

VYBRANÉ PARTIE Z ČÍSLICOVÉ TECHNIKY

1. KOMBINAČNÍ A SEKVENČNÍ OBVODY – DEFINICE

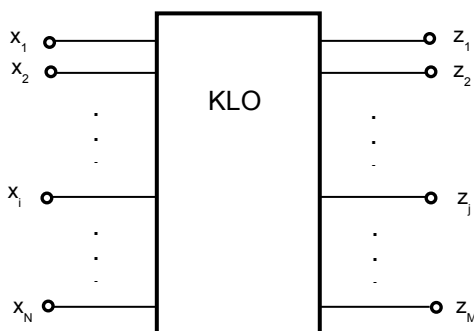
Při stavbě počítačů je třeba zabezpečit dvě základní funkce počítače. Je to potřeba sestavit obvody, které jednak provádějí přímé aritmetické a logické operace, jednak slouží jako paměti. Základem pro konstrukci obvodů, které provádějí přímé operace jsou tzv. **kombinační obvody**, základem pro konstrukci obvodů, které slouží jako paměti jsou tzv. **sekvenční obvody**. Tyto dva základní typy obvodů je možné definovat následujícím způsobem.

Definice:

Kombinační logické obvody jsou takové obvody, u nichž současná logická úroveň výstupních signálů závisí pouze na okamžité úrovni vstupních signálů.

Sekvenční logické obvody jsou takové obvody, u nichž současná logická úroveň výstupních signálů závisí nejen na okamžité úrovni signálů vstupních, ale i na logické úrovni časově předchozích signálů výstupních.

Blokové schéma kombinačního obvodu:



Matematický zápis:

$$z_j(T) = f_j(x_1; x_2; \dots; x_i; \dots; x_N; T)$$

pro všechna $i \in \langle 1; N \rangle$ a $j \in \langle 1; M \rangle$

obecně $N \neq M$

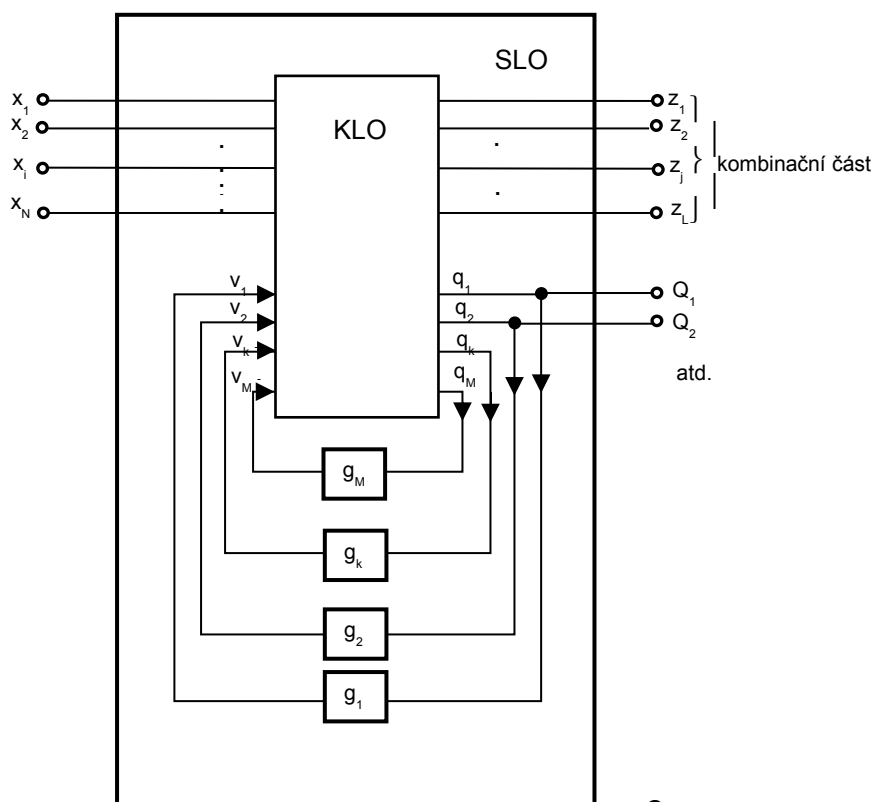
Tabulka hodnot výše uvedeného kombinačního logického obvodu, který realizuje danou logickou funkci, je tabulkové vyjádření nespojitě logické funkce. $z_j(T) = f_j(x_1; x_2; \dots; x_i; \dots; x_N; T)$ pro všechny možné kombinace vstupních hodnot x_i . Protože každá ze vstupních logických proměnných může nabývat hodnot 0 a 1, bude mít tabulka pro dvě vstupní proměnné 4 řádky, tedy 2^2 řádků (viz např. tabulky funkcí $a+b$, nebo $a.b$), tabulka pro tři vstupní proměnné 8 řádků, tedy 2^3 řádků (viz např. tabulky funkcí

c.(a+b),nebo a.b.c), obecně pro N vstupních proměnných má tabulka hodnot logické funkce 2^N řádků. V buňkách pro závisle proměnné hodnoty z_j pak jsou pro každou z kombinací hodnot vstupních veličin zapsány hodnoty, které má KLO vytvořit podle požadavku, kladeného konstruktérem obvodu, respektive jeho zadavatelem. Proto není tabulka v prostoru, vymezeném pro hodnoty z_j vyplněna.

x_1	x_2	...	x_i	...	x_{N-1}	x_N	z_1	z_2	...	z_j	...	z_L	...	z_M
1	1	...	1	...	1	1	...	...	...	...	...	...	...	...
1	1	...	1	...	1	0	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	1	...	...	...	...	0	0	...	...	...	...	...	...	...
1	0	...	1	...	1	1	...	...	...	...	...	...	...	...
1	0	...	1	...	1	0	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	0	...	0	...	0	0	...	...	...	...	...	...	...	...
0	1	...	1	...	1	1	...	...	...	...	...	...	...	...
0	1	...	1	...	1	0	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	1	...	0	...	0	0	...	...	...	...	...	...	...	...
0	0	...	1	...	1	1	...	...	...	...	...	...	...	...
0	0	...	1	...	1	0	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	0	...	0	...	0	0	...	...	...	...	...	...	...	...

tab.č. 1: Tabulka hodnot funkce obecného kombinačního obvodu $z_j = f_j(x_1; x_2; \dots; x_j; \dots; x_{1N})$

Blokové schéma sekvenčního obvodu:



Hlavní částí sekvenčního logického obvodu je obvod kombinační. To co odlišuje sekvenční obvod od kombinačního jsou zpětné vazby které jsou provedeny pomocí bloků g_k z výstupů q_k na vstupy v_k .

Právě tyto zpětné vazby dělají výstup závislým na předchozí hodnotě výstupu a přímé vstupy x_i způsobují, že hodnoty výstupu jsou závislé i na okamžitých hodnotách na vstupu. Na základě těchto vlastností lze konstruovat elektronické paměti. To znamená, že zpětná vazba umožňuje získat paměťovou vlastnost sekvenčních logických obvodů a tuto vlastnost umožňuje vytvořit.

Rovněž je třeba poznamenat, že ve zpětných vazbách vůbec nemusí být klopné obvody, ale v těch obvodech, které jsou obsahem učiva jsou zpětné vazby uskutečněny přímým vodivým propojením výstupu se vstupem.

Některé výstupy q_i z KLO uvnitř SLO nejsou vyvedeny, ale vyvedeny být mohou (viz výstupy Q1 a Q2, které vyvedeny jsou). Rovněž výstupy z_j mohou, ale nemusí být vyvedeny. Vyvedeny však musí být alespoň některé z q_M . Obecná tabulka hodnot SLO obvodu je komplikovaná a proto budou uvedeny zcela konkrétní tabulky SLO obvodů při dalším výkladu.

2. VYBRANÉ KOMBINAČNÍ OBVODY

Kromě základních logických funkcí $a+b$; $a.b$, sestavených na základě významu spojek $\vee$ („nebo“), $\wedge$ („a“) ve výrokové logice, v symbolech Booleho algebry vyjadřovaných pomocí symbolů „+“ a „.“, používá logika ještě další logické spojky, které jsou významné nejen po stránce „čistě logické“, ale jsou významné i jejich příslušné logické obvody, které logické funkce dvou proměnných, užívajících těchto spojek realizují jako elektronický obvod v číslicové technice.

Tyto spojky jsou „ $\Rightarrow$ “, „XOR“ a „ $\equiv$ “. V literatuře jsou používána i jiná označení pro tyto spojky, například pro „EXOR“ je užíván pro zkrácený symbol „XOR“ a pro „ $\equiv$ “ je užíván symbol „XNOR“. Význam uvedených spojek bude postupně vysvětlen. Mezi kombinační obvody také patří multiplexor a demultiplexor, jejichž funkce bude rovněž vysvětlena.

2.1. KOMBINAČNÍ OBVODY PRO LOGICKÉ FUNKCE $a+b$; $a.b$

Základními kombinačními obvody dle výše uvedené definice jsou obvody, realizující funkci logický součet ($a+b$) a logický součin, které byly vyloženy v předchozích partiích ČT.

2.2. OBVOD PRO LOGICKÉ „VYPLÝVÁNÍ“($\Rightarrow$)

Symbol „ $\Rightarrow$ “, je symbolem pro logické vyplývání.

Logický výrok $a \Rightarrow b$, ve kterém se spojka vyskytuje tedy znamená: Z „ a “ vyplývá „ b “. Pojem logického vyplývání je jedním ze základních pojmů nejen v logice samé ale důležitý je zejména v matematice a v přírodních vědách (fyzice, biologii, atd.), je však stejně důležitý ve vědách humanitních (filozofii, psychologii, lingvistice, vědě o právu a zákonech atd.). Pojmu logického vyplývání a aparátu, spojeného s jeho využitím se používá při „dokazování“. Příslušných obvodů se využívá i při tzv. „strojovém dokazování“. Jednou z mnoha možností použití obvodů používajících spojky je i v oblasti strojového (počítačového) vyhledávání poruch. Jestliže víme že něco z něčeho vyplývá, pak při zjištění toho, že toto vyplývání přestalo platit (výsledek ověřování je, že vyplývání přestalo platit) znamená, že systém je porušen. Toho se využívá při počítačovém zjišťování poruch v technologických (výrobních) systémech.

Výroku, ve kterém je použita spojka „ $\Rightarrow$ “ se také říká výrok podmiňovací, protože platnost „ b “ je podmíněna platností „ a “.

Říká se: „**Jestliže** platí „ a “, **pak** musí platit „ b “, pokud mezi „ a “ a „ b “ platí vztah $a \Rightarrow b$.

Příkladem může být použití v automatizační technice.

Výrok „ a “ nechť je: „Teplota v peci je v povolené toleranci“.

Výrok „ b “ nechť je: „V peci temperovaný výrobek má teplotu v povolené toleranci“.

Jestliže například nastane porucha teplotního snímače teploty v peci, může hlásit povolenou toleranci, přestože „ a “ neplatí, ale výrobek nebude mít požadovanou teplotu. Z toho lze usoudit, že v systému se vyskytla porucha, protože přestalo platit předpokládané vyplývání. Příslušný obvod pak může nahlásit poruchu automaticky.

Níže je uvedena tabulka pravdivostních hodnot logického vyplývání. Nad tabulkou jsou uvedeny i rovnocenné další logické výrazy (ekvivalentní), pomocí kterých lze

výraz „ $a \Rightarrow b$ “ realizovat jako logický obvod v číslicové technice. Jednotlivé, svou formou odlišné výrazy mají stejnou pravdivostní tabulku. Je-li tomu tak, pak je zároveň dokázáno, že výrazy jsou rovnocenné. Lze se o tom přesvědčit dosazením pravdivostních hodnot a jejich kombinací do jejich výrazů, při čemž definována je tabulka pro výraz „ $a \Rightarrow b$ “ a u ostatních výrazů se kontroluje, jestli jejich tabulka je shodná s tabulkou definovanou.

Je tedy možné psát:

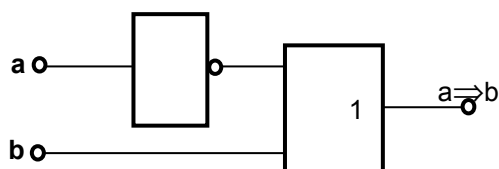
$$(a \Rightarrow b) \equiv \bar{a} + b \equiv \overline{a \cdot \bar{b}}$$

Pravdivostní tabulka logického vyplývání je následující:

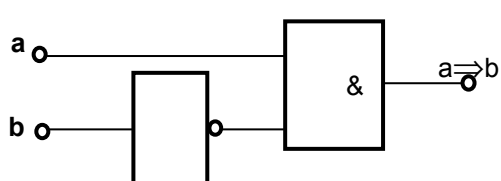
a	b	$a \Rightarrow b$
1	1	1
1	0	0
0	1	1
0	0	1

Příslušné logické obvody jsou:

$$\bar{a} + b$$



$$\overline{a \cdot \bar{b}}$$



obrázek 2-1: Bloková schémata obvodů, uskutečňujících logickou funkci $a \Rightarrow b$

2.3. OBVOD PRO „EKVIVALENCI“ ($\equiv$)

Spojka „je ekvivalentní“ znamená „rovnocenné“, je logickou spojkou ve vztahu, který se nazývá „ekvivalence“. Říká se: „a“ je ekvivalentní „b“. Z hlediska matematické logiky to znamená, že „a“ je ekvivalentní „b“ v případě, že platí $a \Rightarrow b$ a zároveň platí $b \Rightarrow a$. Kromě znaku „ $\equiv$ “ se používá rovnocenný znak „ $\Leftrightarrow$ “, tedy „ $(a \equiv b) = (a \Leftrightarrow b)$ “.

V matematické terminologii se používá pro ekvivalenci věta, podobná vyjádření vztahu logické vyplývání.

Výraz: „b“ vyplývá z „a“ můžeme také vyjádřit větou: „b“ platí tehdy, platí-li „a“.

Výraz: „b“ je ekvivalentní s „a“ můžeme také vyjádřit větou: „b“ platí tehdy, a jen tehdy, platí-li „a“.

Zatímco ve větě „ $a \Rightarrow b$ “ nemůžeme zaměnit „a“ a „b“, ve větě s ekvivalencí větu „ $a \equiv b$ “ můžeme zapsat i ve tvaru „ $b \equiv a$ “ (můžeme symboly zaměnit).

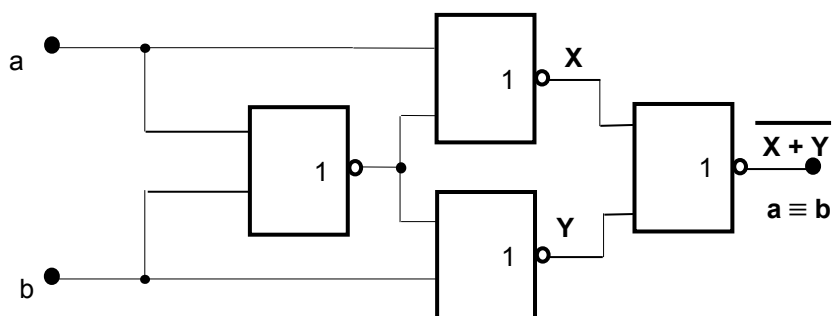
Znamená to tedy, že pravdivostní tabulka pro ekvivalenci má tvar takový, že ekvivalence platí (logická hodnota 1) tehdy, když obě logické proměnné mají stejnou pravdivostní hodnotu a neplatí (logická hodnota 0) tehdy, když logické proměnné mají navzájem rozdílnou pravdivostní hodnotu, tedy:

a	b	$a \equiv b$
1	1	1
1	0	0
0	1	0
0	0	1

Tak jako pro spojku „ $\Rightarrow$ “, tak i pro spojku „ $\equiv$ “ lze dokázat na základě postupného sestavení tabulky pravdivostních hodnot, že platí:

$$(a \equiv b) = (a.b) + (\bar{a}.\bar{b})$$

Blokové schéma obvodu, uskutečňujícího ekvivalenci je následující:



obrázek 2-2: Blokové schéma obvodu, uskutečňujícího logickou funkci $b \equiv a$ s obvody NOR.

O tom, že tento obvod realizuje logickou funkci $a \equiv b$ je možné se přesvědčit jednoduchým výpočtem, ve kterém použijeme De Morganovy zákony:

Určíme nejprve X a Y:

$$X = \overline{a + (\bar{a} + \bar{b})}, \quad Y = \overline{b + (\bar{a} + \bar{b})}$$

Potom určíme:

$$\begin{aligned}\overline{X+Y} &= \overline{a + (\overline{a+b}) + b + (\overline{a+b})} = \overline{a + (\overline{a+b})} \cdot \overline{b + (\overline{a+b})} = (a + (\overline{a \cdot b})) \cdot (b + \overline{a \cdot b}) = \\ &= a \cdot b + a \cdot \overline{a} \cdot \overline{b} + \overline{a} \cdot b + \overline{a} \cdot \overline{b} \cdot \overline{b} = a \cdot b + \overline{a} \cdot \overline{b}\end{aligned}$$

Je tedy uvedené schéma skutečně schématem obvodu, realizujícího logickou funkci ekvivalence, protože $a \cdot \overline{a} = 0$; $\overline{b} \cdot b = 0$; $\overline{a} \cdot \overline{a} = \overline{a}$; $\overline{b} \cdot \overline{b} = \overline{b}$ (laskavý čtenář si zajisté odvodí sám na základě sestavení tabulky pravdivostních hodnot pro uvedené vztahy).

Výpočet pro $\overline{X+Y}$ je matematickým, „konstruktivním důkazem“ v tom smyslu, že výsledek je skutečně sestaven (zkonstruován) na základě použití platných zákonů Booleovy algebry.

Obvody, tvořené funkcí $b \equiv a$ jsou používány např. v KLO zvaných komparátory k porovnání shodnosti binárních čísel.

2.4. OBVOD PRO „VYLUČOVACÍ NEBO (XOR)“,

Spojka XOR (z angličtiny - „exclusive or“) použita v logickém výroku: „a“ XOR „b“ znamená, že logický výrok platí jen tehdy, když „a“ a „b“ mají rozdílné pravdivostní hodnoty. Je tudíž možné napsat pravdivostní tabulku složeného výroku, respektive tabulku logické funkce:

A	b	aXORb	aORb
1	1	0	1
1	0	1	1
0	1	1	1
0	0	0	0

Porovnáním tabulky pro XOR a pro ekvivalenci zjistíme, že logický výrok „a XOR b“ je záporom ekvivalence. Tedy : $a \text{ XOR } b = \overline{a \equiv b} = \overline{a \cdot b + a \cdot \overline{b}}$

Logické proměnné „a“ a „b“ jsou v tomto případě nazývány alternativy, protože může platit jen jedna ze „složek“ alternativního vztahu.

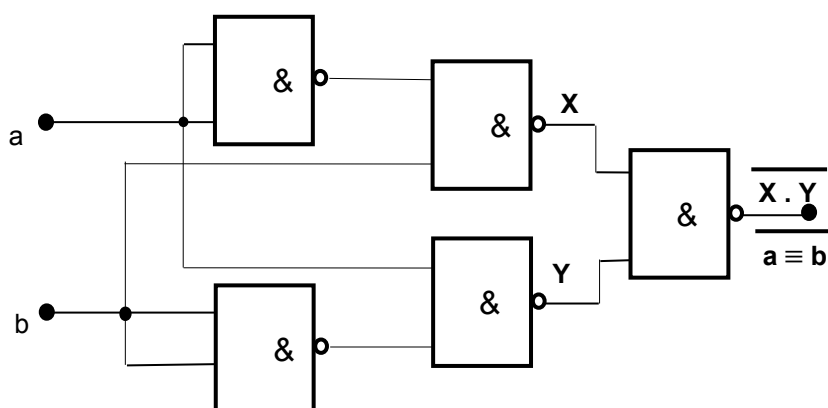
O platnosti uvedeného vztahu je možné se opět přesvědčit výpočtem.

Jestliže $(a \equiv b) = (a \cdot b) + (\overline{a} \cdot \overline{b})$, a uvážíme-li v závěru, že $\overline{a} \cdot a = 0$ a dále, že $b \cdot \overline{b} = 0$

pak:

$$\begin{aligned} \overline{(a \equiv b)} &= \overline{(a.b) + (\bar{a}.\bar{b})} = \overline{a.b} . \overline{\bar{a}.\bar{b}} = (\bar{a} + \bar{b}) . (a + b) = (\bar{a} + \bar{b}) . (a + b) = \bar{a}.a + a.\bar{b} + \bar{a}.b + b.\bar{b} = \\ &= a.\bar{b} + \bar{a}.b \end{aligned}$$

Funkci logického obvodu realizujícího logickou alternativu nazýváme též „logickým přepínačem“. Od obvodu, realizujícího „nevylučovací nebo“ (logický člen „OR“) se liší právě tím, že je nepravdivý v případě, že „a“ i „b“ jsou pravdivé. V tabulce pro spojku „EXOR“ je pro porovnání uveden i výrok s nevylučovacím nebo (logický součet). Všechny tyto uvedené obvody mohou sloužit například k vyhodnocování polohy zaměřovaného předmětu, protože funkce logických spojek může být znázorněna pomocí např. Vennových množinových diagramů a kombinací příslušných obvodů lze vytvořit obvod, který kontroluje polohu předmětů v prostoru. Takové obvody pak slouží ke konstrukci automatů, řízených počítačem, nebo autonomních (samostatně se rozhodujících) robotů.



obrázek 2-3: Blokové schéma obvodu, uskutečňujícího logickou funkci bXORa s obvodu NAND.

To, že tento obvod realizuje logickou funkci a XOR b je nutné dokázat jednoduchým výpočtem, ve kterém nejprve vyjádříme X a Y a poté použijeme De Morganovy zákony.

Tedy

$$X = \overline{a.b}, \quad Y = \overline{a.\bar{b}}$$

Potom určíme:

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

Je tedy uvedené schéma skutečně schématem obvodu, realizujícího logickou funkci „vylučovacího nebo“.

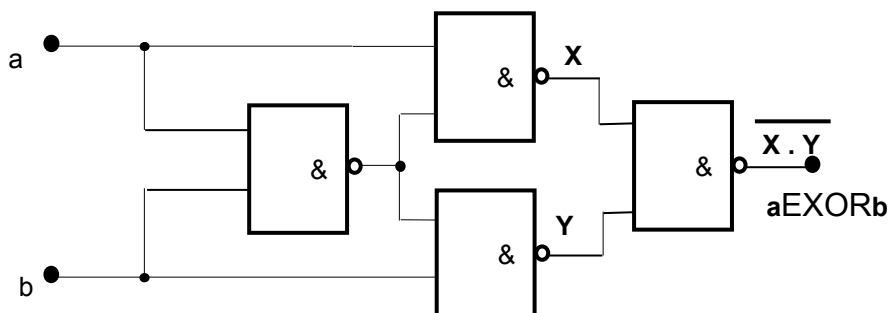
Obvod XOR lze rovněž sestavit formálně shodným způsobem (formálně proto, že forma propojení bloků ve schématu je shodná) s blokovým schématem pro zapojení obvodu s ekvivalencí a uspořít tak jeden funkční blok. Musíme však namísto bloků „NOR“ použít bloky „NAND“, viz obrázek 3-2

$$\text{Opět lze vyjádřit: } X = \overline{a \cdot a \cdot b}; Y = \overline{b \cdot a \cdot b}$$

A dále vypočítat:

$$\overline{X \cdot Y} =$$

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



obrázek 2-4: Blokové schéma obvodu, uskutečňujícího logickou funkci bEXORa s obvodu NAND v odlišném zapojení než dle obrázku 3 –1, formálně shodném se zapojením dle obrázku 2-1.

Podobně lze dokázat, že změnou bloků NAND na bloky NOR ve schématu na obrázku 3-1 se mění funkce obvodu z „a XOR b“ na funkci „a ≡ b“.

Poznámka ke značení funkcí: Logická funkce „a ≡ b“ je též označována jako „a XNOR b“, protože ekvivalence je zápor vylučovacího nebo, tedy:

$$(a \equiv b) \equiv (\overline{a \text{ XOR } b}) \equiv (a \text{ XNOR } b)$$

Toto označení se vyskytuje nejen v literatuře, ale i na některých druhích kalkulaček.

Použití všech těchto druhů jednoduchých kombinačních obvodů je velmi rozsáhlé a to nejen v číslicové technice, ale i v automatizaci atd.

2.5. MULTIPLEXOR

Multiplexor je kombinační logický obvod, který umožňuje převádět číslicové informace z jednoho vstupu z celkového počtu **n vstupních datových vstupů** na jediný datový výstup. Výběr určitého datového vstupu provádíme řídicími adresovými vstupy, při čemž číslo datového vstupu je udáno binárním kódem.

Mezi počtem datových vstupů **n** a počtem adresových vstupů **M** platí vztah

$$n = 2^M$$

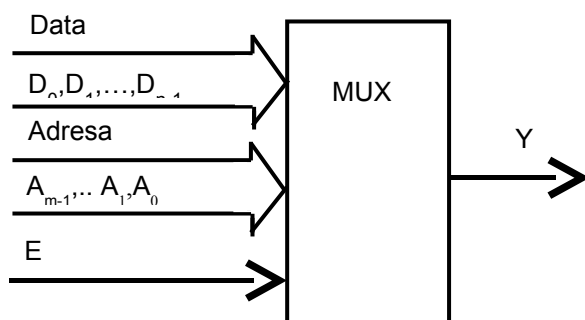
Pro multiplexor se 4 vstupy D_0, D_1, D_2, D_3 , jsou příslušné adresy v binárním kódu:

$$A_0 = 00, A_1 = 01, A_2 = 10, A_3 = 11$$

Znamená to tudíž, že při čtyřech datových vstupech postačí dva adresové vstupy.

Obecně $A_{(bin)} = a_{m-1} \dots a_1 a_0$ což znamená, že adresa M – tého kanálu je binární číslo a na prvním adresovém kanálu se vyskytuje nultý řád čísla, na druhém adresovém kanálu první řád čísla a na m - prvním adresovém kanálu m – první řád čísla adresy v binárním kódu.

Schématický znak multiplexoru je na následujícím obrázku:



Vstup **E** (Enable) je používán k řízení činnosti celého obvodu. Jeho aktivní úroveň umožňuje snímání dat právě z tohoto obvodu. Je-li multiplexor vytvořen z více kusů integrovaných obvodů, jsou jednotlivé IO spínány právě, když $E=1$. Pro výstup z multiplexoru můžeme psát Booleovu funkci:

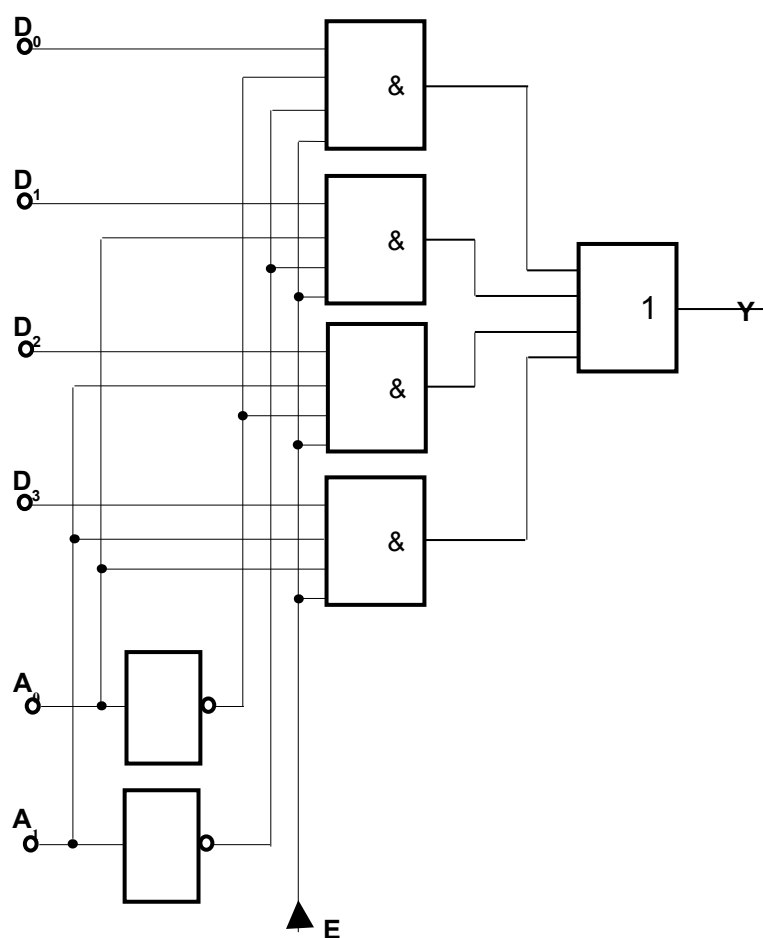
$$Y = (A_0 \cdot D_0 + A_1 \cdot D_1 + A_2 \cdot D_2 + \dots + A_{(n-1)} \cdot D_{(n-1)}) \cdot E$$

A_i znamená celou adresu datového kanálu D_i

Pravdivostní tabulka pro čtyřvstupový multiplexor.

E	D ₀	D ₁	D ₂	D ₃	A ₁	A ₀	Y
1	0	X	X	X	0	0	0
1	1	X	X	X	0	0	1
1	X	0	X	X	0	1	0
1	X	1	X	X	0	1	1
1	X	X	0	X	1	0	0
1	X	X	1	X	1	0	1
1	X	X	X	0	1	1	0
1	X	X	X	1	1	1	1
0	X	X	X	X	X	X	0

Na následujícím schématu je uvedeno blokové schéma čtyřvstupového multiplexoru



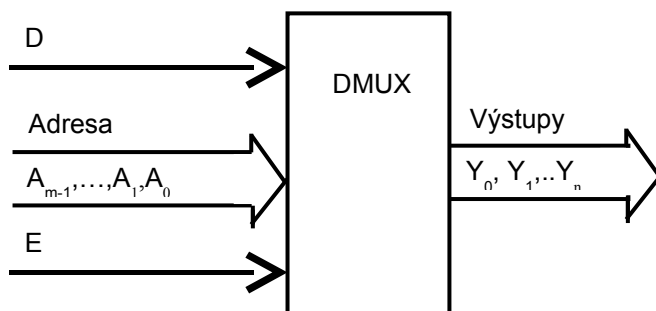
2.6. DEMULTIPLEXOR

Demultiplexor je kombinační logický obvod, který umožňuje převádět číslicové informace z jednoho, jediného datového vstupu na některý z N datových výstupů, při čemž daný výstup je určen adresou v binárním kódu, která je na adresových vstupech demultiplexoru.

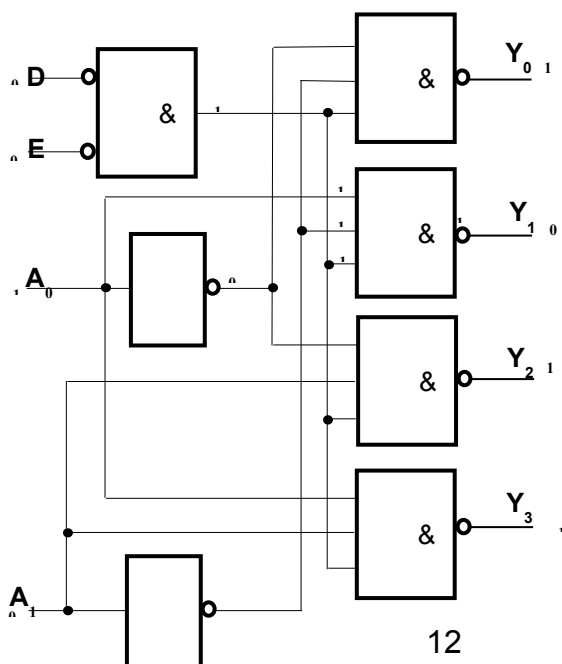
Demultiplexor plní opačnou funkci než multiplexor. V závislosti na logickém stavu adresových vodičů demultiplexor přenáší z jediného vstupního datového vodiče data na jeden z výstupních vodičů, zatímco na ostatních výstupních vodičích setrvává neaktivní stav. Co je neaktivní stav závisí na vlastní konstrukci logického obvodu po stránce elektronické.

Jde tedy o kombinační obvod, jehož výstup můžeme popsat v závislosti na vstupu následujícími Booleovými výrazy $Y_i = A_i \cdot D \cdot E$

Schématický znak demultiplexoru je na následujícím obrázku:



Čtyřvýstupový demultiplexor má následující blokové schéma



Na schématu jsou vyznačeny některé úrovně, které jsou v obvodu při zápisu nuly na výstup Y_1 . Kroužky před vstupy E a D do demultiplexoru jsou schematicky zapsané invertory, podobně jako jsou invertující kroužky na výstupu NAND a NOR obvodů.

Pro tento obvod platí následující pravdivostní tabulka.

E	D	$A_{II(1)}$	$A_{I(0)}$	Y_0	Y_1	Y_2	Y_3
0	1	0	0	1	1	1	1
0	0	0	0	0	1	1	1
0	1	0	1	1	1	1	1
0	0	0	1	1	0	1	1
0	1	1	0	1	1	1	1
0	0	1	0	1	1	0	1
0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0
1	X	X	X	1	1	1	1

Z tabulky je vidět, že tento demultiplexor je vypínán jedničkou na vstupu E.