

# Struktura počítačů

Jan Outrata



KATEDRA INFORMATIKY  
UNIVERZITA PALACKÉHO V OLMOUCI

přednášky



# Číselné soustavy

Počítač = počítačící stroj ... počítání s čísly (původně)

## Člověk:

- deset hodnot a symbolů pro ně: číslice **0** až **9**
- pro reprezentaci (zápis) čísla použití **desítkové (dekadické) poziční číselné soustavy**: číslo jako součet mocninné řady o základu (radixu) 10, zápis = posloupnost symbolů pro koeficienty řady, pozice (pořadí) symbolu určuje mocninu (řád)

$$(1024)_{10} = 1 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0$$

- jiné číselné soustavy: dvanáctková (hodiny), šedesátková (minuty, sekundy), dvacítková (dřívější platidla) aj.

## Počítač:

- první (elektro)mechanické počítačí stroje dekadické (tj. používající desítkovou soustavu) – u součástí potřeba 10 stabilních stavů (pro deset hodnot)
- elektromechanické a elektronické součásti: nejnadhěji realizovatelné **2 stabilní stavy** pro 2 hodnoty, symboly (číslíce) **0** a **1** ( $\Rightarrow$  **digitální zařízení/elektronika**)
- pro reprezentaci (zápis) čísla použití **dvojkové (binární) poziční číselné soustavy**: číslo jako součet mocninné řady o základu 2, zápis = posloupnost symbolů pro koeficienty, pozice symbolu určuje mocninu

$$(11)_{10} = (1011)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

- další typy dat (čísla s řádovou čárkou, znaky a texty, obrázky, zvuky, videa atd.) odvozeny od (celých) čísel  $\Rightarrow$  **binární reprezentace** všech typů dat

## Počítač:

- první (elektro)mechanické počítačí stroje dekadické (tj. používající desítkovou soustavu) – u součástí potřeba 10 stabilních stavů (pro deset hodnot)
- elektromechanické a elektronické součásti: nejnadhěji realizovatelné **2 stabilní stavy** pro 2 hodnoty, symboly (číslice) **0** a **1** ( $\Rightarrow$  **digitální zařízení/elektronika**)
- pro reprezentaci (zápis) čísla použití **dvojkové (binární) poziční číselné soustavy**: číslo jako součet mocninné řady o základu 2, zápis = posloupnost symbolů pro koeficienty, pozice symbolu určuje mocninu

$$(11)_{10} = (1011)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

- další typy dat (čísla s řádovou čárkou, znaky a texty, obrázky, zvuky, videa atd.) odvozeny od (celých) čísel  $\Rightarrow$  **binární reprezentace** všech typů dat

## Počítač pro člověka:

- použití (pozičních) číselných soustav o základu  $2^k$  ( $\approx$  „kompromis“)
  - **osmičkové (oktalové)**: symboly (číslice) **0** až **7**
  - **šestnáctkové (hexadecimální)**: symboly (číslice) **0** až **9** a **A** až **F**

## Věta (O reprezentaci přirozených čísel (včetně 0))

*Libovolné přirozené číslo  $N$  (včetně 0) lze vyjádřit jako součet mocninné řady o základu  $B \geq 2, B \in \mathbb{N}$ :*

$$N = a_{n-1} \cdot B^{n-1} + a_{n-2} \cdot B^{n-2} + \dots + a_1 \cdot B^1 + a_0 \cdot B^0,$$

*kde  $0 \leq a_i < B, a_i \in \mathbb{N}$  jsou koeficienty řady.*

*Číslo  $N$  se (v poziční číselné soustavě o základu  $B$ ) zapisuje jako řetěz symbolů (číslic)  $S_i$  pro koeficienty  $a_i$  zleva v pořadí pro  $i$  od  $n - 1$  k 0:*

$$(S_{n-1}S_{n-2} \dots S_1S_0)_B$$

Získání (hodnoty) čísla  $N$  z jeho zápisu  $(S_{n-1}S_{n-2}\dots S_1S_0)_B$  – postupným přičítáním:

$$N = a_0$$

$$B' = B$$

**for**  $i = 1$  **to**  $n - 1$  **do**

$$N = N + a_i * B'$$

$$B' = B' * B$$

pro  $(1024)_{10}$  ( $B = 10, n = 4, a_3 = 1, a_2 = 0, a_1 = 2, a_0 = 4$ ):

$$N = 4, B' = 10$$

$$i = 1 : N = 24, B' = 100$$

$$i = 2 : N = 24, B' = 1000$$

$$i = 3 : N = 1024, B' = 10000$$

Získání (hodnoty) čísla  $N$  z jeho zápisu  $(S_{n-1}S_{n-2}\dots S_1S_0)_B$  – postupným přičítáním:

```
 $N = a_0$   
 $B' = B$   
for  $i = 1$  to  $n - 1$  do  
   $N = N + a_i * B'$   
   $B' = B' * B$ 
```

```
pro  $(1024)_{10}$  ( $B = 10, n = 4, a_3 = 1, a_2 = 0, a_1 = 2, a_0 = 4$ ):  
 $N = 4, B' = 10$   
 $i = 1 : N = 24, B' = 100$   
 $i = 2 : N = 24, B' = 1000$   
 $i = 3 : N = 1024, B' = 10000$ 
```

Získání zápisu  $(S_{n-1}S_{n-2}\dots S_1S_0)_B$  (hodnoty) čísla  $N$  – postupným odečítáním:

```
 $B' = 1, i = 0$   
while  $B' * B \leq N$  do  
   $B' = B' * B$   
   $i = i + 1$   
for  $i$  to  $0$  do  
   $a_i = N / B'$  ; celočíselné dělení  
   $N = N - a_i * B'$  ;  $= N \bmod B'$  (zbytek)  
   $B' = B' / B$ 
```

```
pro  $N = 1024, B = 10$ :  
 $B' = 1, i = 0$   
 $10 \leq 1024 : B' = 10, i = 1$   
 $100 \leq 1024 : B' = 100, i = 2$   
 $1000 \leq 1024 : B' = 1000, i = 3$   
 $10000 \not\leq 1024$   
 $i = 3 : a_i = 1, N = 24, B' = 100$   
 $i = 2 : a_i = 0, N = 24, B' = 10$   
 $i = 1 : a_i = 2, N = 4, B' = 1$   
 $i = 0 : a_i = 4, N = 0, B' = 0$ 
```



## Hodnota čísla vs. jeho zápis (rychleji)



$$\begin{aligned} N &= a_{n-1} \cdot B^{n-1} + a_{n-2} \cdot B^{n-2} + \dots + a_1 \cdot B + a_0 \\ &= (\dots (\mathbf{a}_{n-1} \cdot \mathbf{B} + \mathbf{a}_{n-2}) \cdot \mathbf{B} + \dots + \mathbf{a}_1) \cdot \mathbf{B} + \mathbf{a}_0 \end{aligned}$$

## Hodnota čísla vs. jeho zápis (rychleji)



$$\begin{aligned} N &= a_{n-1} \cdot B^{n-1} + a_{n-2} \cdot B^{n-2} + \dots + a_1 \cdot B + a_0 \\ &= (\dots (\mathbf{a}_{n-1} \cdot \mathbf{B} + \mathbf{a}_{n-2}) \cdot \mathbf{B} + \dots + \mathbf{a}_1) \cdot \mathbf{B} + \mathbf{a}_0 \end{aligned}$$

Získání (hodnoty) čísla  $N$  z jeho zápisu  $(S_{n-1}S_{n-2}\dots S_1S_0)_B$  – postupným násobením:

```
 $N = a_{n-1}$   
for  $i = n - 2$  to  $0$  do  
   $N = N * B + a_i$ 
```

```
pro  $(1024)_{10}$  ( $B = 10, n = 4, a_3 = 1, a_2 = 0, a_1 = 2, a_0 = 4$ ):  
 $N = 1$   
 $i = 2 : N = 10$   
 $i = 1 : N = 102$   
 $i = 0 : N = 1024$ 
```

# Hodnota čísla vs. jeho zápis (rychleji)



$$\begin{aligned} N &= a_{n-1} \cdot B^{n-1} + a_{n-2} \cdot B^{n-2} + \dots + a_1 \cdot B + a_0 \\ &= (\dots (\mathbf{a}_{n-1} \cdot \mathbf{B} + \mathbf{a}_{n-2}) \cdot \mathbf{B} + \dots + \mathbf{a}_1) \cdot \mathbf{B} + \mathbf{a}_0 \end{aligned}$$

Získání (hodnoty) čísla  $N$  z jeho zápisu  $(S_{n-1}S_{n-2}\dots S_1S_0)_B$  – postupným násobením:

```
 $N = a_{n-1}$   
for  $i = n - 2$  to  $0$  do  
   $N = N * B + a_i$ 
```

```
pro  $(1024)_{10}$  ( $B = 10, n = 4, a_3 = 1, a_2 = 0, a_1 = 2, a_0 = 4$ ):  
 $N = 1$   
 $i = 2 : N = 10$   
 $i = 1 : N = 102$   
 $i = 0 : N = 1024$ 
```

Získání zápisu  $(S_{n-1}S_{n-2}\dots S_1S_0)_B$  (hodnoty) čísla  $N$  – postupným dělením:

```
 $a_0 = N \bmod B$   
 $i = 1$   
while  $N \geq B$  do  
   $N = N / B$  ; celočíselné dělení  
   $a_i = N \bmod B$  ; zbytek  
   $i = i + 1$ 
```

```
pro  $N = 1024, B = 10$ :  
 $a_0 = 4, i = 1$   
 $1024 \geq 10 : N = 102, a_1 = 2, i = 2$   
 $102 \geq 10 : N = 10, a_2 = 0, i = 3$   
 $10 \geq 10 : N = 1, a_3 = 1, i = 4$   
 $1 \not\geq 10$ 
```



- 1 získání (hodnoty) čísla  $N$  z jeho zápisu  $(S_{n-1}S_{n-2} \dots S_1S_0)_{B_z}$  (v soustavě o základu  $B_z$ )
- 2 získání zápisu  $(S_{n-1}S_{n-2} \dots S_1S_0)_{B_{na}}$  (hodnoty) čísla  $N$  (v soustavě o základu  $B_{na}$ )
  - jednodušší převod zápisu čísla v soustavě o základu  $B^k$  ( $k \in \mathbb{N}$ ) na zápis v soustavě o základu  $B$ , a naopak:

každý symbol soustavy o základu  $B^k$  zapisující nějaké číslo nahradíme  $k$ -ticí symbolů soustavy o základu  $B$  zapisující stejné číslo, a naopak ( $k$ -tice symbolů v zápisu brány zprava, chybějící symboly nahrazeny 0)

pro  $B = 2, k = 4, 3, 2, 1$ :

$$(4CD)_{16} = (2315)_8 = (103031)_4 = (010011001101)_2$$

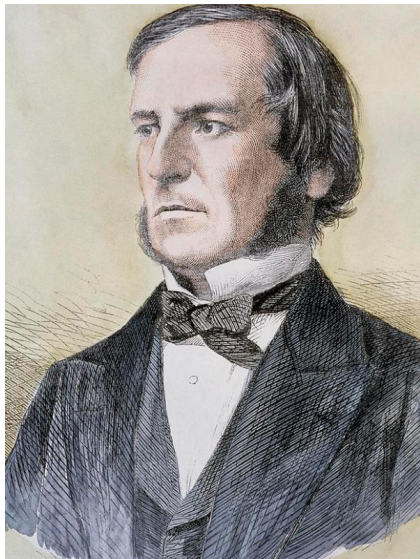
- 1 Pro několik čísel zjistěte (hodnotu) čísla z jeho zápisů ve dvojkové, osmičkové, desítkové a šestnáctkové soustavě.
- 2 Pro několik čísel zjistěte zápis (hodnoty) čísla ve dvojkové, osmičkové, desítkové a šestnáctkové soustavě.
- 3 Pro několik čísel převedte zápis čísla mezi dvojkovou, osmičkovou a šestnáctkovou soustavou.



# Binární logika

Počítač = digitální zařízení (2 stabilní stavy součástí) . . . základní operace dvouhodnotové  
⇒ binární logika

- formální logický základ: **výroková logika** – zkoumá pravdivostní hodnotu výroků (pravda/nepravda, spojky/operátory “neplatí, že” → operace negace  $\neg$ , “a současně platí” → konjunkce  $\wedge$ , “nebo platí” → disjunkce  $\vee$ , “jestliže platí, pak platí” → implikace  $\Rightarrow$  aj.)
- výroky jako **logické výrazy** vyhodnocované na hodnoty (pravda/nepravda, 1/0)
- matematický aparát pro práci s dvouhodnotovými log. výrazy: **Booleova algebra (binární/dvouhodnotová logika)** (G. Boole, 1854) – použití i u množin
- fyzická realizace: (elektronické binární) **logické obvody** (C. E. Shannon, 1937) – základ digitálních zařízení
- univerzální, teoreticky zvládnutá, efektivně realizovatelná logickými obvody



George Boole, zdroj





Claude Elwood Shannon, [zdroj](#)

## Logická proměnná $x$

- veličina nabývající dvou možných diskrétních logických hodnot:  $0$  (nepravda) a  $1$  (pravda)
- definice:  $x = 1$  jestliže  $x \neq 0$  a  $x = 0$  jestliže  $x \neq 1$

## Logická funkce $f(x_1, \dots, x_n)$

- funkce  $n$  logických proměnných  $x_1, \dots, x_n$  ( $= n$ -ární funkce) nabývající dvou možných diskrétních hodnot  $0$  (nepravda) a  $1$  (pravda)
- logická proměnná = logická funkce identity proměnné, skládání funkcí
- základní = **logické operace**

## Logická proměnná $x$

- veličina nabývající dvou možných diskrétních logických hodnot:  $0$  (nepravda) a  $1$  (pravda)
- definice:  $x = 1$  jestliže  $x \neq 0$  a  $x = 0$  jestliže  $x \neq 1$

## Logická funkce $f(x_1, \dots, x_n)$

- funkce  $n$  logických proměnných  $x_1, \dots, x_n$  ( $= n$ -ární funkce) nabývající dvou možných diskrétních hodnot  $0$  (nepravda) a  $1$  (pravda)
- logická proměnná = logická funkce identity proměnné, skládání funkcí
- základní = **logické operace**

## Booleova algebra (binární logika)

- algebra („matematika“) logických proměnných a logických funkcí
- dvouhodnotová algebra, algebra dvou stavů
- relace rovnosti:  $f = g$ , právě když ( $f = 1$  a  $g = 1$ ) nebo ( $f = 0$  a  $g = 0$ )

3 základní:

## Negace (inverze)

- pravdivá, když operand nepravdivý, jinak nepravdivá

$x$	$\bar{x}$
<b>0</b>	<b>1</b>
<b>1</b>	<b>0</b>

- operátory:  $\bar{x}$ , NOT  $x$ ,  $\neg x$  (výrokově negace, algebraicky negace),  $\bar{X}$  (množinově doplněk)

## Logický součin (konjunkce)

- pravdivá, když oba operandy pravdivé, jinak nepravdivá

$x$	$y$	$x \cdot y$
<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>1</b>

- operátory:  $x \cdot y / xy$  (prázdný),  $x$  AND  $y$ ,  $x \wedge y$  (výrokově konjunkce, algebraický průsek),  $X \cap Y$  (množinově průnik)

## Logický součet (disjunkce)

- nepravdivá, když oba operandy nepravdivé, jinak pravdivá

$x$	$y$	$x + y$
<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>1</b>

- operátory:  $x + y$ ,  $x \text{ OR } y$ ,  $x \vee y$  (výrokově disjunkce, algebraicky spojení),  $X \cup Y$  (množinově sjednocení)

## Logický výraz

= korektně vytvořená posloupnost (symbolů) logických proměnných a funkcí (operátorů) spolu se závorkami

- priority sestupně: negace, log. součin, log. součet

- např.  $x \cdot \bar{y} + f(x, z) = (x \cdot (\bar{y})) + f(x, z)$

= zápis logické funkce

## Logická rovnice

= dva logické výrazy v relaci rovnosti =

- ekvivalentní úpravy = zachování (pravdivosti) rovnosti výrazů: např. negace obou stran, logický součin/součet obou stran se stejným výrazem, ..., log. funkce obou stran se stejnými ostatními operandy funkce

- NEkvivalentní úpravy: “krácení” obou stran o stejný (pod)výraz, např.  $x + y = x + z$  na  $y = z$

## Axiomy (Booleovy algebry)

### ■ komutativita:

$$x \cdot y = y \cdot x \quad x + y = y + x$$

### ■ distributivita:

$$x \cdot (y + z) = x \cdot y + x \cdot z \quad x + y \cdot z = (x + y) \cdot (x + z)$$

### ■ identita/neutrálnost (existence neutrální hodnoty):

$$\mathbf{I} \cdot x = x \quad \mathbf{0} + x = x$$

### ■ komplementárnost:

$$x \cdot \bar{x} = \mathbf{0} \quad x + \bar{x} = \mathbf{I}$$



## Vlastnosti základních logických operací

### ■ agresivita (nuly a jedničky):

$$\mathbf{0} \cdot x = \mathbf{0} \quad \mathbf{1} + x = \mathbf{1}$$

### ■ idempotence:

$$x \cdot x = x \quad x + x = x$$

### ■ asociativita:

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \quad x + (y + z) = (x + y) + z$$

### ■ involuce (dvojitá negace):

$$\overline{\overline{x}} = x$$

### ■ De Morganovy zákony:

$$\overline{x \cdot y} = \overline{x} + \overline{y} \quad \overline{x + y} = \overline{x} \cdot \overline{y}$$

### ■ absorpce:

$$x \cdot (x + y) = x \quad x + x \cdot y = x$$

### ■ a další



## Vlastnosti základních logických operací – použití

- důkazy: s využitím axiomů a již dokázaných vlastností, rozбором případů (přiřazením všech možných kombinací hodnot **0** a **1** proměnným)
- ekvivalentní úpravy logických výrazů – pro jejich zjednodušení
- ...

### Implikace

- nepravdivá, když první operand pravdivý a druhý nepravdivý, jinak pravdivá

$x$	$y$	$x \rightarrow y$
<b>0</b>	<b>0</b>	<b>I</b>
<b>0</b>	<b>I</b>	<b>I</b>
<b>I</b>	<b>0</b>	<b>0</b>
<b>I</b>	<b>I</b>	<b>I</b>

- operátory:  $x \rightarrow y$ ,  $x \rightarrow y$  (výrokově i algebraicky implikace),  $X \subseteq Y$  (množinově podmnožina)

## Ekvivalence

- pravdivá, když operandy mají stejnou hodnotu, jinak nepravdivá

$x$	$y$	$x \equiv y$
<b>0</b>	<b>0</b>	<b>I</b>
<b>0</b>	<b>I</b>	<b>0</b>
<b>I</b>	<b>0</b>	<b>0</b>
<b>I</b>	<b>I</b>	<b>I</b>

- operátory:  $x \equiv y$ ,  $x \text{ XNOR } y$ ,  $x \equiv y$  (výrokově i algebraicky ekvivalence),  $X \equiv Y$  (množinově ekvivalence nebo rovnost)

## Nonekvivalence (negace ekvivalence, aritmetický součet modulo 2)

- pravdivá, když operandy mají různou hodnotu, jinak nepravdivá

$x$	$y$	$x \oplus y$
<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>0</b>

- operátory:  $x \oplus y$ ,  $x$  XOR  $y$ ,  $x \not\equiv y$  (výrokově i algebraicky negace ekvivalence),  $X \not\equiv Y$  (množinově negace ekvivalence)

## Shefferova funkce (negace logického součinu)

- nepravdivá, když oba operandy pravdivé, jinak pravdivá

$x$	$y$	$x \uparrow y$
<b>0</b>	<b>0</b>	<b>1</b>
<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>0</b>

- operátory:  $x \uparrow y$ ,  $x \text{ NAND } y$

## Piercova funkce (negace logického součtu)

- pravdivá, když oba operandy nepravdivé, jinak nepravdivá

$x$	$y$	$x \downarrow y$
<b>0</b>	<b>0</b>	<b>1</b>
<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>

- operátory:  $x \downarrow y$ ,  $x \text{ NOR } y$

## ■ zadání **pravdivostní tabulkou**:

- úplně – funkční hodnota  $f(x_i)$  definována pro všech  $2^n$  možných přiřazení hodnot proměnným  $x_i, 0 \leq i < n$
- neúplně – funkční hodnota pro některá přiřazení není definována

## ■ **základní tvary** (výrazu):

- **součinný (úplná konjunktivní normální forma, ÚKNF)** – log. součin log. součtů všech proměnných nebo jejich negací (úplných elementárních disjunkcí, ÚED)

$$(X_0 + \dots + X_{n-1}) \cdot \dots \cdot (X_0 + \dots + X_{n-1}) \quad X_i = x_i \text{ nebo } \bar{x}_i \text{ (literál)}$$

- **součtový (úplná disjunktivní normální forma, ÚDNF)** – log. součet log. součinů všech proměnných nebo jejich negací (úplných elementárních konjunkcí, ÚEK)

$$(X_0 \cdot \dots \cdot X_{n-1}) + \dots + (X_0 \cdot \dots \cdot X_{n-1}) \quad X_i = x_i \text{ nebo } \bar{x}_i$$



## Převod log. funkce $f(x_i)$ na základní tvar (normální formu)

- ekvivalentními úpravami a doplněním chybějících proměnných nebo jejich negací
- **tabulkovou metodou:**
  - 1 řádky pro všechna možná přiřazení hodnot všem proměnným  $x_i$  funkce ( $2^n$  pro  $0 \leq i < n$ )
  - 2 pro řádky s  $f(x_i) = \mathbf{0/I}$  sestroj log. součet/součin všech  $x_i$  pro  $x_i = \mathbf{0/I}$  nebo  $\bar{x}_i$  pro  $x_i = \mathbf{I/0}$
  - 3 výsledná ÚKNF/ÚDNF je log. součinem/součtem těchto log. součtů/součinů (ÚED/ÚEK)

$x$	$y$	$z$	$f(x, y, z)$	ÚED	ÚEK
0	0	0	0	$x + y + z$	
0	0	1	0	$x + y + \bar{z}$	
0	1	0	0	$x + \bar{y} + z$	
0	1	1	1		$\bar{x} \cdot y \cdot z$
1	0	0	0	$\bar{x} + y + z$	
1	0	1	1		$x \cdot \bar{y} \cdot z$
1	1	0	1		$x \cdot y \cdot \bar{z}$
1	1	1	1		$x \cdot y \cdot z$

$$\text{ÚKNF}(f(x, y, z)): (x + y + z) \cdot (x + y + \bar{z}) \cdot (x + \bar{y} + z) \cdot (\bar{x} + y + z)$$

$$\text{ÚDNF}(f(x, y, z)): \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z} + x \cdot y \cdot z$$



Převeďte několik log. funkcí se třemi a více proměnnými do ÚKNF a ÚDNF.

## Zjednodušení výrazu logické funkce

= optimalizace za účelem dosažení co nejmenšího počtu operátorů (v kompromisu s min. počtem druhů operátorů)

### Algebraická minimalizace

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

// dvakrát přičteme  $x \cdot y \cdot z$  (idempotence)

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

// distributivita

$$f = y \cdot z \cdot (\bar{x} + x) + x \cdot z \cdot (\bar{y} + y) + x \cdot y \cdot (\bar{z} + z) // \text{komplementárnost}$$

$$f = x \cdot y + y \cdot z + x \cdot z$$

- pro složitější výrazy intelektuálně náročná (podobně jako důkaz) – kdy jaké ekvivalentní úpravy?

## Zjednodušení výrazu logické funkce

### Karnaughova metoda (Veitch diagram)

- nahrazení algebraických ekvivalentních úprav algoritmickými geometrickými postupy
  - nalezení minimálního výrazu funkce v (neúplném) součtovém tvaru
- 1 pro  $n$  proměnných funkce tabulka s  $2^n$  buňkami, v maximálně stejném počtu řádků a sloupců, reprezentujícími všechny možné log. součiny (ÚEK) základního součtového tvaru (ÚDNF), součiny reprezentované sousedními buňkami se liší právě v jednom literálu
  - 2 pro výraz funkce v ÚDNF tzv. **Karnaughova mapa (K-mapa)** = vyplnění tabulky **I** v buňkách reprezentujících ÚEK
  - 3 nalezení minimálního počtu skupin buněk v mapě, tvořících maximální obdélníkové oblasti buněk obsahujících pouze **I** v počtu mocniny 2 ( $i$  jedna), a pokrývajících všechny **I** v mapě (skupiny se mohou překrývat a krajní buňky jsou také „sousední“ v oblasti)
  - 4 součiny, reprezentované buňkami ve skupinách, po vyřazení proměnných vyskytujících se i s jejich negací = log. součiny výsledného (neúplného) součtového tvaru

## Zjednodušení výrazu logické funkce

### Karnaughova metoda (Veitch diagram)

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

	$\bar{x} \cdot \bar{y}$	$\bar{x} \cdot y$	$x \cdot y$	$x \cdot \bar{y}$
$\bar{z}$			1	
$z$		1	1	1

Obrázek: Karnaughova mapa

$$f = x \cdot y + y \cdot z + x \cdot z$$

- pro složitější výrazy (funkcí více proměnných) výpočetně náročná – hledání skupin
- Další algoritmické metody: tabulační (Quine-McCluskey), branch-and-bound (Petrick), Espresso logic minimizer aj.

# ÚKOL



Pokuste se minimalizovat log. funkce z přechozího úkolu.



## Věta (O počtu log. funkcí)

*Existuje právě  $2^{(2^n)}$  logických funkcí s  $n$  proměnnými ( $n$ -árních).*

## Věta (O počtu log. funkcí)

Existuje právě  $2^{(2^n)}$  logických funkcí s  $n$  proměnnými ( $n$ -árních).

Funkce  $f^0$  žádné proměnné  
(konstantní, nulární)

$f_0$	$f_1$
<b>0</b>	<b>I</b>

Funkce  $f^1$  jedné proměnné (unární)

$x$	$f_0$	$f_1$	$f_2$	$f_3$
	<b>0</b>	$x$	$\bar{x}$	<b>I</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>
<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>

Funkce  $f^2$  dvou proměnných (binární)

$x$	$y$	$f_0$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$
		<b>0</b>	$\cdot$		$x$		$y$	$\oplus$	$+$	$\downarrow$	$\equiv$	$\bar{y}$		$\bar{x}$	$\rightarrow$	$\uparrow$	<b>I</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>
<b>0</b>	<b>I</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>
<b>I</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>	<b>0</b>	<b>0</b>	<b>I</b>	<b>I</b>
<b>I</b>	<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>	<b>0</b>	<b>I</b>



## Funkce více než dvou proměnných

pro  $n = 3$  (ternární funkce):

$$\begin{aligned}f(x, y, z) &= x \cdot f(\mathbf{1}, y, z) + \bar{x} \cdot f(\mathbf{0}, y, z) \\ &= (\bar{x} + f(\mathbf{1}, y, z)) \cdot (x + f(\mathbf{0}, y, z))\end{aligned}$$

a podobně pro  $n > 3$

## Funkce více než dvou proměnných

pro  $n = 3$  (ternární funkce):

$$\begin{aligned}f(x, y, z) &= x \cdot f(\mathbf{1}, y, z) + \bar{x} \cdot f(\mathbf{0}, y, z) \\ &= (\bar{x} + f(\mathbf{1}, y, z)) \cdot (x + f(\mathbf{0}, y, z))\end{aligned}$$

a podobně pro  $n > 3$

### Věta (O reprezentaci log. funkcí, Shannonův expanzní teorém)

*Jakoukoliv logickou funkci libovolného počtu proměnných lze vyjádřit pomocí logických funkcí dvou proměnných (binárních, např. logických operací).*

## Úplný systém logických funkcí

- = množina log. funkcí, pomocí kterých je možné vyjádřit jakoukoliv log. funkci (libovolného počtu proměnných)
- množina binárních log. funkcí (Věta o reprezentaci log. funkcí)
  - (1) negace  $\bar{x}$ , log. součin  $x \cdot y$  a log. součet  $x + y$
  - (2) negace  $\bar{x}$  a implikace  $x \rightarrow y$
  - a další

## Úplný systém logických funkcí

= množina log. funkcí, pomocí kterých je možné vyjádřit jakoukoliv log. funkci (libovolného počtu proměnných)

→ množina binárních log. funkcí (Věta o reprezentaci log. funkcí)

- (1) negace  $\bar{x}$ , log. součin  $x \cdot y$  a log. součet  $x + y$
- (2) negace  $\bar{x}$  a implikace  $x \rightarrow y$
- a další

## Minimální úplný systém logických funkcí

= úplný systém, ze kterého nelze žádnou funkci vyjmout tak, aby zůstal úplný

- (1) NENÍ
- (2) JE
- (3) negace  $\bar{x}$  a log. součin  $x \cdot y$
- (4) negace  $\bar{x}$  a log. součet  $x + y$
- a další

## Minimální úplný systém logických funkcí

Jediná funkce:

- **Shefferova**  $\uparrow$  (negace log. součinu)
- **Piercova**  $\downarrow$  (negace log. součtu)

## Minimální úplný systém logických funkcí

Jediná funkce:

- **Shefferova**  $\uparrow$  (negace log. součinu)
- **Piercova**  $\downarrow$  (negace log. součtu)

## Vyjádření logické funkce pomocí Shefferovy nebo Piercovy funkce

- 1 vyjádření funkce v základním součtovém tvaru (ÚDNF)
- 2 zjednodušení ÚDNF funkce, např. pomocí Karnaughovy metody
- 3 aplikace De Morganových zákonů, involuce a idempotence pro úpravu výrazu do tvaru, který obsahuje pouze Shefferovy nebo pouze Piercovy funkce

## Vyjádření logické funkce pomocí Shefferovy nebo Piercovy funkce

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

$$f = x \cdot y + y \cdot z + x \cdot z$$

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

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

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

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

$$f = (x + y) \cdot (y + z) \cdot (x + z)$$

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

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

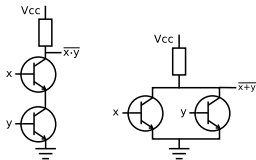
$$f = \overline{\overline{\overline{\overline{\overline{\overline{x + y + \overline{y + z} + \overline{x + y + \overline{y + z} + \overline{x + z}}}}}}}}$$



Vyjádřete log. operace negace, log. součin, log. součet, implikace, ekvivalence a nonekvivalence pomocí (1) Shefferovy funkce a (2) Piercovy funkce.

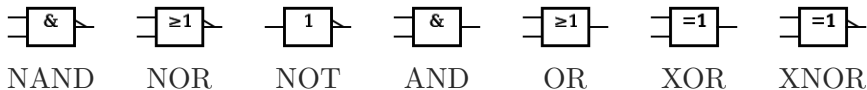


- dříve pomocí **spínačích relé** a **elektronek**, plus pasivní součástky (rezistor aj.)
- dnes pomocí **tranzistorů** (a diod a pasivních součástek) v **integrovaných obvodech**: technologie RTL, DTL, **TTL**, **CMOS**, **MOSFET** aj.

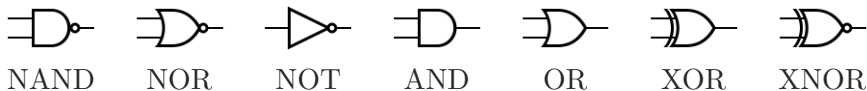


Obrázek: Příklad realizace log. operací NAND a NOR (v rezistorovětranzistorové logice, RTL)

- **logický člen, hradlo** = realizace log. operace (pomocí integrovaného obvodu)
  - (binární) vstupy  $\sim$  log. proměnné – i více než dvě (rozšíření binárních operací)
  - (binární) výstup  $\sim$  výsledek log. operace (funkční hodnota)
  - stavy (signály) na vstupech/výstupu = log. (binární) hodnoty **0/I** – míra informace s jednotkou **1 bit**
- **logický obvod** = realizace (složitější) log. funkce nebo více funkcí současně – symbolické značky log. členů ve schématech zapojení obvodu

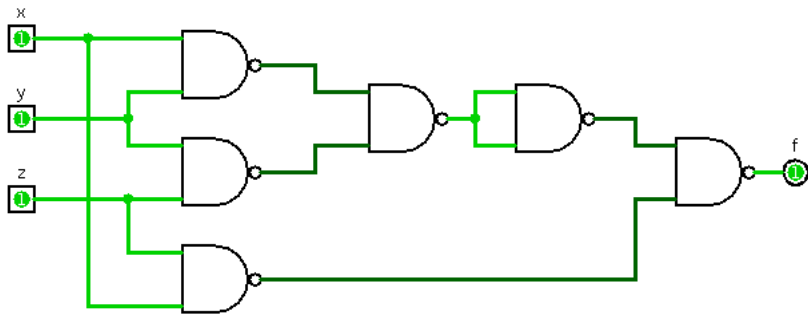


Obrázek: Symbolické značky logických členů (IEC)



Obrázek: Symbolické značky logických členů (tradiční, ANSI)

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



**Obrázek:** Schéma zapojení log. obvodu realizujícího log. funkci  $f$  pomocí log. členů realizujících log. operaci NAND



Nakreslete schéma zapojení log. obvodu realizujícího log. operace NOT, AND, OR, implikace, ekvivalence a XOR pomocí log. členů realizujících operaci (1) NAND a (2) NOR.



- jeden výstup = realizace jedné log. funkce
- více výstupů = realizace více log. funkcí současně → realizace **vícebitové log. funkce**  
 $m_f$
- $n$ -tice vstupů  $\sim$  **vícebitové ( $n$ -bitové) log. proměnné**  ${}^n\mathbf{x} = x_{n-1} \dots x_i \dots x_0 \rightarrow$   
**vícebitový ( $n$ -bitový) log. obvod**

- jeden výstup = realizace jedné log. funkce
- více výstupů = realizace více log. funkcí současně → realizace **vícebitové log. funkce**  
 $m_f$
- $n$ -tice vstupů  $\sim$  **vícebitové ( $n$ -bitové) log. proměnné**  ${}^n\mathbf{x} = x_{n-1} \dots x_i \dots x_0 \rightarrow$   
**vícebitový ( $n$ -bitový) log. obvod**
  
- **kombinační**: stavy na výstupech obvodu (funkční hodnoty) závisí pouze na okamžitých stavech na jeho vstupech (hodnotách proměnných)
- **sekvenční**: stavy na výstupech obvodu (funkční hodnoty) závisí nejen na okamžitých stavech na jeho vstupech (hodnotách proměnných), ale i na předchozích stavech na vstupech

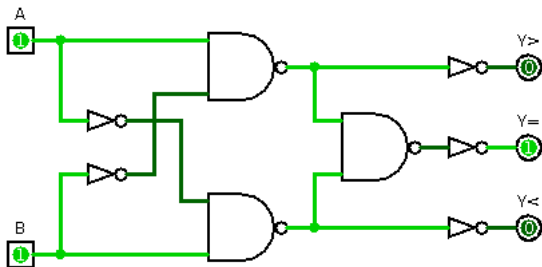
- stavy na výstupech obvodu (funkční hodnoty) závisí pouze na okamžitých stavech na jeho vstupech (hodnotách proměnných)
- jedné kombinaci stavů na vstupech odpovídá jediná kombinace stavů na výstupech

- srovnává dvě hodnoty  $A$  a  $B$  na vstupech
- tři výstupy udávající pravdivost vztahů  $A < B$ ,  $A > B$  a  $A = B \Rightarrow$  tříbitová funkce  $Y_{<} = Y(A < B)$ ,  $Y_{>} = Y(A > B)$ ,  $Y_{=} = Y(A = B)$ :

$$Y_{<} = \bar{A} \cdot B \quad Y_{>} = A \cdot \bar{B} \quad Y_{=} = A \cdot B + \bar{A} \cdot \bar{B}$$

$$Y_{<} = \overline{\bar{A} \cdot B} \quad Y_{>} = \overline{A \cdot \bar{B}} \quad Y_{=} = \overline{\overline{A \cdot B} \cdot \overline{\bar{A} \cdot \bar{B}}}$$

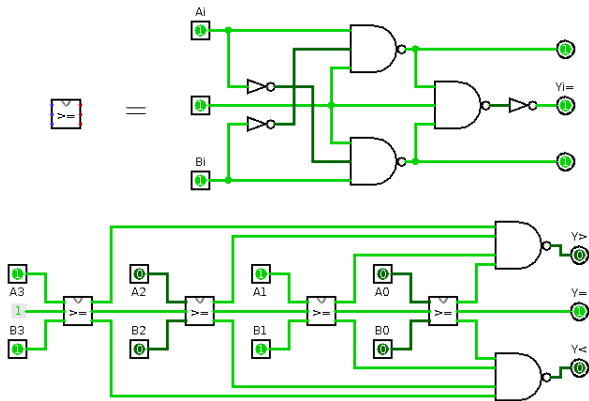
$A$	$B$	$Y_{<}$	$Y_{>}$	$Y_{=}$
0	0	0	0	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1



Obrázek: Pravdivostní tabulka a schéma zapojení (jednobitového) komparátoru



- vícebitový: zřetěžené zapojení jednobitových pro každý řád  $i$  vícebitových ( $n$ -bitových) hodnot  $A = A_{n-1} \dots A_i \dots A_0$  a  $B = B_{n-1} \dots B_i \dots B_0$  od nejvýznamnějšího  $n - 1$  po nejméně významný  $0$

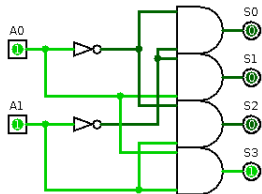


Obrázek: Schéma zapojení čtyřbitového komparátoru

- nastaví (na **I**) jeden z  $2^n$  výstupů  $S_i$  odpovídající  $n$ -bitové hodnotě na adresním (řídícím) vstupu  $A$
- např. dvoubitový (dvoubitový adresní vstup, 4 výstupy)  $\Rightarrow$  čtyřbitová funkce  $S_0 = S(A = 00), S_1 = S(A = 10), S_2 = S(A = 01), S_3 = S(A = 11)$ :

$$S_0 = \overline{A_0} \cdot \overline{A_1} \quad S_1 = A_0 \cdot \overline{A_1} \quad S_2 = \overline{A_0} \cdot A_1 \quad S_3 = A_0 \cdot A_1$$

$A_0$	$A_1$	$S_0$	$S_1$	$S_2$	$S_3$
0	0	1	0	0	0
1	0	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1



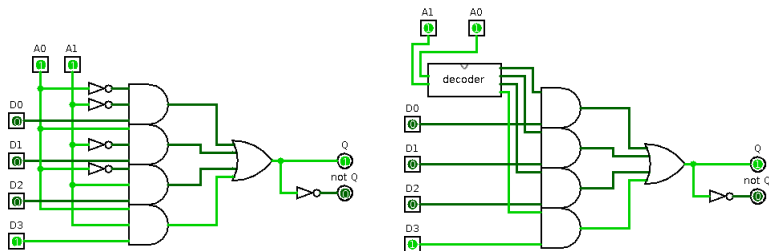
**Obrázek:** Pravdivostní tabulka a schéma zapojení dvoubitového dekodéru

- použití: dekódování adresy (řídící hodnoty) pro výběr např. místa v paměti (části obvodu)

- předává na výstup  $Q$  hodnotu na jednom z  $2^n$  datových vstupů  $D_j$  dle  $n$ -bitové hodnoty na adresním (řídícím) vstupu  $A$
- vedle výstupu  $Q$  obvykle ještě negovaný (invertovaný) výstup  $\overline{Q}$
- např. dvoubitový (4 datové vstupy, dvoubitový adresní vstup) realizuje funkci

$$Q = \overline{A_0} \cdot \overline{A_1} \cdot D_0 + A_0 \cdot \overline{A_1} \cdot D_1 + \overline{A_0} \cdot A_1 \cdot D_2 + A_0 \cdot A_1 \cdot D_3$$

$A_0$	$A_1$	$Q$
0	0	$D_0$
1	0	$D_1$
0	1	$D_2$
1	1	$D_3$



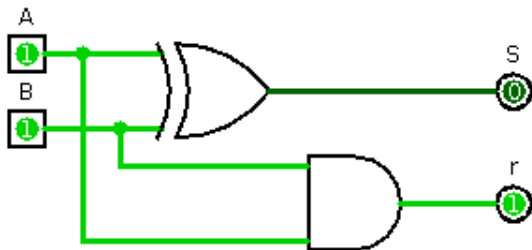
Obrázek: Pravdivostní tabulka a schéma zapojení dvoubitového multiplexoru

- použití: výběr (multiplexování) datových vstupů na základě adresy (řídící hodnoty)

- (aritmeticky) sčítá dvě hodnoty  $A$  a  $B$  (plus přenos  $r_{i-1}$  z nižšího řádu) na vstupech, s přenosem  $r_i$  do vyššího řádu
- dva výstupy, pro součet  $S$  (aritmetický modulo 2) a přenos  $r \Rightarrow$  dvoubitová funkce
- **poloviční sčítačka (half adder)** = bez přenosu z nižšího řádu:

$$S = A \oplus B \quad r = A \cdot B$$

$A$	$B$	$S$	$r$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

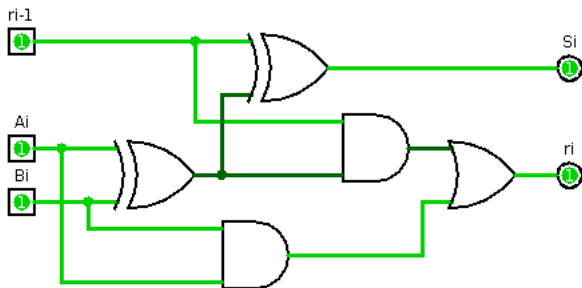


**Obrázek:** Pravdivostní tabulka a schéma zapojení (jednobitové) poloviční sčítačky

- **plná sčítačka (full adder)** = s přenosem  $r_{i-1}$  z nižšího řádu (součet  $S_i$  v řádu  $i$  a přenos  $r_i$  do vyššího řádu):

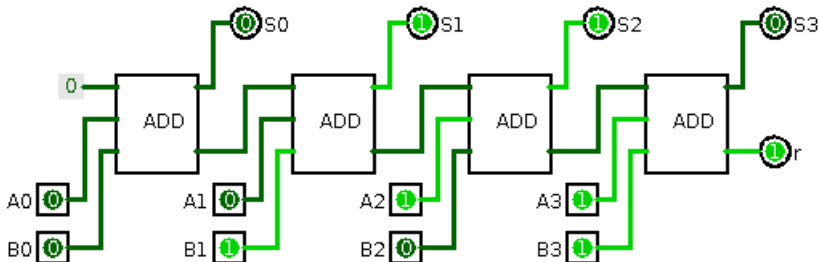
$$S_i = A_i \oplus B_i \oplus r_{i-1} \quad r_i = A_i \cdot B_i + (A_i \oplus B_i) \cdot r_{i-1} \quad (r_{-1} = 0)$$

$A_i$	$B_i$	$r_{i-1}$	$S_i$	$r_i$
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1



Obrázek: Pravdivostní tabulka a schéma zapojení (jednobitové plné) sčítačky (pro řád  $i$ )

- vícebitová (ripple-carry adder): zřetěžené zapojení jednobitových pro každý řád  $i$  vícebitových ( $n$ -bitových) hodnot  $A = A_{n-1} \dots A_i \dots A_0$  a  $B = B_{n-1} \dots B_i \dots B_0$  od nejméně významného 0 po nejvýznamnější  $n - 1$
- použití: (aritmetický) součet hodnot  $A$  a  $B$  (binárně reprezentovaných čísel = ve dvojkové soustavě), s přenosem do vyššího řádu



Obrázek: Schéma zapojení čtyřbitové sčítačky



- (aritmeticky) násobí dvě hodnoty  $A$  a  $B$  na vstupech:  $M = A \cdot B$  (jednabitově)

- (aritmeticky) násobí dvě hodnoty  $A$  a  $B$  na vstupech:  $M = A \cdot B$  (jednobitově)
- vícebitová: zapojení jednobitových násobiček a sčítaček pro dvojnásobný počet řádů vícebitových ( $n$ -bitových) hodnot  $A = A_{n-1} \dots A_i \dots A_0$  a  $B = B_{n-1} \dots B_i \dots B_0$ :

$$\begin{aligned} M &= (A_{n-1} \cdot 2^{n-1} + \dots + A_1 \cdot 2 + A_0) \cdot (B_{n-1} \cdot 2^{n-1} + \dots + B_1 \cdot 2 + B_0) \\ &= A_{n-1} \cdot B_{n-1} \cdot 2^{2(n-1)} + \dots + (A_{n-1} \cdot B_1 + \dots + A_1 \cdot B_{n-1}) \cdot 2^{(n-1)+1} + \\ &\quad (A_{n-1} \cdot B_0 + \dots + A_0 \cdot B_{n-1}) \cdot 2^{n-1} + \dots + (A_1 \cdot B_1 + \dots) \cdot 2^{1+1} + (A_1 \cdot B_0 + A_0 \cdot B_1) \cdot 2 + A_0 \cdot B_0 \end{aligned}$$

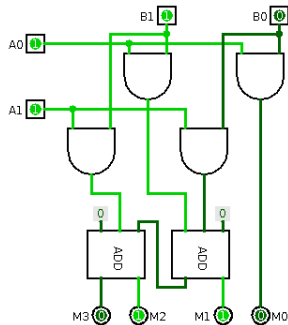


- (aritmeticky) násobí dvě hodnoty  $A$  a  $B$  na vstupech:  $M = A \cdot B$  (jednobitově)
- vícebitová: zapojení jednobitových násobiček a sčítaček pro dvojnásobný počet řádů vícebitových ( $n$ -bitových) hodnot  $A = A_{n-1} \dots A_i \dots A_0$  a  $B = B_{n-1} \dots B_i \dots B_0$ :

$$\begin{aligned}
 M &= (A_{n-1} \cdot 2^{n-1} + \dots + A_1 \cdot 2 + A_0) \cdot (B_{n-1} \cdot 2^{n-1} + \dots + B_1 \cdot 2 + B_0) \\
 &= A_{n-1} \cdot B_{n-1} \cdot 2^{2(n-1)} + \dots + (A_{n-1} \cdot B_1 + \dots + A_1 \cdot B_{n-1}) \cdot 2^{(n-1)+1} + \\
 &\quad (A_{n-1} \cdot B_0 + \dots + A_0 \cdot B_{n-1}) \cdot 2^{n-1} + \dots + (A_1 \cdot B_1 + \dots) \cdot 2^{1+1} + (A_1 \cdot B_0 + A_0 \cdot B_1) \cdot 2 + A_0 \cdot B_0
 \end{aligned}$$

(+ značí aritmetické sčítání)

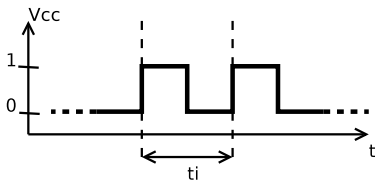
		$A_1 A_0$
	·	$B_1 B_0$
	$A_1 \cdot B_0$	$A_0 \cdot B_0$
+	$A_1 \cdot B_1$	$A_0 \cdot B_1$
	$A_1 \cdot B_1$	$A_1 \cdot B_0 + A_0 \cdot B_1$
		$A_0 \cdot B_0$



Obrázek: Schéma dvoubitového násobení a zapojení dvoubitové násobičky

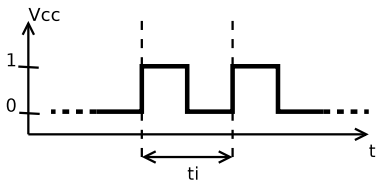
Nakreslete schéma zapojení log. obvodu představujícího „základ“ aritmeticko-logické jednotky (ALU) procesoru, realizující aritmetické operace součtu a násobení a log. operace NAND a NOR dvou dvoubitových hodnot (binárně reprezentovaných čísel) na datových vstupech obvodu, kde prováděná operace je určena dvoubitovou hodnotou na řídicím vstupu obvodu.

- stavy na výstupech obvodu (funkční hodnoty) závisí nejen na okamžitých stavech na jeho vstupech (hodnotách proměnných), ale i na předchozích stavech na vstupech – zachyceny **vnitřním stavem obvodu**
- nutné identifikovat a synchronizovat stavy obvodu v čase
- čas: periodický impulzní signál = „hodiny“ (clock), diskrétně určující okamžiky synchronizace obvodu



Obrázek: Časový signál „hodin“ (clock)

- stavy na výstupech obvodu (funkční hodnoty) závisí nejen na okamžitých stavech na jeho vstupech (hodnotách proměnných), ale i na předchozích stavech na vstupech – zachyceny **vnitřním stavem obvodu**
- nutné identifikovat a synchronizovat stavy obvodu v čase
- čas: periodický impulzní signál = „hodiny“ (clock), diskrétně určující okamžiky synchronizace obvodu

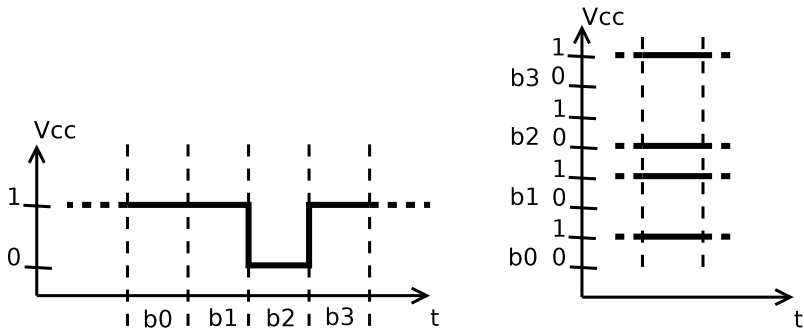


Obrázek: Časový signál „hodin“ (clock)

→ zpětné vazby z (některých) výstupů na (některé) vstupy

Přenos dat (vícebitových hodnot):

- **sériový**: hodnoty **0/I** (bity) postupně v čase za sebou, po jednom vodiči
- **paralelní**: bity zároveň v čase, po více vodičích
- úlohy transformace mezi sériovým a paralelním přenosem



Obrázek: Sériový a paralelní přenos dat

- nejjednodušší sekvenční obvody
- **astabilní**: žádný stabilní stav, periodické překlápění výstupů z jednoho stavu do druhého („kmitání“); použití jako generátory impulzů
- **monostabilní**: jeden stabilní stav na výstupech, po určitém řídicím signálu po definované dobu v nestabilním stavu; použití k vytváření impulzů dané délky
- **bistabilní**: oba stavy na výstupech stabilní, trvání jednoho dokud není určitým řídicím signálem překlopení do druhého; použití pro realizaci **paměti**

- nejjednodušší sekvenční obvody
- **astabilní**: žádný stabilní stav, periodické překlápění výstupů z jednoho stavu do druhého („kmitání“); použití jako generátory impulzů
- **monostabilní**: jeden stabilní stav na výstupech, po určitém řídicím signálu po definované dobu v nestabilním stavu; použití k vytváření impulzů dané délky
- **bistabilní**: oba stavy na výstupech stabilní, trvání jednoho dokud není určitým řídicím signálem překlopení do druhého; použití pro realizaci **paměti**

Řízení:

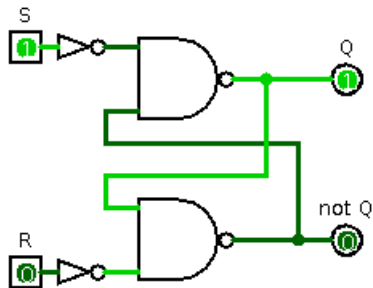
- **asynchronně**: signály/stavy (**0** nebo **1**) na (datových) vstupech
- **synchronně**: (periodickým) signálem na hodinovém vstupu
- **hladinou** signálu (latch): horní (hodnota **1**) nebo dolní (**0**)
- **hranami** signálu: nástupní (**0** → **1** u horní hladiny) nebo sestupní (**1** → **0** u dolní hladiny)

- nejjednodušší bistabilní, základ ostatních
- asynchronní vstupy  $S$  (Set) pro nastavení hodnoty (na  $\mathbf{1}$ ) a  $R$  (Reset) pro nulování hodnoty (na  $\mathbf{0}$ ) na výstupu  $Q$  (v čase  $i$ )
- vedle výstupu  $Q$  obvykle ještě negovaný (invertovaný) výstup  $\overline{Q}$
- při stavu  $S = R = \mathbf{0}$  „pamatování“ hodnoty na výstupu  $Q$  (a  $Q'$ , v čase  $i$ )
- stav  $S = R = \mathbf{1}$  „nedefinovaný“ – varianty preferující vstup  $S$  (= SR) nebo  $R$  (= RS)

$S$	$R$	$Q_i$	$\overline{Q}_i$
$\mathbf{0}$	$\mathbf{0}$	$Q_{i-1}$	$\overline{Q}_{i-1}$
$\mathbf{0}$	$\mathbf{1}$	$\mathbf{0}$	$\mathbf{1}$
$\mathbf{1}$	$\mathbf{0}$	$\mathbf{1}$	$\mathbf{0}$
$\mathbf{1}$	$\mathbf{1}$	N/A	N/A



=

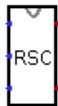


Obrázek: Pravdivostní tabulka a schéma zapojení klopného obvodu SR/RS

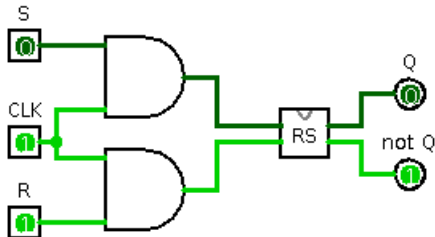


- synchronní varianta s hodinovým (synchronizačním) vstupem  $CLK$  – „funkční“ pouze při  $CLK = 1$ , jinak „pamatování“ hodnoty na výstupu  $Q$  (a  $Q'$ , v čase  $i$ ):

$S$	$R$	$CLK$	$Q_i$	$\overline{Q}_i$
0	0	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
0	1	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	0	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	1	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
0	0	1	$Q_{i-1}$	$\overline{Q}_{i-1}$
0	1	1	0	1
1	0	1	1	0
1	1	1	N/A	N/A



=

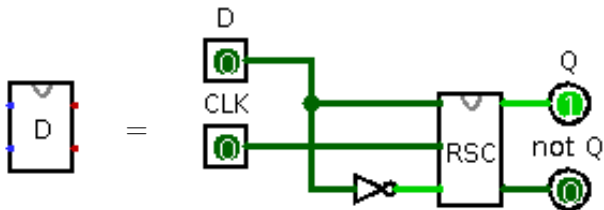


**Obrázek:** Pravdivostní tabulka a schéma zapojení klopného obvodu SR/RS s hodinovým vstupem  $CLK$

- varianta Master-Slave: dva obvody SR/RS (s hodinovým vstupem, u druhého obvodu negovaný) za sebou, nastavení na vzestupnou hranu na hodinovém vstupu, výstup na sestupní

- stav  $S = R = \mathbf{I}$  u obvodu SR/RS nemůže nastat
- **jednabitový paměťový člen**

$D$	$CLK$	$Q_i$	$\overline{Q_i}$
0	0	$Q_{i-1}$	$\overline{Q_{i-1}}$
1	0	$Q_{i-1}$	$\overline{Q_{i-1}}$
0	1	0	1
1	1	1	0

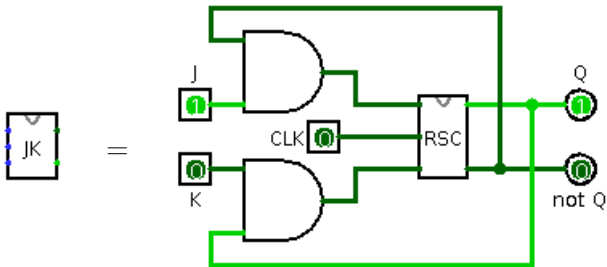


Obrázek: Pravdivostní tabulka a schéma zapojení klopného obvodu D

- varianta Master-Slave: obvody D a SR/RS (s hodinovým vstupem, u druhého obvodu negovaný) za sebou, nastavení na vzestupnou hranu na hodinovém vstupu, výstup na sestupní

- při stavu  $S = R = \mathbf{I}$  u obvodu SR/RS invertuje výstup  $Q$  (a  $Q'$ , v čase  $i$ )

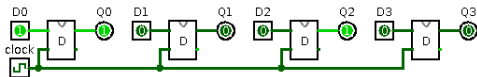
$J$	$K$	$CLK$	$Q_i$	$\overline{Q_i}$
0	0	0	$Q_{i-1}$	$\overline{Q_{i-1}}$
0	1	0	$Q_{i-1}$	$\overline{Q_{i-1}}$
1	0	0	$Q_{i-1}$	$\overline{Q_{i-1}}$
1	1	0	$Q_{i-1}$	$\overline{Q_{i-1}}$
0	0	1	$Q_{i-1}$	$\overline{Q_{i-1}}$
0	1	1	0	1
1	0	1	1	0
1	1	1	$\overline{Q_{i-1}}$	$Q_{i-1}$



**Obrázek:** Pravdivostní tabulka a schéma zapojení klopného obvodu JK

## Paralelní registr (střadač)

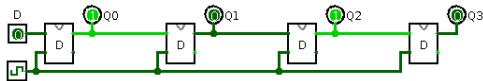
- paměť pro vícebitovou hodnotu (dodanou paralelně na více vstupů)
- paralelní zapojení klopných obvodů D



Obrázek: Schéma zapojení čtyřbitového paralelního registru

## Sériový (posuvný) registr

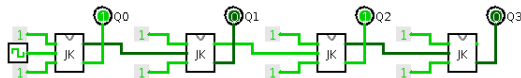
- paměť pro vícebitovou hodnotu dodanou sériově na (jednom) vstupu (synchronně po jednotlivých řádech)
- zřetěžené zapojení klopných obvodů D
- použití pro transformaci sériových dat na paralelní



Obrázek: Schéma zapojení čtyřbitového sériového registru

## Čítač

- paměť počtu impulzů na hodinovém vstupu – binárně reprezentovaný počet na vícebitovém výstupu
- zřetěžené zapojení klopných obvodů JK



Obrázek: Schéma zapojení čtyřbitového čítače

## Sériová sčítačka/násobička

- (aritmetické) sčítání/násobení hodnot dodaných sériově na vstupy (synchronně po jednotlivých řádech)
- zřetěžené zapojení sériových registrů pro hodnoty a sčítačky/násobičky

Nakreslete schéma zapojení log. obvodu realizujícího děličku frekvence signálu na hodinovém vstupu obvodu faktorem 1 (= původní frekvence), 2, 4 a 8, kde faktor je určen dvoubitovou hodnotou na řídicím vstupu obvodu.