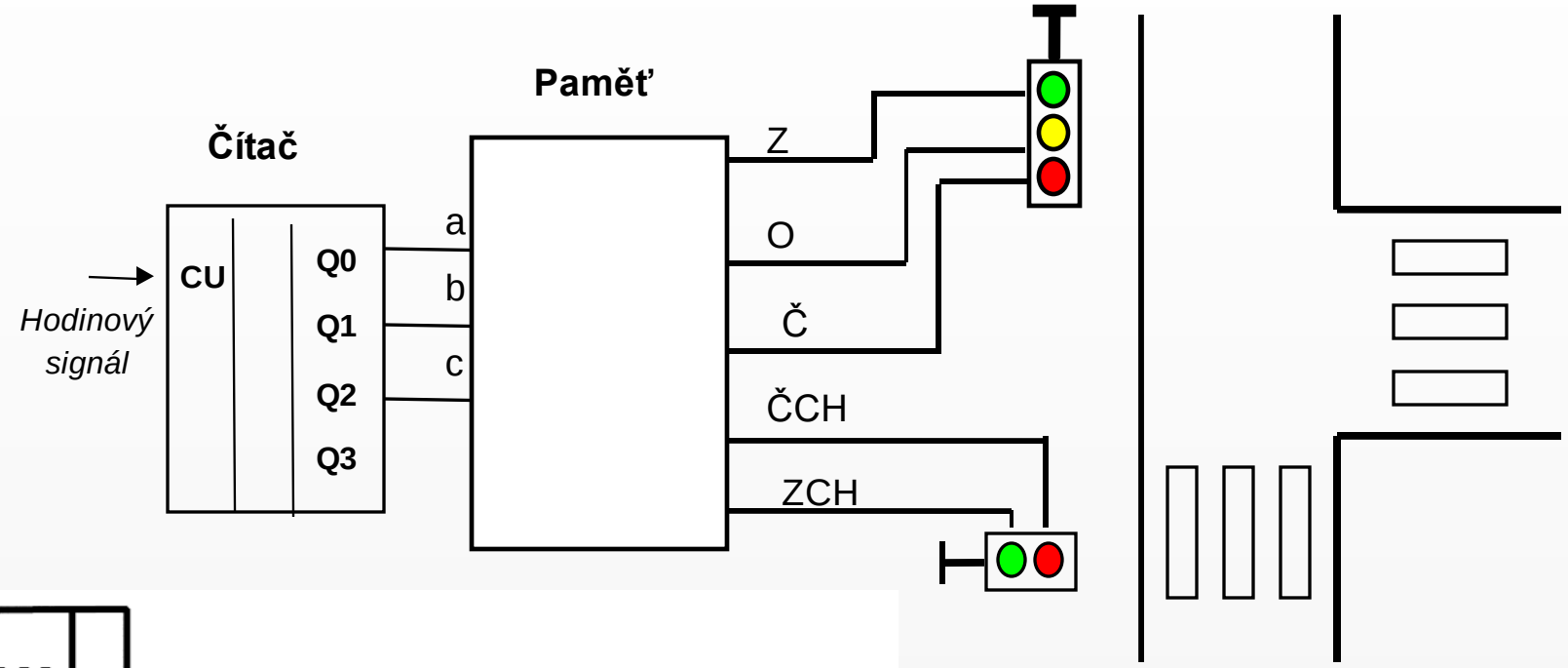


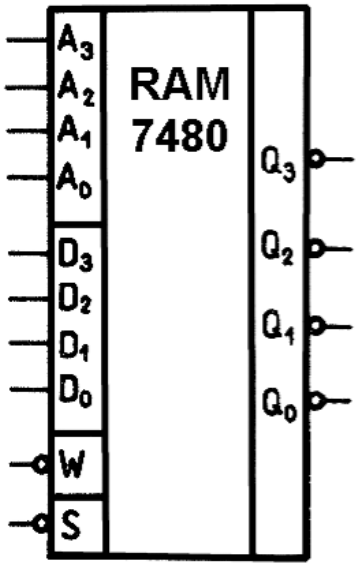
Návrh sekvenčních obvodů





Návrh sekvenčních obvodů



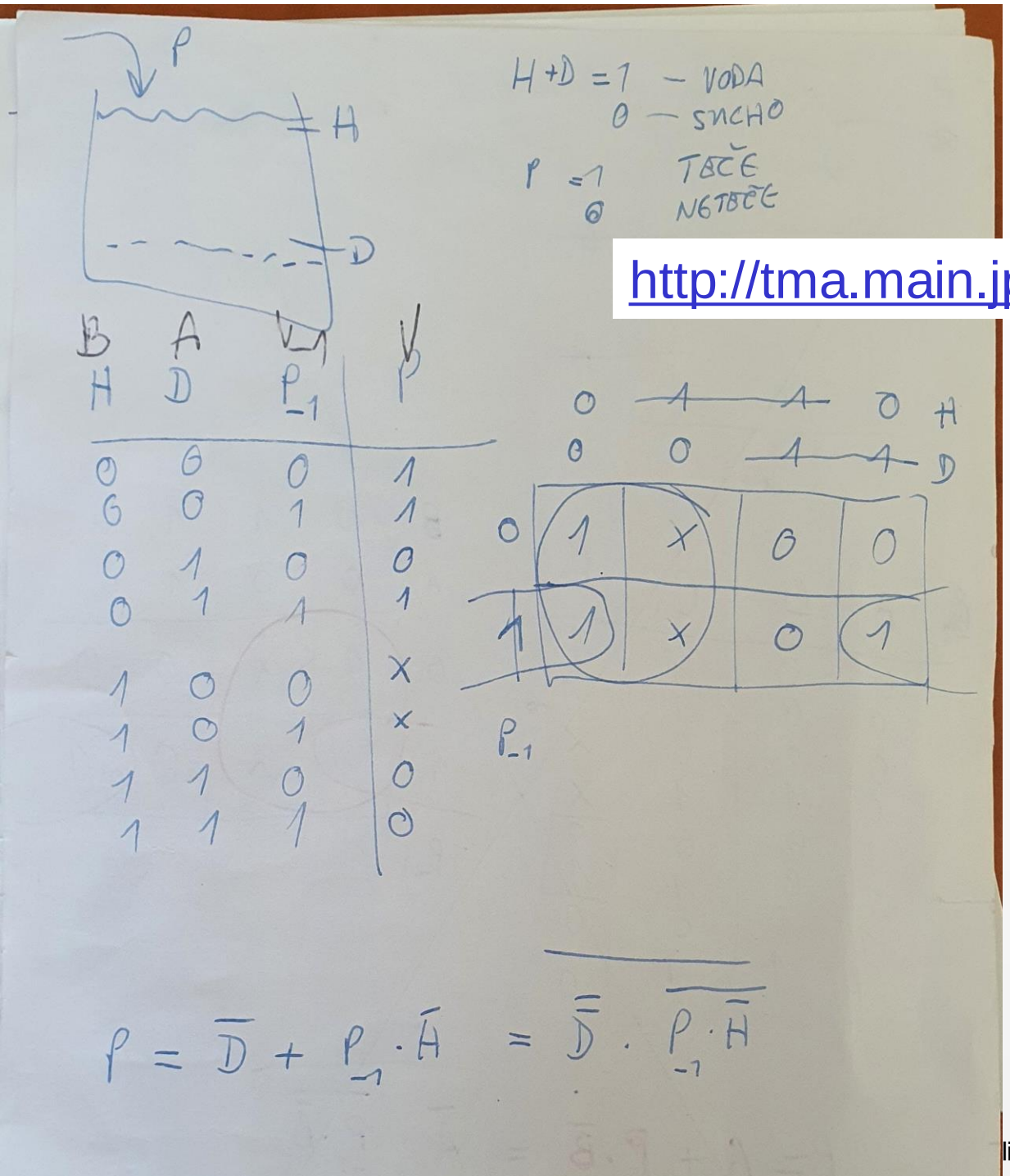


S	W	Akce
0	0	zápis dat
0	1	čtení dat
1	0	přímý přenos dat
1	1	zablokování paměti

Paměť RAM

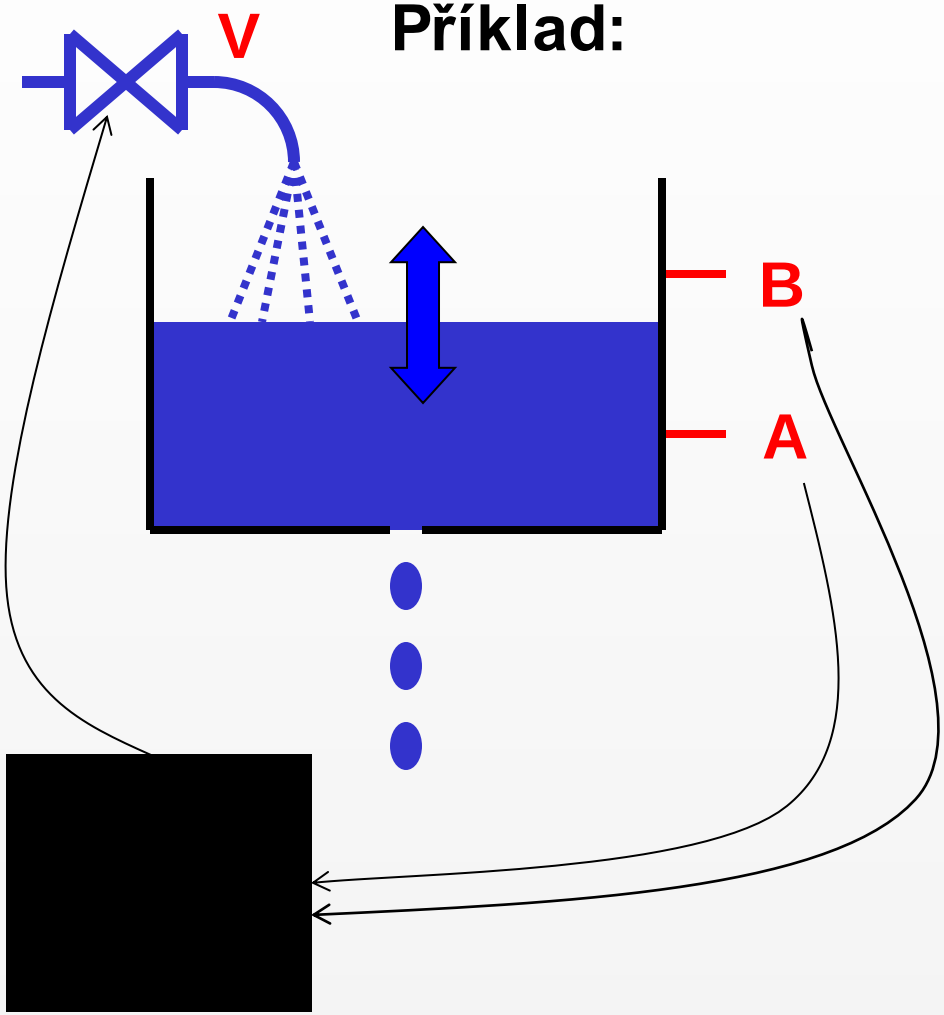
Návrh sekvenčních obvodů

- Čítač + Kombinační logika
- Čítač + Dekodér + hradla NAND
- Čítač + Multiplexer
- Čítač + Paměť
- Počítač
- atd.



Sekvenční automaty

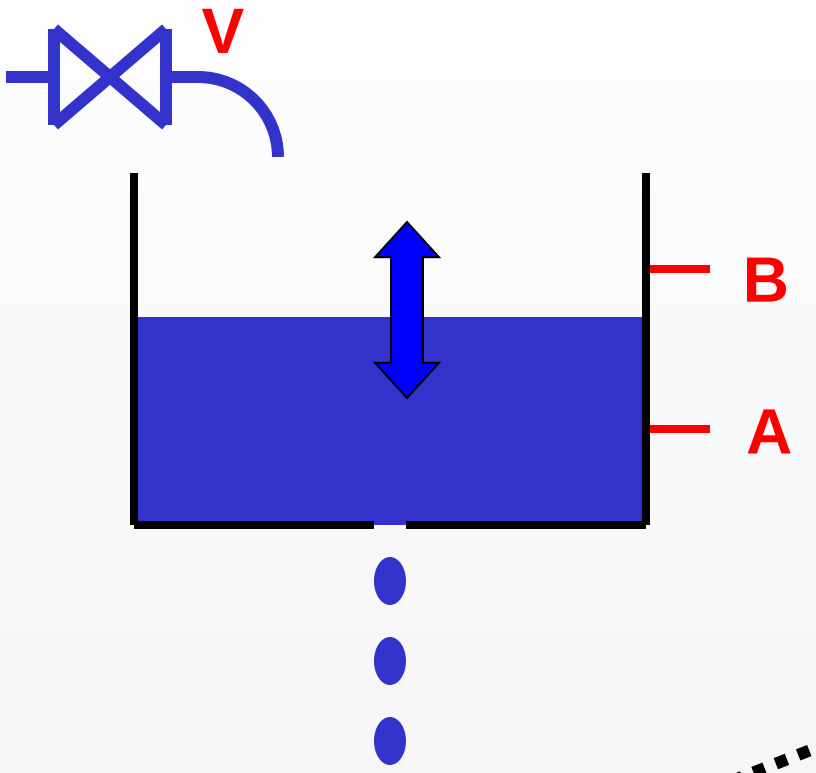
Příklad:



A, B = 1 … pod vodou
A, B = 0 … na suchu

V = 0 … voda neteče
V = 1 … voda teče

Sekvenční automaty

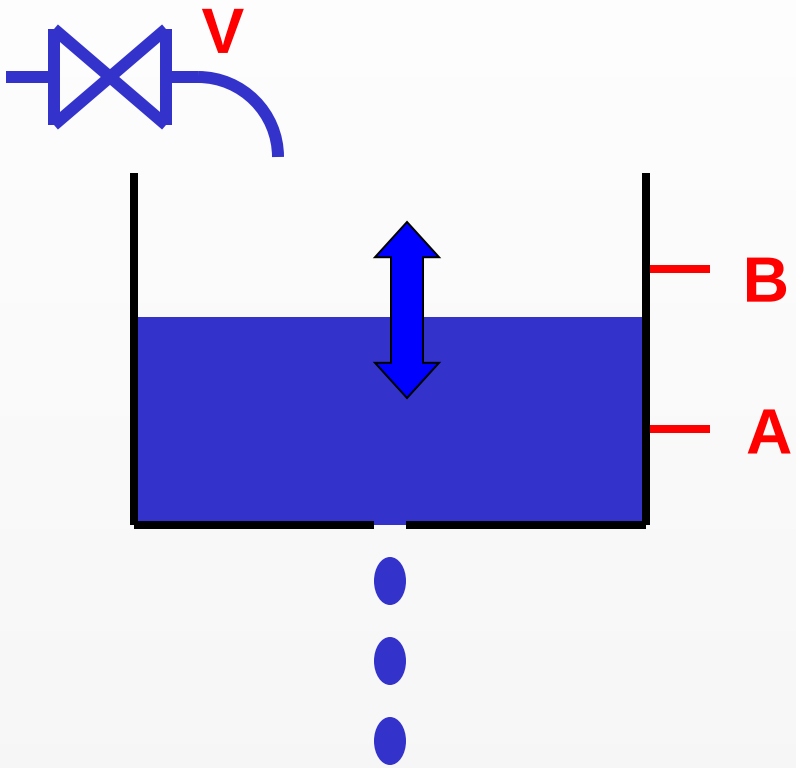


A	B	V
1	1	0
0	0	1
1	0	V_{-1}
0	1	??

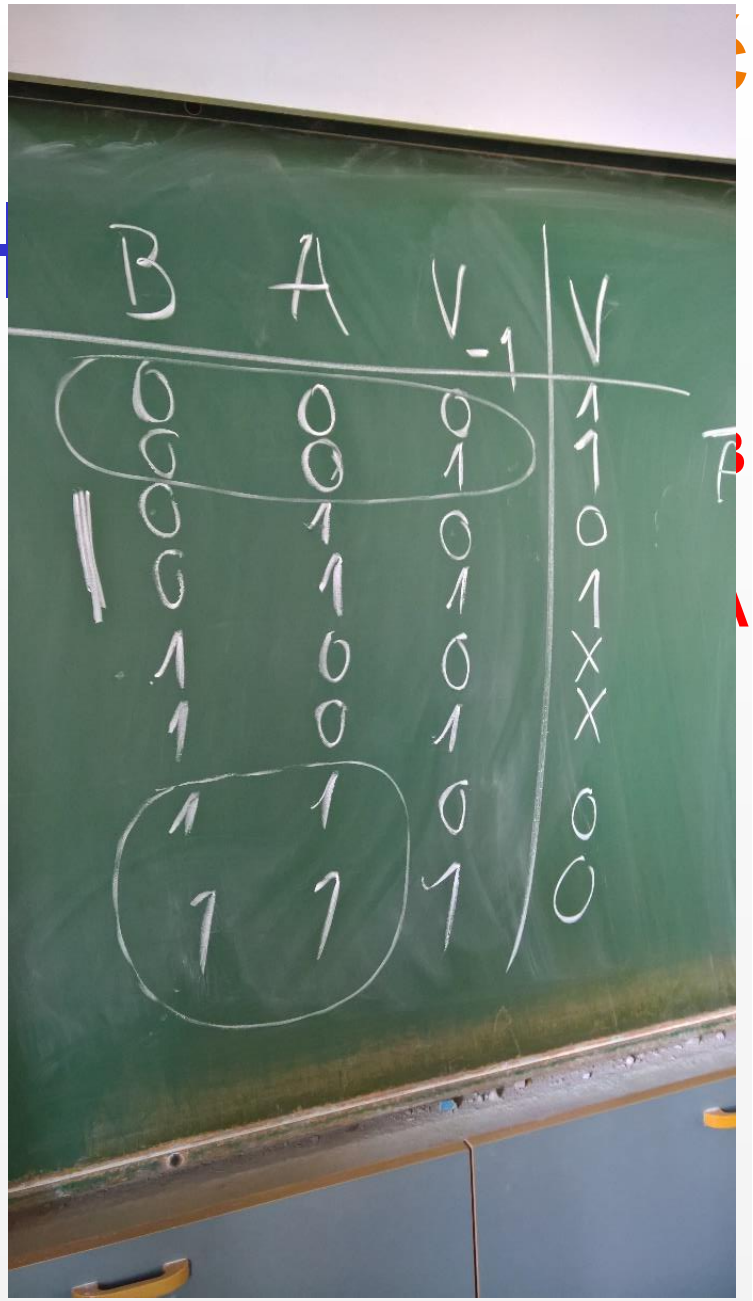
RS

Kde S = /A

Sekvenční automaty



…řešení TF na tabuli…



http://tma.main.jp/logic/index_en.html

Truth table

B	A	N	Output
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

DNF (with ~) = ~B~A~N + ~B~AN + ~BAN + B~A~N + B~AN

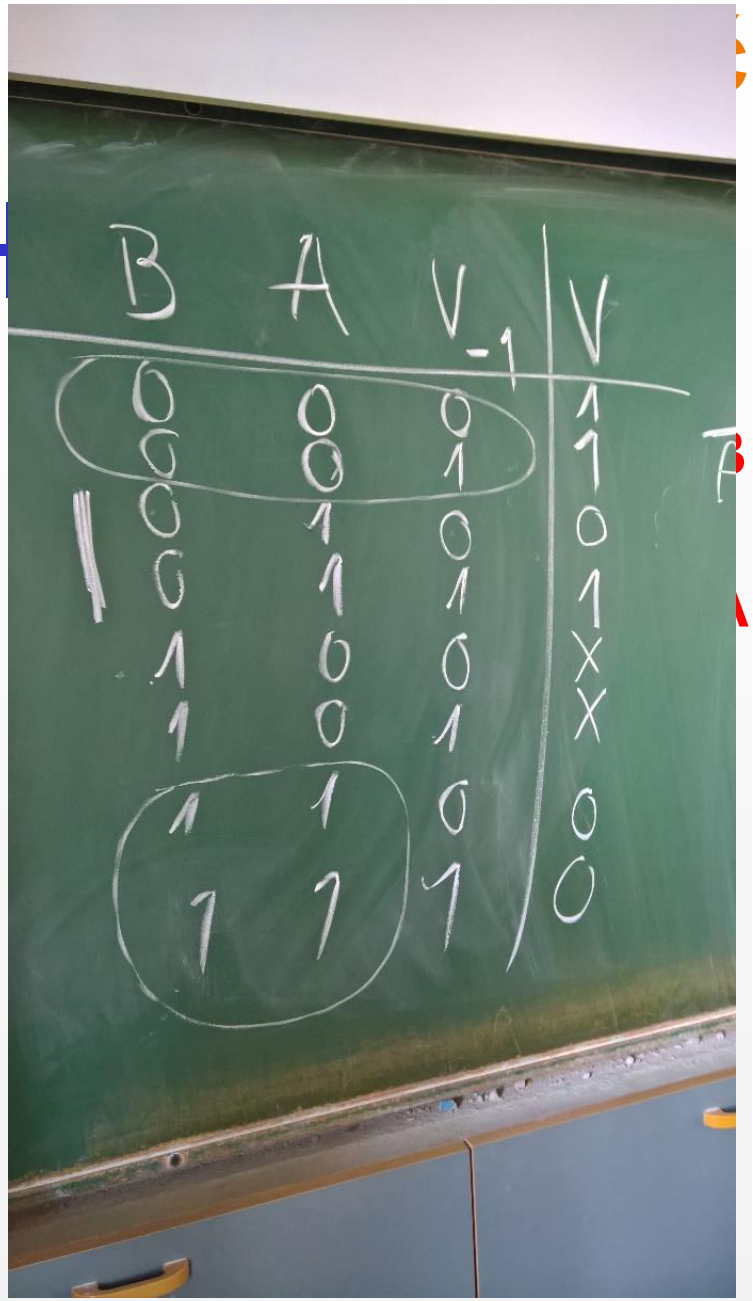
DNF (with overline) = $\overline{B}\overline{A}\overline{N} + \overline{B}\overline{A}N + \overline{B}AN + B\overline{A}\overline{N} + B\overline{A}N$

CNF (with ~) = (B + ~A + N) (~B + ~A + N) (~B + ~A + ~N)

CNF (with overline) = $(B + \overline{A} + N)\ (\overline{B} + \overline{A} + N)\ (\overline{B} + \overline{A} + \overline{N})$

Minimal Form (with ~) = ~BN + ~A

Minimal Form (with overline) = $\overline{B}N + \overline{A}$



http://tma.main.jp/logic/index_en.html

Truth table

B	A	N	Output
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Karnaugh map

	$\overline{N}$	N
$\overline{B}\overline{A}$	1	1
$\overline{B}A$	0	1
BA	0	0
$B\overline{A}$	1	1

DNF (with ~) = ~B~A~N + ~B~AN + ~BAN + B~A~N + B~AN

DNF (with overline) = $\overline{B}\overline{A}\overline{N} + \overline{B}\overline{A}N + \overline{B}AN + B\overline{A}\overline{N} + B\overline{A}N$

CNF (with ~) = (B + ~A + N) (B + ~A + ~N) (~B + ~A + N) (~B + ~A + ~N)

CNF (with overline) = $(B + \overline{A} + N)\ (B + \overline{A} + \overline{N})\ (\overline{B} + \overline{A} + N)\ (\overline{B} + \overline{A} + \overline{N})$

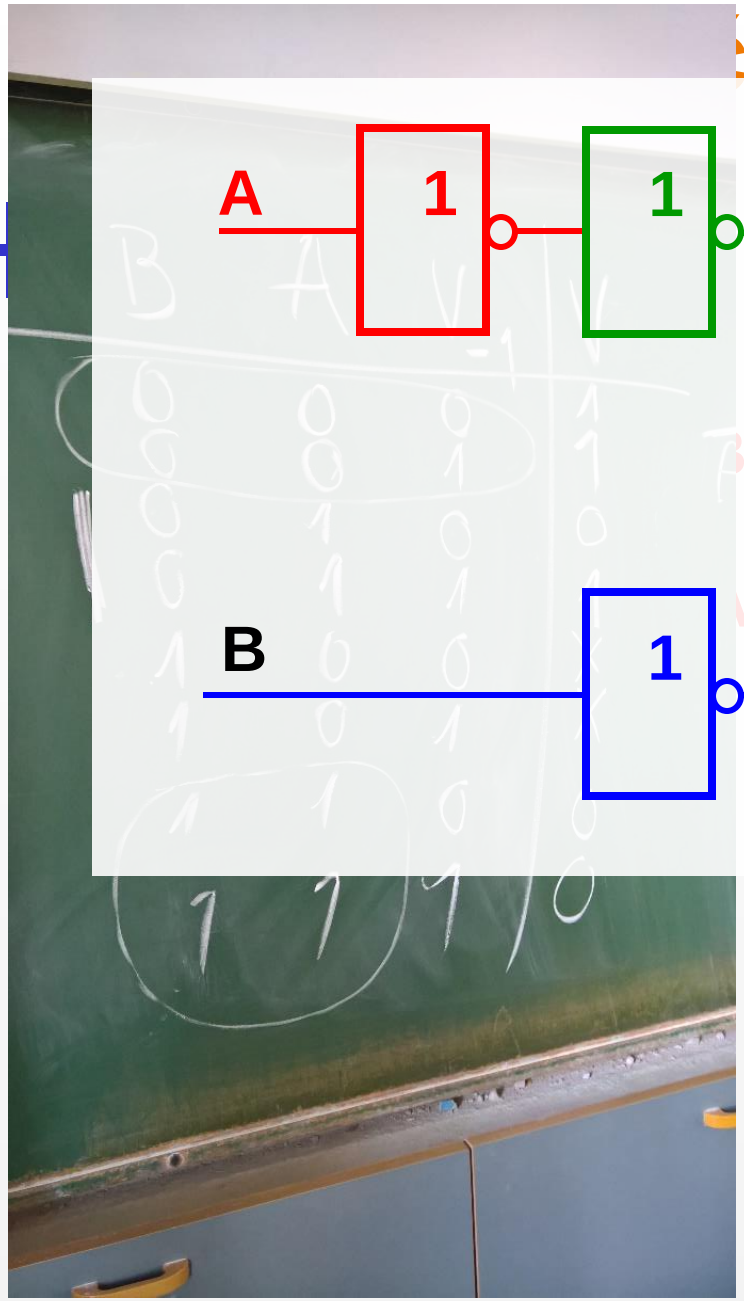
$$V = \overline{A} + V_{-1}.\overline{B}$$

Minimal Form (with ~) = ~BN + ~A

Minimal Form (with overline) = $\overline{B}N + \overline{A}$

$$V = A \,.\, V_{-1}.B$$

http://tma.main.jp/logic/index_en.html



Truth table

B	A	V	Output
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

DNF (with ~) = ~B~A~V + ~B~AV + ~BAV + B~A~V + B~AV

DNF (with overline) = $\bar{B}\bar{A}\bar{V} + \bar{B}\bar{A}V + \bar{B}AV + B\bar{A}\bar{V} + B\bar{A}V$

CNF (with ~) = (B + ~A + V) (~B + ~A + V) (~B + ~A + ~V)

CNF (with overline) = $(B + \bar{A} + V)\ (\bar{B} + \bar{A} + V)\ (\bar{B} + \bar{A} + \bar{V})$

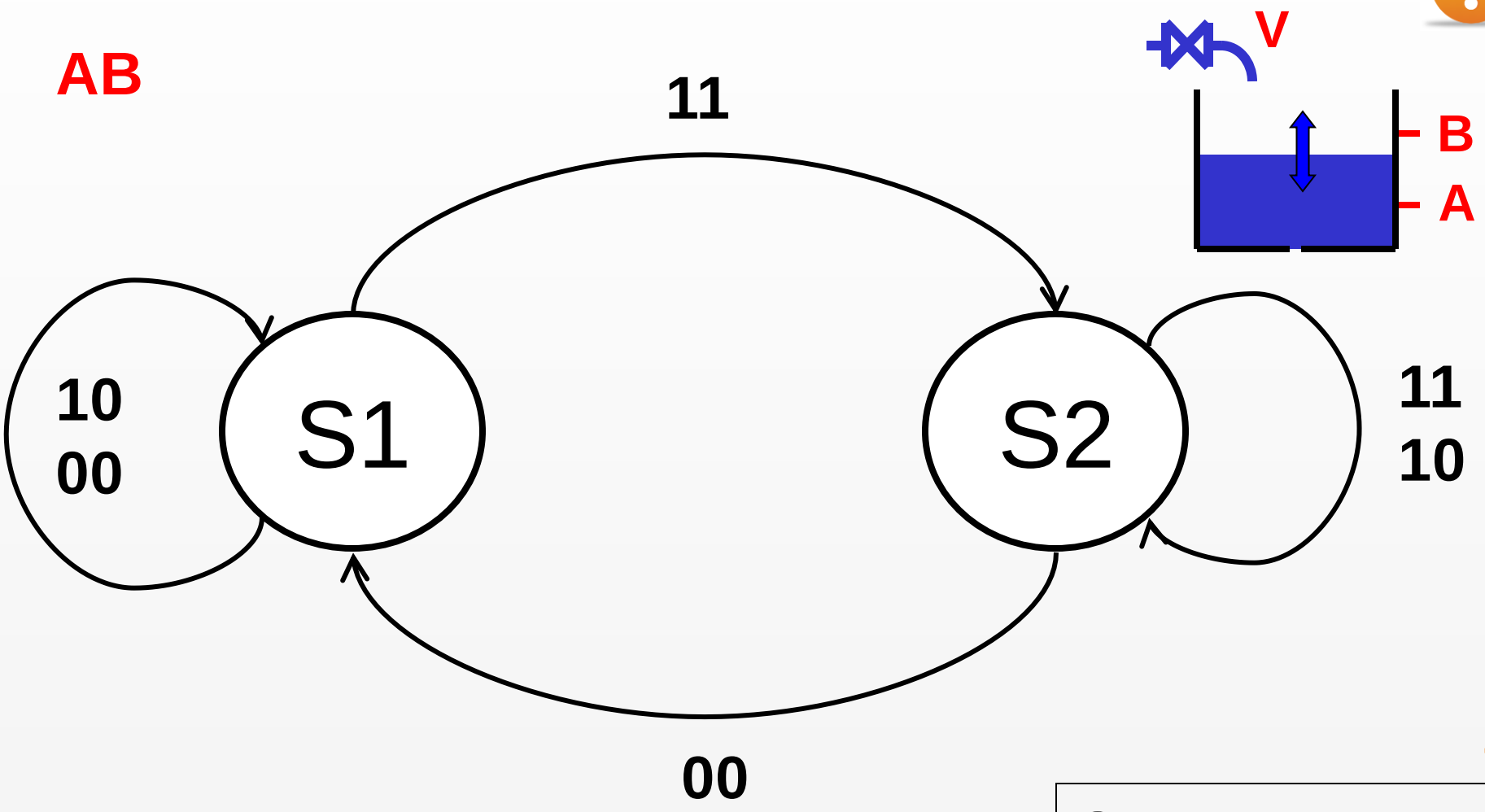
$$V = \bar{A} + V_{-1}.\bar{B}$$

Minimal Form (with ~) = ~BV + ~A

Minimal Form (with overline) = $\bar{B}V + \bar{A}$

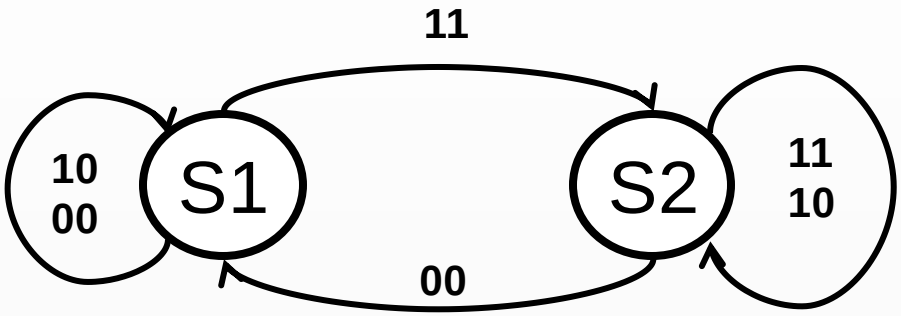
$$V = A \,.\, V_{-1}.B$$

Sekvenční automaty



S1 … V = 1
S2 … V = 0

Sekvenční automaty



S1 … V = 1
S2 … V = 0

B
A

S2	S1	S2	S2	X
S1	S1	S1	S2	X

Tabulka přechodů

B
A

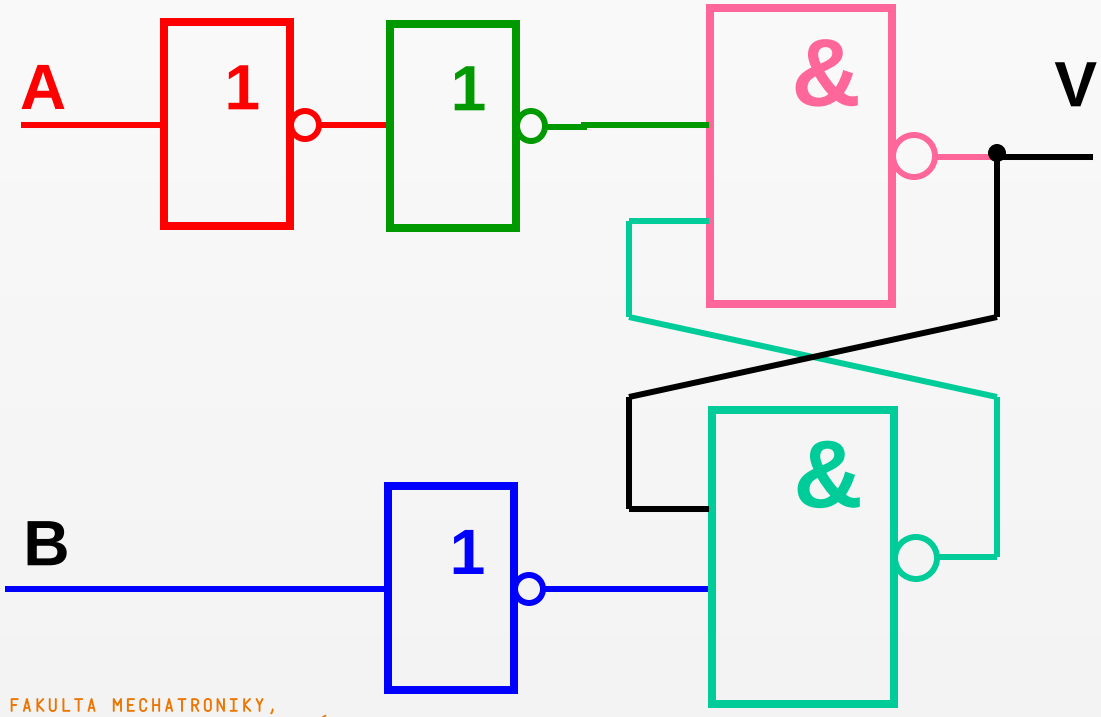
Tabulka výstupů

V	1	0	0	X
	1	1	0	X

Sekvenční automaty

v \ B A				
	1	0	0	X
	1	1	0	X

$$V_+ = \overline{A} + V.\overline{B}$$



$$V_+ = \overline{A} \,.\, \overline{V.B}$$

Jazyky HDL

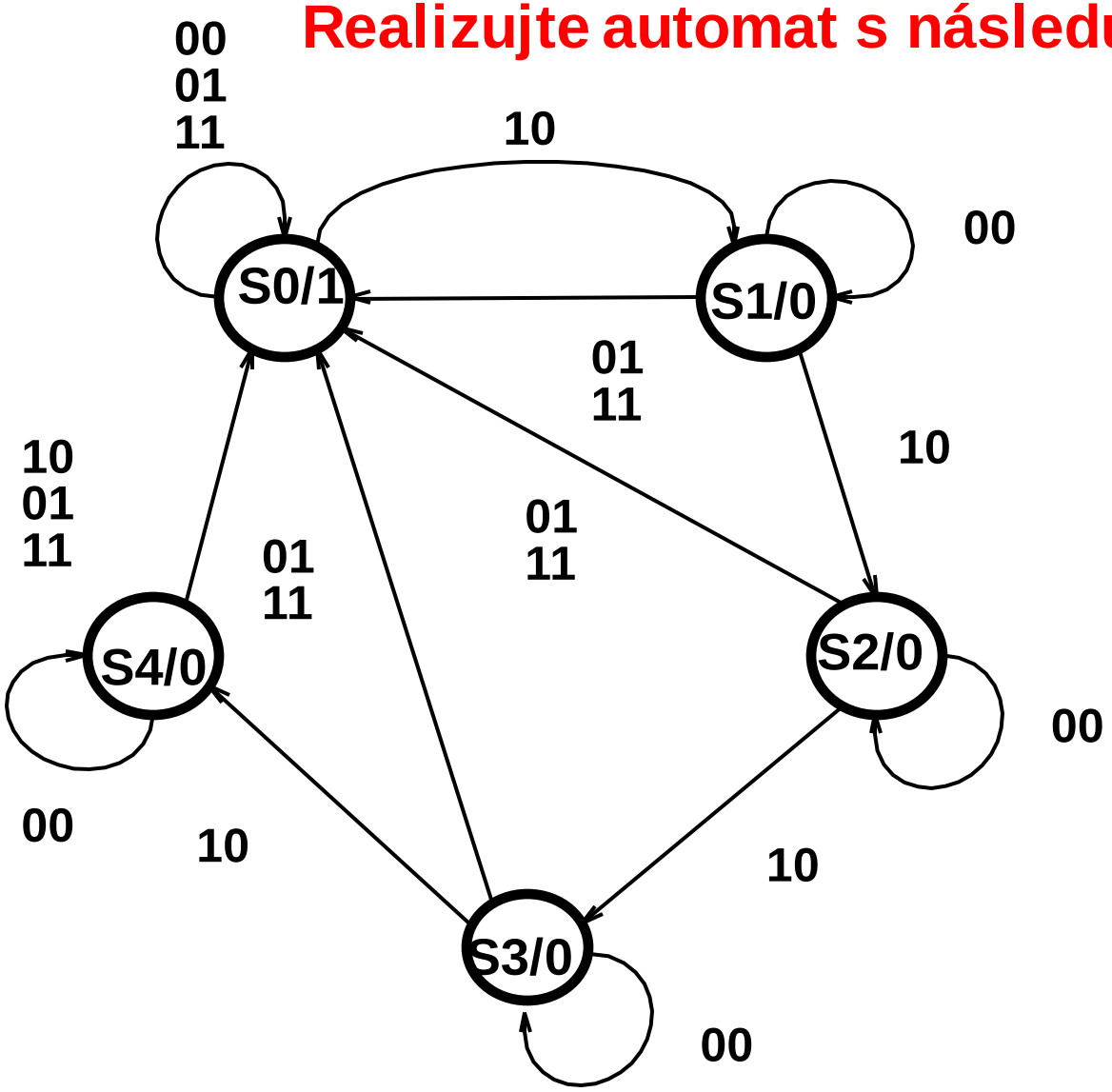
- **HDL** (Hardware Description Language)
  - jazyk pro vstupní popis logických obvodů pomocí rovnic, pravdivostních tabulek a stavových diagramů
  - není standardizováno (Abel-HDL, Altera-HDL, …)
- **VHDL** (Very High Speed Integrated Circuits HDL) + **Verilog** HDL
  - popis chování složitých systémů
  - nezávislé na budoucí technologii realizace
  - vhodné pro návrh metodou shora-dolů
  - standardy IEEE

Příklad souboru HDL

```
MODULE citac
TITLE 'Desitkovy citac'
DECLARATIONS
   clock, reset PIN ;
   q3, q2, q1, q0 PIN 17,16,15,14 ISTYPE 'reg' ;
   citac = [q3 .. q0] ;
EQUATIONS
   citac := (citac + 1) & (citac < 9) & (!reset) ;
   citac.clk = clock ;
TEST_VECTORS ( [clock,reset] -> citac )
                  [  .c.  ,   1   ] ->   0 ;
                  [  .c.  ,   0   ] ->   1 ;
END
```

Příklad

Realizujte automat s následujícím grafem



```
STATE_DIAGRAM sreg          " stavový diagram automatu
STATE reset: IF vstup==2 THEN S1 ELSE S0 ;
" při vstupu=2, jdi na stav S1, jinak na S0
STATE S0:
        IF vstup==2 THEN S1 ELSE S0 ;
STATE S1:
        IF vstup==0 THEN S1 ELSE
        IF vstup==2 THEN S2 ELSE S0 ;
STATE S2:
        IF vstup==0 THEN S2 ELSE
        IF vstup==2 THEN S3 ELSE S0 ;
STATE S3:
        IF vstup==0 THEN S3 ELSE
        IF vstup==2 THEN S4 ELSE S0 ;
STATE S4:
        IF vstup==0 THEN S4 ELSE S0 ;
```

Rozdělení pamětí

Podle uchování obsahu

volatilní - po odpojení napájecího napětí ztrácí svůj původní obsah (závislé na nap. napětí)

nonvolatilní - jsou nezávislé na napájecím napětí.

Volatilní paměti

= při ztrátě napájení ztratí obsah, t.j. po přivedení napájení náhodný obsah

- zápis stejně rychlý jako čtení
- neomezený počet cyklů zápisu

❑ **SRAM** – do 16 MB

- klopný obvod z 6 tranzistorů
- zanedbatelná klidová spotřeba; rychlé

❑ **DRAM** – do 128 MB

- paměťový *kondenzátor* (<< 1 pF): 0 = vybitý, 1 = nabitý

+ tranzistorový spínač (vyšší integrace)

⇒ nutné dobíjení (refresh) periodicky po několika ms (pomocné obvody)

- pomalejší, vyšší spotřeba

Nevolatilní paměti

Speciální nevolatilní

- při ztrátě napájení se obsah neztrácí, jinak jako SRAM

- **FRAM** (Ferroelectric RAM)
  - změna polohy atomu uvnitř krystalu elektrickým polem
  - rychlé jako SRAM; vysoká integrace jako DRAM, zatím malé kapacity

- **NV SRAM** (Nonvolatile SRAM, Battery-backed SRAM)
  - vestavěná baterie, velkokapacitní C. akumulátor zálohuje napájení

Nevolatilní paměti

= při ztrátě napájení se obsah neztrácí

- zápis řádově pomalejší než čtení
- omezený počet cyklů zápisu

- **ROM** (Read Only Memory) - programována maskou u výrobce, nelze změnit; pro hromadné aplikace (generátory znaků, melodií,…)
- **PROM** (Programmable ROM) – propojky n. OTP EPROM, jednou programovatelná za *jednotky minut* zvýšeným napětím v *programátoru* (u uživatele)
- **EPROM** (Erasable PROM) – vymazatelné UV zářením přes okénko (⇒ drahé pouzdro), ca 10^2 cyklů, do 512 KB
- **EEPROM** (Electrically EPROM) – i mazání jen zvýšeným napětím (v programátoru nebo *v systému*), až 10^7 cyklů za *desítky sekund*; podle typu přepis po slovech, mazání po stránkách nebo celá paměť
- **Flash (EEPROM)** – zápis i mazání prováděn po blocích, mazání je velmi rychlé (mžikové), až 10^5 cyklů, větší integrace – do 256 MB

Rozdělení pamětí

Podle způsobu obnovy informace

- **statické** - informace zůstává uchována bez obnovování, (vyšší cena)
- **dynamické** - informace se musí periodicky obnovovat cyklem čtení, náročnější na řídící logiku

Další dělení pamětí

Podle fyz. principu uložení informace

- **magnetické** - magnetické vlastností materiálu, směr magnetizace.
- **optické** - optických vlastností materiálu, (odraz světla)
- **magnetooptické** – světelným paprskem (laser) se mění magnetické vlastnosti materiálu
- **feritové** - feritové jádro (rozměr cca 0,8 mm), magnetická orientace se překlápí proudovým impulsem (zastaralé)
- **polovodičové** - polovodičové tranzistory (TTL, CMOS)

Další dělení pamětí

Podle mechanického provedení:

- pásková paměť
- disková paměť
- disketa
- CD-ROM, DVD
- flash paměť
- SIMM, DIMM ...

Rozdělení pamětí

Podle přístupu k jedn. buňkám

- **adresovatelné**
  - **RAM** - paměť s libovolným přístupem; doba přístupu k obsahu není závislá na umístění (adrese)
  - **SAM** - paměť se sekvenčním přístupem, doba přístupu k obsahu je závislá na umístění, například páska
- **asociativní**
  - adresovaná obsahem, adresou je klíčová hodnota ukládaná s informací

Rozdělení pamětí

Podle možnosti změny obsahu

- **RWM** (Read Write Memory) - paměť pro opakovaný zápis i čtení
- **ROM** (Read Only Memory) - paměť pouze pro čtení, informace je uložena jednorázově při výrobě
- **WOM** (Write Only Memory) - při provozu jen pro zápis, informace je čtena jednorázově na konci provozního cyklu - např. tzv. černá skříňka

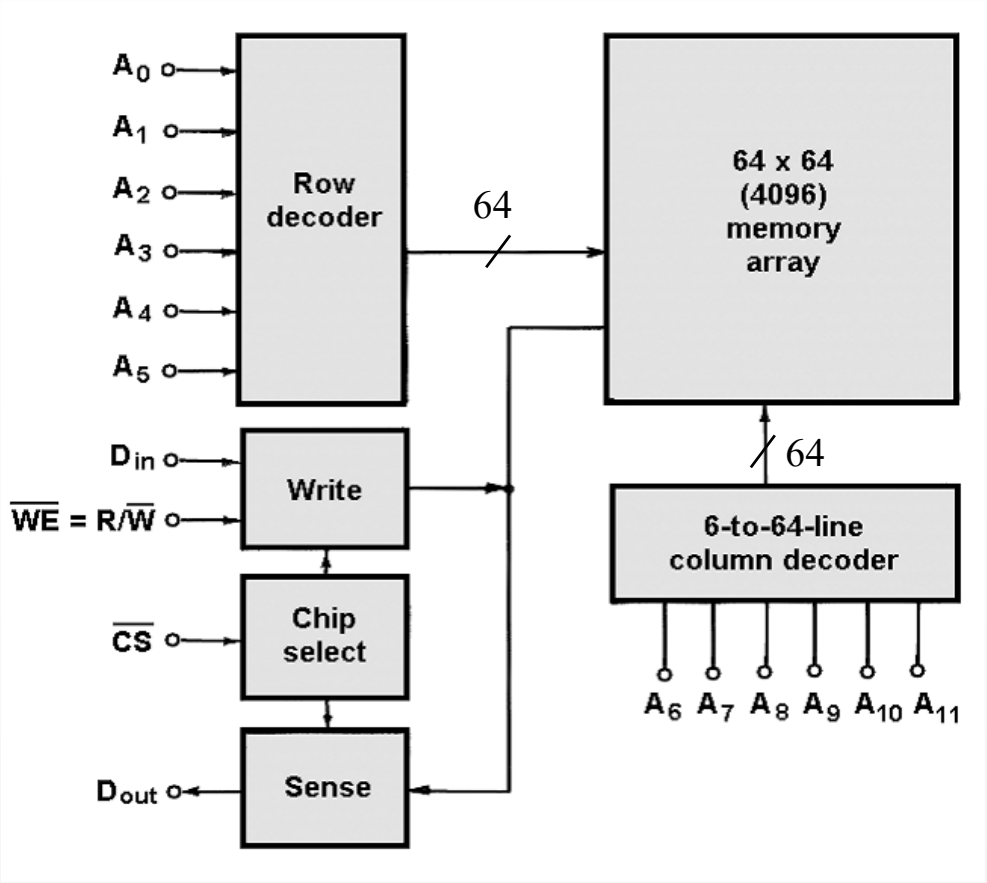
Pozor:
*Výraz **RWM-RAM** se nevžil a místo něj se běžně používá označení **RAM**,*
*Ani výraz **ROM-RAM** se nevžil a používá se označení **ROM***

Další dělení pamětí

Podle určení:

- **Registry procesoru** - paměť s velmi malou kapacitou, rychlá stejně jako procesor, součást čipu procesoru, který ji používá pro uchovávání operandů a výsledků aritmetických a logických operací
- **Operační paměť** - vnitřní paměť pro práci procesoru počítače, rychlá, ale podstatně pomalejší než procesor
- **Cache** - rychlá vyrovnávací paměť s malou kapacitou, rychlost srovnatelná s procesorem
- **Vnější paměť** - pro dlouhodobé a bezpečné uložení souborů počítače, velká kapacita, malá rychlost

Uspořádání pamětí



Integrovaná paměť = paměťové buňky s pomocnými obvody
př. SRAM 4096 b, t.j. 12bitová adresa; uspořádání po 1 b

Uspořádání pamětí

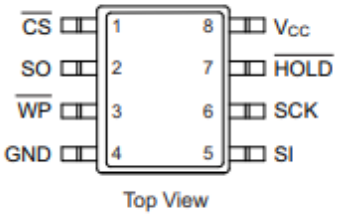
AT25128B and AT25256B

SPI Serial EEPROM

128K (16,384 x 8), 256K (32,768 x 8)

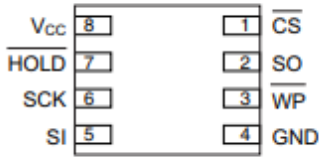
Pin Name	Function
$\overline{CS}$	Chip Select
GND	Ground
$\overline{HOLD}$	Suspends Serial Input
SCK	Serial Data Clock
SI	Serial Data Input
SO	Serial Data Output
V_{CC}	Power Supply
$\overline{WP}$	Write Protect

8-lead SOIC

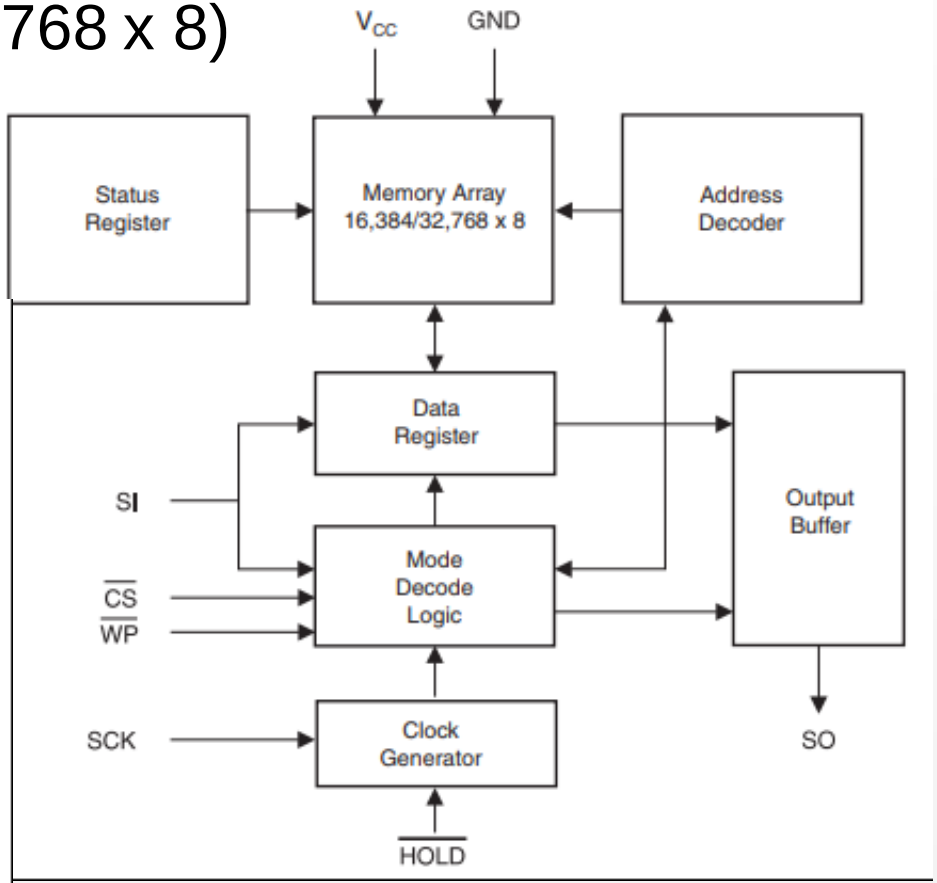


Top View

8-pad UDFN



Bottom View



Uspořádání pamětí

- Délka slova: 1, 4, 8, 16 bitů

ROZHRANÍ

- **paralelní**
  - bity adresní (dle počtu buněk)
- - bity datové (dle šířky slova)
  - řídicí signály CS (Chip Select), RD, WR
- Rychlost přístupu 10^{-6}...10^{-8} s
- **Sériové sběrnice:**
- **SPI** 4 linky až 33 Mb/s
- **I²C** 2 linky až 400 kb/s
- Po 1 datovém vodiči se přenáší příkazy, adresa i data postupně.

Děkuji za pozornost…

Zdeněk Plíva
zdenek.pliva@tul.cz
Tel.: 3536