# Maturitní témata 2020/21

## Otázka č. 2 – Hardware a aplikační software

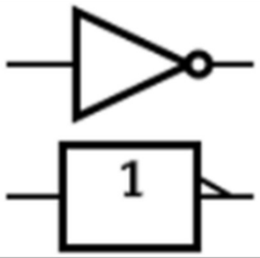
### Logická funkce:

- je funkce, která pro konečný počet vstupních parametrů vrací logické hodnoty
- v praxi se používá v mikroprocesorové technice, parametry logické funkce jsou logické proměnné
- přiřazuje-li logická funkce výstupní hodnoty všem kombinacím vstupních logických proměnných, pak se nazývá úplně zadaná logická funkce, v opačném případě se nazývá neúplné zadaná logická funkce, kombinace vstupních logických proměnných, k níž není určena hodnota výstupní logické funkce, se nazývá neurčitý stav

**NON (NOT):** logická negace (invertor) - na výstupu je vždy opačná logická hodnota než na vstupu, matematický zápis: Y = /A

| Pravdivostní tabulka | |
|---|---|
| A | Y |
| 0 | 1 |
| 1 | 0 |

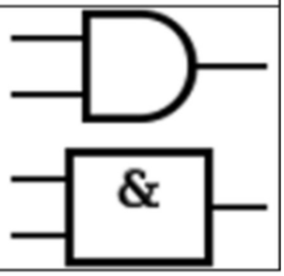
Schematická značka




**AND:** logický součin - má na výstupu log. 1 pouze tehdy, je-li na všech jeho vstupech log. 1, matematický zápis: Y= A * B

| Pravdivostní tabulka | | |
|---|---|---|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

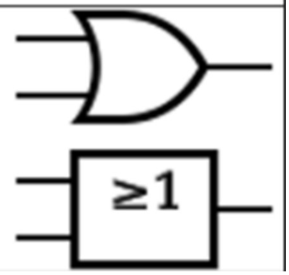
Schematická značka



**OR:** logický součet - má na výstupu log. 1 pouze tehdy, pokud je alespoň na jednom vstupu log. 1, matematický zápis: Y= A + B

| Pravdivostní tabulka | | |
|---|---|---|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

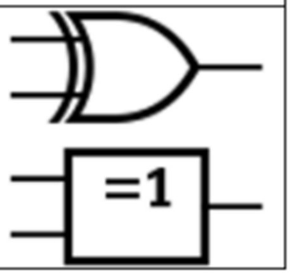
Schematická značka




**XOR:** exklusivní logický součet - má na výstupu log. 1 pouze tehdy, pokud je na vstupech rozdílná log. hodnota, matematický zápis: Y= /(A B)

| Pravdivostní tabulka | | |
|---|---|---|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Schematická značka




## Pravdivostní tabulka:

Pravdivostní tabulka je nejběžnější způsob popisu logické funkce. Popisuje zcela přesně chování logického obvodu, ale neobsahuje žádný návod pro jeho realizaci. Můžeme tedy na ni pohlížet jako na model chování logického systému. Obsahuje výčet všech kombinací vstupních proměnných a jim odpovídajících výstupů. Má-li logická funkce n nezávislých proměnných, bude mít pravdivostní tabulka 2n řádků.

| A | B | Y = A or B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## ÚNDF:

- úplná normální disjunktní forma - tu získáme z pravdivostní tabulky tak, že vytvoříme součiny vstupních proměnných v řádcích kde má výstupní funkce hodnotu y = 1 tzv. mintermy

- všechny tyto mintermy pak sečteme, každá proměnná v součinu je zapsána tak, že pokud nabývá hodnoty log. 0, pak ji píšeme s negací, pokud log. 1, pak píšeme bez negace

| a | b | c | d | y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

ÚNDF: $\cancel{\bar{a}\cdot\bar{b}\cdot\bar{c}\cdot\bar{d} + \bar{a}\,b\,\bar{c}\,\bar{d} +}$ $\bar{a}\,b\,c\,\bar{d}$

$$y = \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{a}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}bc\bar{d} + \bar{a}b\bar{c}d + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + a\bar{b}cd + ab\bar{c}d$$

## ÚNKF:

- úplná normální konjunktní forma - která se skládá ze součtů vstupních proměnných v řádcích, kde má výstupní funkce hodnotu y = 0 tzv. maxtermů, a všechny tyto maxtermy pak vynásobíme
- každá proměnná v součtu je zapsána tak, že pokud nabývá hodnoty log. 0, pak ji píšeme bez negace, pokud log. 1, pak píšeme s negací

ÚNKF:

$$y = (a+b+c+d)\cdot(a+\bar{b}+c+d)\cdot(a+\bar{b}+c+\bar{d})\cdot(a+\bar{b}+\bar{c}+\bar{d})\cdot(\bar{a}+\bar{b}+c+d)\cdot(\bar{a}+\bar{b}+\bar{c}+d)\cdot(\bar{a}+\bar{b}+\bar{c}+\bar{d})$$

## Booleova algebra:

Booleova algebra je důležitý pomocník, který slouží k minimalizaci logických funkcí pomocí zákonů a pravidel. Ne vždy se nám podaří řešit úkol, kde se objevuje nějaká jednoduchá funkce. Jakmile se 2/4 vyskytne složitější funkce, je lepší ji minimalizovat pomocí Booleovy algebry. Booleovou algebrou, jakožto jednoduchým a přesným prostředkem, popisujeme logické obvody a navrhujeme je, když známe vstupní logickou funkci

- 9 zákonů Booleovy algebry:

- Komutativní zákon KZ

$$b + a = a + b$$
$$d \cdot c = c \cdot d$$

- Asociativní zákon AZ

$$c+(b+a) = (c+b)+a$$

$$c\cdot(b+a) = (c+b)\cdot a$$

- Distribuční zákon Di

$$cb + ca = c\cdot(b+a)$$

$$(c+b)(c+a) = c+b\cdot a$$

$$L: (c+b)\cdot(c+a) \overset{Di}{=} c\cdot(c+a) + b\cdot(c+a) \overset{Di}{=} c\cdot c + c\cdot a + c\cdot b + b\cdot a \overset{AB}{=}$$

$$c + c\cdot a + c\cdot b + b\cdot a \overset{Di}{=} c\cdot(\underbrace{1+a+b}_{1}) + b\cdot a \overset{AG}{=} c\cdot 1 + ba \overset{NE}{=} c + ba$$

- Absorpční zákon AB

$$c\cdot c = c$$

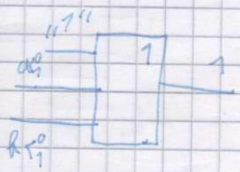
$$c + c = c$$


- Agresivní zákon AG

(5) agresivný zákon AG

$$1+a = 1$$

$$0\cdot b = 0$$

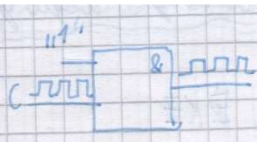



- Zákon neutrality NE

(6) zákon neutrality NE

$$c\cdot 1 = c$$

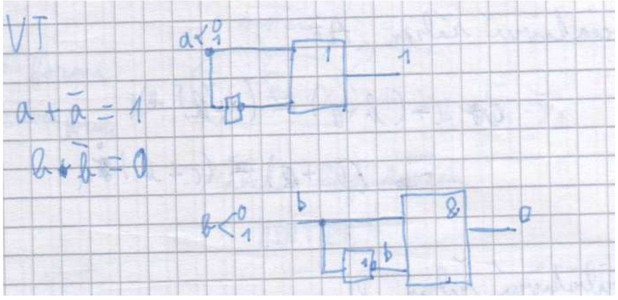
$$b+0 = b$$



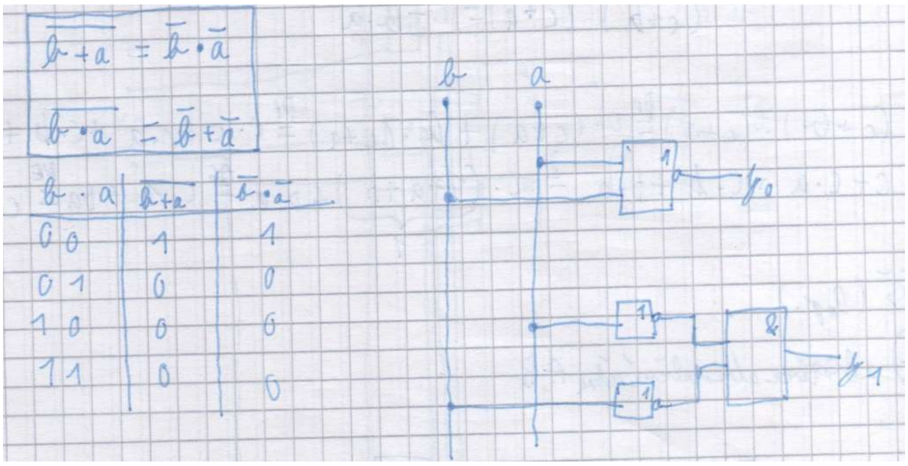

- Dvojitá negace DN

$$\bar{\bar{a}} = a$$

- Zákon vyloučení třetího VT



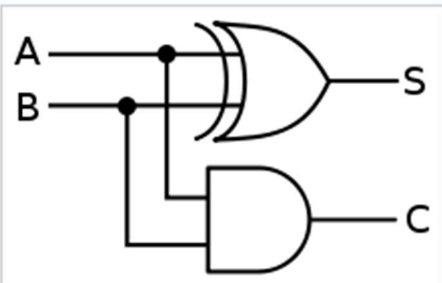

- De Morgenovo zákon DM



## Poloviční sčítačka:

- (half adder) realizuje sčítání dvou jednobitových čísel, vstup jsou dva jednobitové sčítance (A, B)
- výstupem je jednobitový součet (S) a jednobitový příznak přenosu do vyššího řádu (c)
- poloviční sčítačka dále přenáší příznak přenosu do vyššího řádu, sama však nedokáže zpracovat přenos z nižšího řádu, nestačí proto k realizaci vícebitového sčítání

| vstup | | výstup | |
|---|---|---|---|
| A | B | C | S |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |



Poloviční sčítačka z hradel XOR a AND

## Úplná sčítačka:

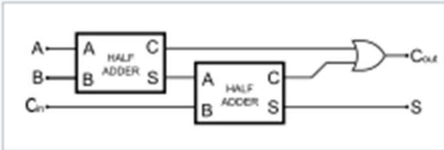
- (full adder) realizuje sčítání dvou jednobitových čísel s přihlédnutím k přenosu z předchozího řádu
- vstupem jsou tři jednobitové sčítance: A, B, Ci (Carry-in)

- výstupem je jednobitový součet (S) a jednobitový příznak přenosu do vyššího řádu (Co - Carry-out)

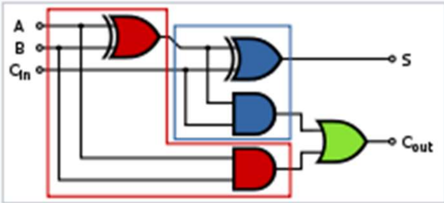
- úplnou sčítačku je možné složit ze dvou polovičních sčítaček a hradla OR, hradlo OR je navíc možné bez vlivu na funkčnost nahradit pomocí hradla XOR, protože kombinace vstupů (1, 1), v němž by se jejich výstupy lišily, nemůže v případě sčítání nastat (buď nastane přenos pouze v první poloviční sčítačce, nebo pouze v druhé), takže k vytvoření úplné sčítačky stačí mít 2 typy hradel, což může být praktické pro realizaci

- úplné sčítačky se spolu mohou vedle sebe řetězit (výstup Co jedné sčítačky propojit se vstupem Ci další) a provádět tak sčítání vícebitových čísel

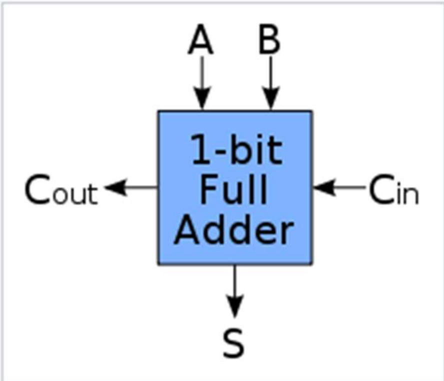
| vstup | | | výstup | |
|---|---|---|---|---|
| A | B | $C_i$ | $C_o$ | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |



Úplná sčítačka, složená ze dvou sčítaček a hradla NEBO (blokové schéma)



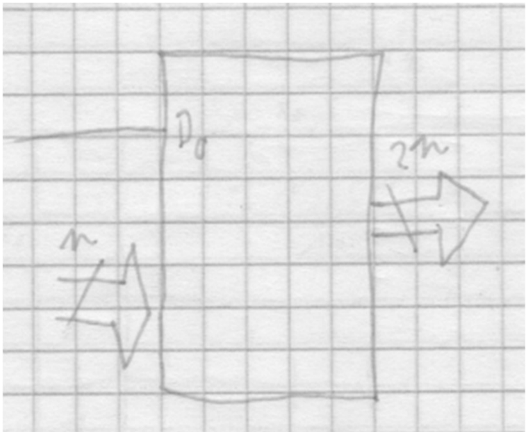
Úplná sčítačka z hradel (2x XOR, 2x AND, 1x NEBO). Dvě poloviční sčítačky jsou ve schématu označeny červeně a modře.

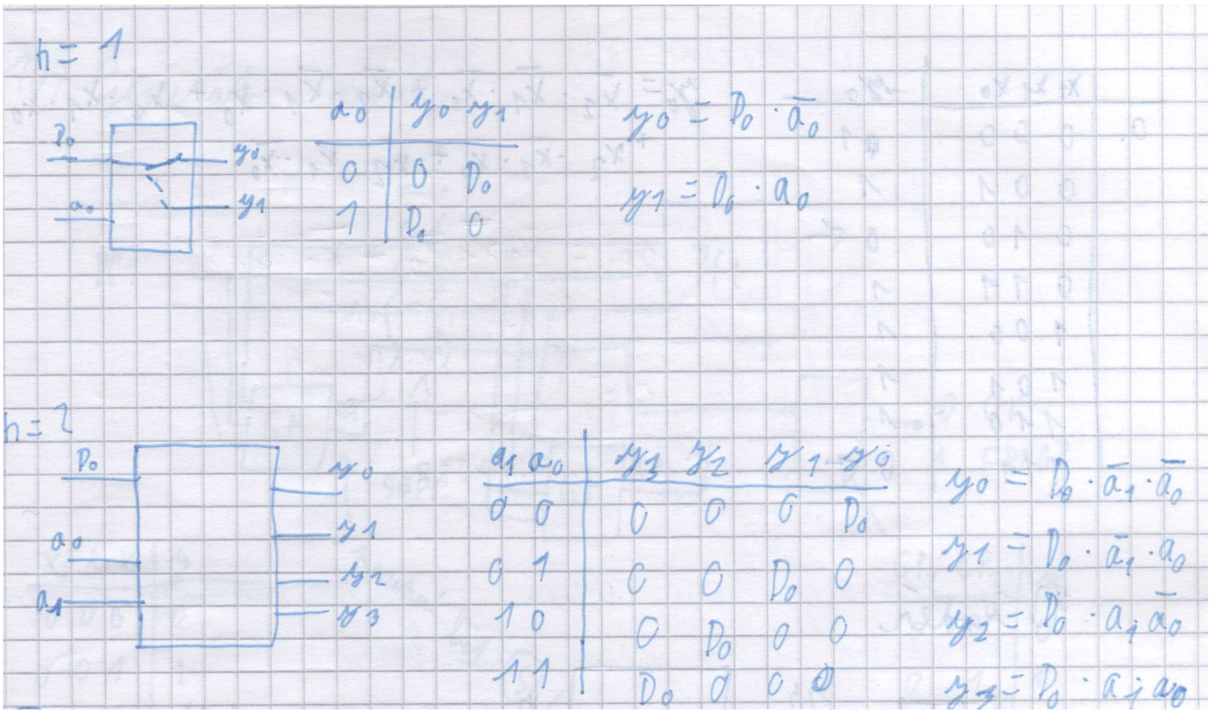


Schematické uspořádání značky jednobitové úplné sčítačky

## De-multiplexor:

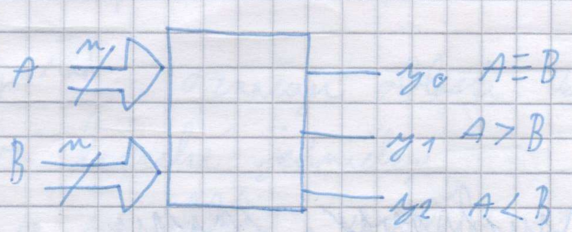
- má 1 datový vstup n adresových a až 2 na n výstupů
- elektronická součástka, fungující na principu přepínače, kdy je podle řídících signálů (a) přiváděn na výstupy (y) vstupní signál (x)
- podle adresy se hodnota datového vstupu objeví na příslušném výstupu
- demultiplexor je principiálně podobný binárnímu dekodéru, rozdíl spočívá v tom, že u demultiplexoru je nosičem informace vstup x





## Porovnávací obvod:

- kombinační logický obvod, který porovnává dvě více bitová slova a na výstupech generuje signály pro rovnost, větší a menší, obvod má 3 výstupy

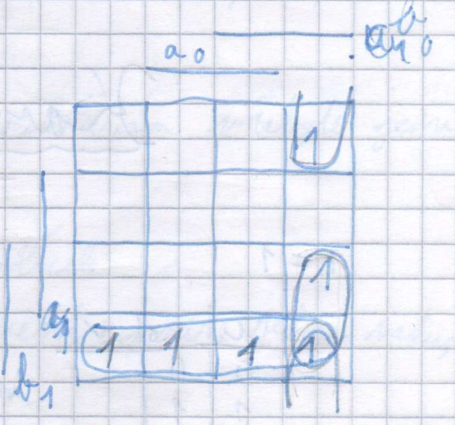



| $b_0$ | $a_0$ | $y_2$ (A < B) | $y_1$ (A > B) | $y_0$ (A ≡ B) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | ~~1~~ |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |

$y_2 = b_0 \bar{a}_0$ $y_1 = \bar{b}_0 \cdot a_0$ $y_0 = \overline{b_0 \oplus a_0}$

A < B

$y_2$




$y_2 = b_1 \bar{a}_1 + b_0 \bar{a}_1 \bar{a}_0 + b_1 b_0 \bar{a}_0$

$n = 2$

| | $b_1$ | $b_0$ | $a_1$ | $a_0$ | $y_2$ | $y_1$ | $y_0$ |
|---|---|---|---|---|---|---|---|
| 0. | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1. | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2. | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 3. | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 4. | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 5. | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 6. | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 7. | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 8. | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9. | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 10. | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 11. | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

| | $b_1$ | $b_0$ | $a_1$ | $a_0$ | $y_2$ | $y_1$ | $y_0$ |
|---|---|---|---|---|---|---|---|
| 13. | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 14. | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 15. | 1 | 1 | 1 | 1 | 0 | 0 | 1 |