ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

SADA PŘÍKLADŮ PRO PODPORU VÝUKY LOGICKÉ ALGEBRY

BAKALÁŘSKÁ PRÁCE

Jiří Noska
Informatika se zaměřením na výuku

Vedoucí práce: Ing. Petr Michalík, Ph.D.

Plzeň, 2019

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni, 18. dubna 2019

.....

vlastnoruční podpis

MÉ PODĚKOVÁNÍ PATŘÍ ING. PETRU MICHALÍKOVI, PH.D., ZA ODBORNÉ VEDENÍ, TRPĚLIVOST A OCHOTU, KTEROU MI V PRŮBĚHU ZPRACOVÁNÍ BAKALÁŘSKÉ PRÁCE VĚNOVAL.

ZDE SE NACHÁZÍ ORIGINÁL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

SEZNAM ZKRATEK

NOT	**logická negace**
AND	**logický součin**
OR	**logický součet**
NAND	**logický negovaný součin**
NOR	**logický negovaný součet**
EX-OR, XOR	**nonekvivalence, neshoda**
EX-NOR, XNOR	**ekvivalence, shoda**

ÚVOD

Logická algebra, je jednou z důležitých matematických disciplín. Její přínos, zejména na poli informatiky, je neopomenutelný. Stroje, stejně jako lidská osobnost, se rozhodují na základě vstupních údajů, které lze právě díky logické algebře velmi dobře popsat a matematicky vyjádřit. Na základě zákonů, které navazují na logickou algebru, lze proces rozhodování zjednodušit bez toho, aby tím byl ovlivněn výsledek. Na základě požadavku vytváříme pro stroje programy. V případě, že by docházelo k programování s úplnými logickými výrazy bez procesu, který nazýváme minimalizace, docházelo by ke zbytečně složitým realizacím včetně neefektivního využívání číslicových komponent. Druhým hlediskem může být i možná náchylnost ke zvýšené chybovosti obvodu díky složitějšímu zapojení. Znalost procesu minimalizace je tak důležitým požadavkem pro úspěchy v oblasti technologií.

Hlavním cílem této práce je podpora výuky logické algebry tak, aby případní studenti byli schopni porozumět a pochopit minimalizace logických funkcí. Sada příkladů, kterou práce přináší, by k tomu měla napomoci. Postup při řešení by se neměl ubírat pouze jedním směrem, ale cílem práce bude ukázat více způsobů, kterými lze dosáhnout požadovaného výsledku. Vedle pochopení významu a užití jednotlivých zákonů Booleovy algebry při minimalizaci, je uvedena možnost procesu minimalizace na základě znalosti pravdivostních tabulek za pomoci softwarového vybavení, v daném případě se jedná o možnosti simulačního programu Multisim.

Dalším cílem této diplomové práce je demonstrovat ekvivalenci vstupních zadaných logických funkcí a výsledků minimalizace. K tomu bude často využito prostředí simulačního programu Multisim, který umožňuje analyzovat jednotlivé logické obvody a na tomto základě reprodukovat chování číslicového obvodu.

1 STRUČNĚ Z HISTORIE

Matematika a logika jsou navzájem těsně propojené disciplíny. Významově pochází slovo logika z řeckého slova „logos" a vyjadřuje činnosti zabývající se pravdou, výroky a jejich vzájemným působením (Mareš, 2008). Postupný vývoj logiky vyústil v dnešní matematické pojetí logiky, která obsahuje výrokové formule, funkce a relace. Ve starověkém Řecku se již Platón zabýval metodikou deduktivních logických teorií (Mareš, 2008).

Jednou z hlavních postav vývoje logiky jako vědní disciplíny je Aristoteles ze Stageiry. Jeho spisy, které pojednávají o logice, jsou „Organon" a „Metafyzika". Již v sedmnácti letech je žákem Platónovy akademie, kde zůstává do roku 347 př. n. l. Jeho přínos na poli logiky je v jeho rozboru slov a vět, v charakteristice sylogismu[1] a v odvozovacích pravidlech. Aristoteles celý život budoval logiku jako deduktivní vědu (Mareš, 2008).

Ačkoliv Aristoteles pokládá poměrně komplexní základy logiky, zbývá na pokračovatele jeho práce poměrně dost práce. Mezi pokračovatele lze zařadit Diodora Krona a jeho žáka Filona z megarské školy. Tito dva přispívají studiem složených výroků. I další pokračovatelé studují logiku pomocí logických spojek a negace (Mareš, 2008).

Logika jako vědní obor však zánikem starověkých civilizací neupadne v zapomnění, ale dále se pracuje na zdokonalení této vědy. Významnými nástupci Aristotela jsou scholastici. Ti rozpracovávají formální logiku do stavu, ve kterém ji lze začlenit do matematických disciplín. Mezi známé scholastiky dané doby patří Tomáš Akvinský. Tento mnich aplikuje výše uvedenou aristotelskou logiku na studium bible. Díky jeho výroku *„Musí se říci, že pod všemohoucnost boží nespadá, co obsahuje odporování.*" (Mareš, 2008) mu však právě studium bible pomocí logiky nepřináší problémy, které by se jinak daly očekávat. Spis Tomáše Akvinského, který dokonce při tridentském koncilu roku 1545 leží na oltáři vedle bible, se jmenuje „Suma Theologicae" (tamtéž).

Dalšími scholastiky na poli logiky jsou františkánský mnich William Ockham a jeho žák Jean Buridan, oba žijící na přelomu 13. a 14. století. Na základě Ockhamovy logiky je nahlíženo na teorie jinak, než tomu bylo běžné do té doby. Díky Williamovi lze odfiltrovat

[1] Sylogismus je logické tvrzení, kdy závěr je odvozen z daných předpokladů. (Zdroj: http://www.slovnik-cizich-slov.cz/sylogismus.html)

z teorie přídavky, které nejsou podstatné pro vysvětlení této teorie. Ta je tak platná bez nich stejně jako s nimi (Mareš, 2008).

Nejvýznamnější osobou na poli logické algebry je George Boole, který se narodil 2. listopadu 1815. Již od narození se jevil jako velmi nadané dítě. Jeho otec, John Boole, kterého zajímala matematika, mladého George vedl ke stejné zálibě. Jeho znalosti však stačily pouze na výuku do osmého roku mladého George. Knihu Geometrie, kterou roku 1811 napsal skotský matematik a fyzik John Leslie, měl prostudovanou již ve svých jedenácti letech. Ve dvanácti letech pak již došlo na překlad latinské poezie. Čtrnáctý rok věku pak byl ve znamení zvládnuté němčiny, italštiny, francouzštiny a starořečtiny. V pouhých dvaceti letech si otevírá vlastní školu. Poté již přichází první matematický článek *Researches on the Theory of Analytical Transformations, with a Special Application to the Reduction of the General Equation of the Second Order* (*Výzkumy v teorii analytické transformace, se speciálním použitím pro redukci obecných rovnic druhého řádu)* vydaný roku 1840. Díky dopisováním s matematikem Augustem de Morganem se Boole dostává do matematické komunity. V roce 1847 vydává publikaci *The Mathematical Analysis of Logic (Matematická analýza logiky)* (Mareš, 2008). Kniha svým obsahem navazující na dílo Gottfrieda Leibnitze zabývající se binární soustavou. Dílo zařazuje logiku spíše na pole matematické, nikoliv k filozofii. V roce 1854 vydává George Boole dílo *Laws of Thoughts (Zákony myšlení)*, na kterém spolupracoval i se svou budoucí manželkou Mary Everest. Díky tomuto dílu byla logika již obecně vnímána jako matematická věda. Dne 8. prosince 1864 dostává zápal plic a umírá. Otisk George Booleho v historii logické algebry je nesmazatelný. Logická algebra nese jeho jméno, stejně jako kráter na měsíci pojmenovaný po něm roku 1967 (Lebrová, 2009). Booleova představa o logické algebře byly následující. Opíral se v nich především o dvě stěžejní myšlenky. První z nich bylo, že v případě logiky existuje množina prvků, se kterou se dají provádět matematické operace. V případě logické algebry je tato množina velmi malá, jedná se pouze o dva prvky – PRAVDA a NEPRAVDA. Dále si Boole uvědomil, že lze zaznamenat logické operace s těmito hodnotami. Boole vypracoval pravidla, dle kterých se Booleova algebra řídí (Mareš, 2008).

2 ÚVOD DO PROBLEMATIKY LOGICKÉ ALGEBRY

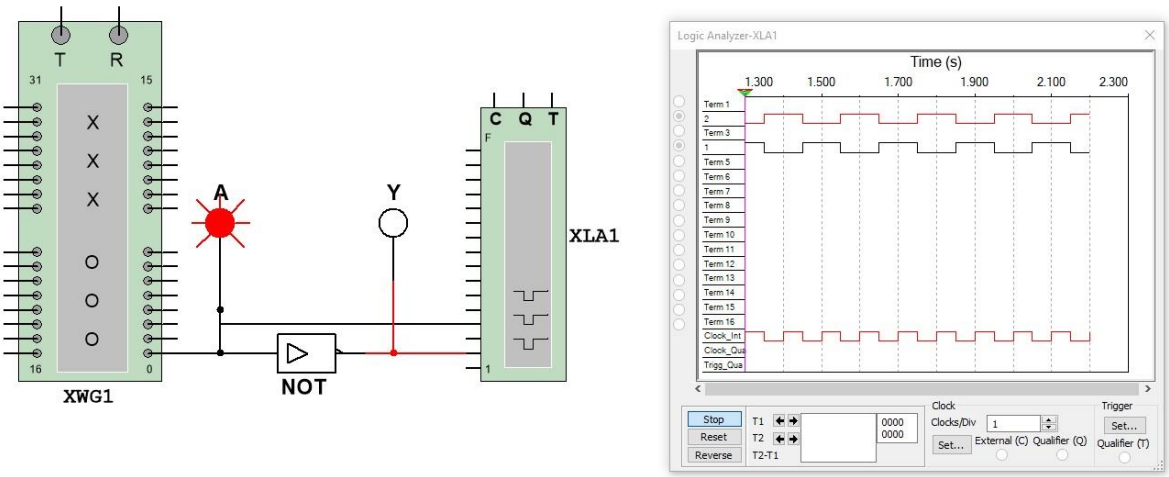
Logická neboli Booleova algebra je svazem, který má právě nulu a jednotku, je komplementární a distributivní. Lze tedy říci, že tedy ke každému prvku Booleovy algebry existuje právě jeden jeho komplement (Kopka, 1991).

2.1 LOGICKÉ FUNKCE

Logická funkce je vztah mezi vstupní proměnou a výstupní proměnou. Tento vztah se nazývá operací. Jako vstupní proměnnou si lze definovat v podstatě jakékoliv dvě hodnoty, které jsou si navzájem v opozici, výše byla uvedena hodnota *PRAVDA* a *NEPRAVDA*, obecně se pak používají ještě hodnoty *TRUE* a *FALSE*, *HIGH* a *LOW*, *ANO* a *NE* a dále *logická 1* a *logická 0*. Tyto proměnné pak mohou vstupovat do vzájemných vztahů, výsledkem je pak výrok závisející na druhu logické funkce. Mezi základní operace patří negace, logický součin, logický součet a jejich negované tvary, dále ekvivalence a její negovaná podoba (Michalík & Semrád, 2007).

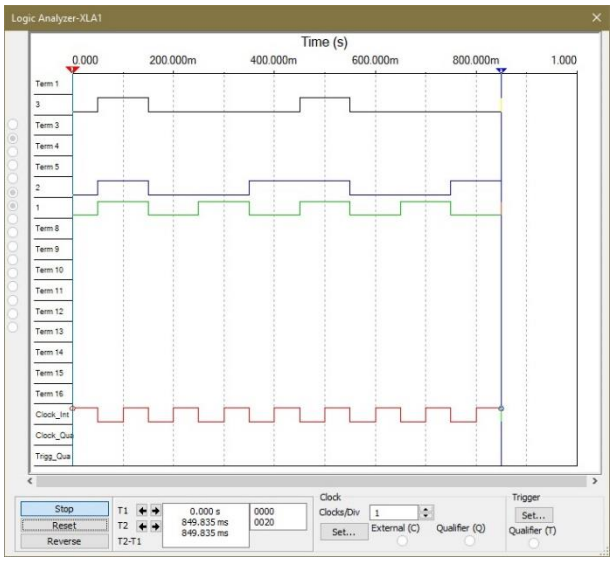
2.1.1 ZÁKLADNÍ LOGICKÉ OPERACE

Negaci neboli NOT lze interpretovat poměrně snadno. Pokud je vstupní hodnota logická 0, pak výstupní proměnou je její inverzní hodnota, tj. logická 1. Pokud bude vstupní hodnotou logická 1, pak výstupní hodnota bude samozřejmě logická 0. Tuto operaci označujeme vložením pruhu nad proměnnou (např.$\bar{A}$), popř. znakem ¬. Negace pracuje s jedinou vstupní hodnotou (Matoušek, 2004).



Obrázek 1 - Simulace logické funkce NOT (Zdroj: vlastní)

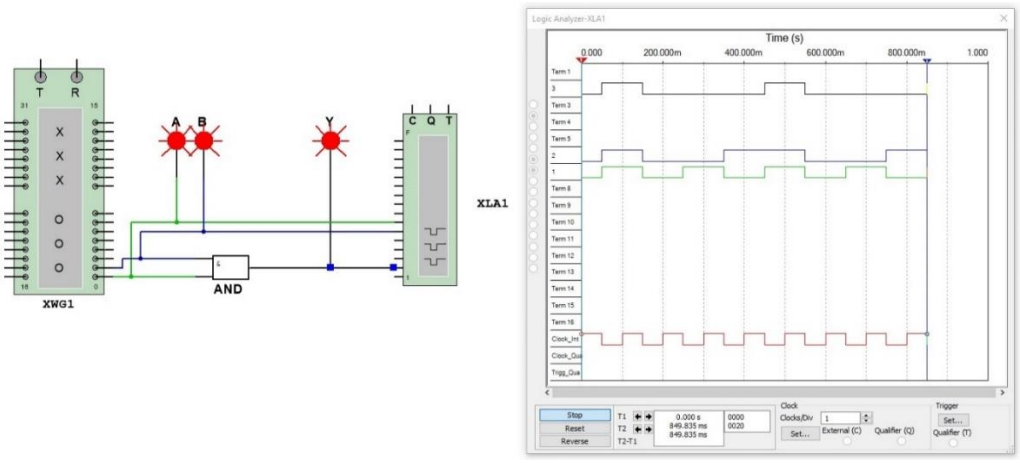
V simulacích je použit komponent logický analyzátor, který slouží k diagnostice digitálních obvodů (Juránek, 2008).



Obrázek 2 - Ukázka výstupu logického analyzátoru (Zdroj: vlastní)

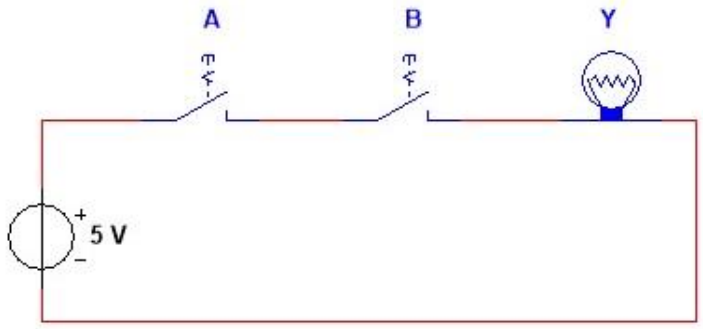
Na vstupy logického analyzátoru připojíme různé body prověřovaného logického obvodu, nejčastěji však, v našem případě, hodnoty vstupních a výstupních obvodů. V ukázkovém případu na obrázku 2 je znázorněn průběh logické funkce AND, kde vstupy do logického prvku AND jsou označeny 1 a 2 (barvy modrá a zelená) a výstup z logického prvku AND je označen číslem 3 (barva černá). Dále je znázorněn interní hodinový puls (Clock_Int – vyznačen barvou červenou), který lze měnit dle potřeby.

Logický součin neboli AND je průnikem minimálně dvou vstupních proměnných. Výstupní proměnnou je logická 1 pouze v případech, že všechny vstupní proměnné jsou rovny logické 1, logickou 0 v roli výstupní proměnné pak dostaneme v případech, že alespoň jedna ze vstupních proměnných je rovna logické 0. Operaci zapisujeme jako $*$, popř. mezi členy neuvádíme žádný znak, zápis pak vypadá AB.



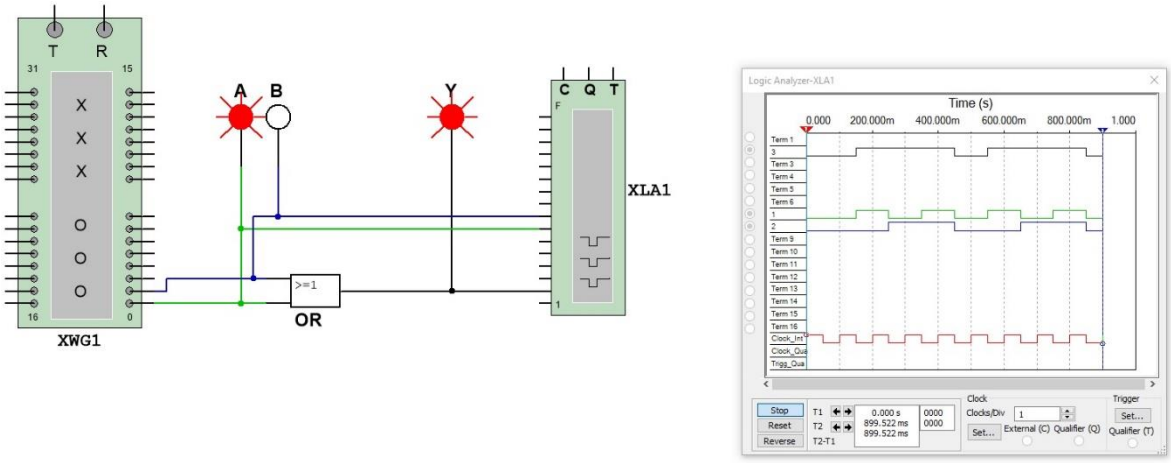
Obrázek 3 - Simulace logické funkce AND (Zdroj: vlastní)

Pro názornější představu si lze tuto logickou funkci simulovat na neomezeném počtu sériově zapojených spínačů. Logická 1 se na výstupu objeví pouze v případě, že jsou všechny spínače sepnuty.



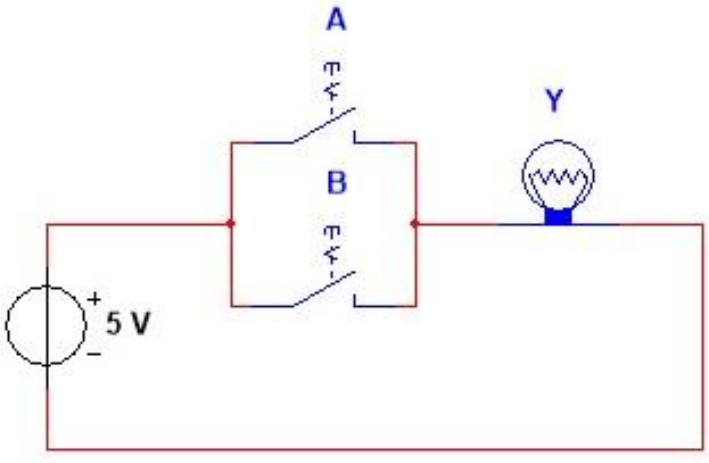
Obrázek 4 - Logická funkce AND pomocí spínačů (Zdroj: vlastní)

Logický součet neboli OR je sjednocením minimálně dvou vstupních proměnných. Výstupní proměnnou je logická 1 v případech, že alespoň jedna ze vstupních hodnot je rovna logické 1, logickou 0 jako výstupní proměnnou získáme pouze v případě, že jsou všechny vstupní hodnoty rovny logické 0. Znakem této operace je $+$.



Obrázek 5 - Simulace logické funkce OR (Zdroj: vlastní)

Stejně jako v předchozím případě i logickou funkci OR si lze zjednodušit jako soustavu neomezeného počtu paralelně zapojených spínačů. Proud prochází obvodem za předpokladu, že je minimálně jeden ze spínačů sepnut.



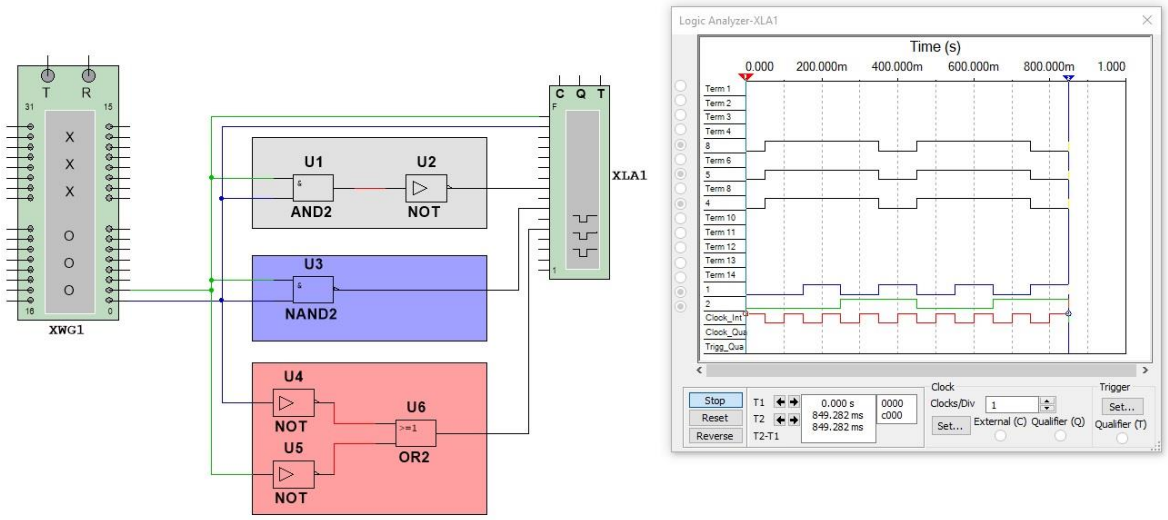
Obrázek 6 - Logická funkce OR pomocí spínačů (Zdroj: vlastní)

Negovaný logický součin neboli NAND je negovaným průnikem vstupních proměnných. Tuto operaci lze realizovat také součtem negovaných vstupních proměnných. Tato operace se též nazývá Shefferova funkce. Výstupní proměnnou je logická 1 v případech, že alespoň jedna ze vstupních hodnot je rovna logické 0, logickou 0 získáme na výstupu jen, pokud jsou všechny vstupní hodnoty rovny logické 1. Tato logická funkce není závislá na počtu vstupních proměnných.

Tabulka 1 - Pravdivostní tabulka logické funkce AND a NAND

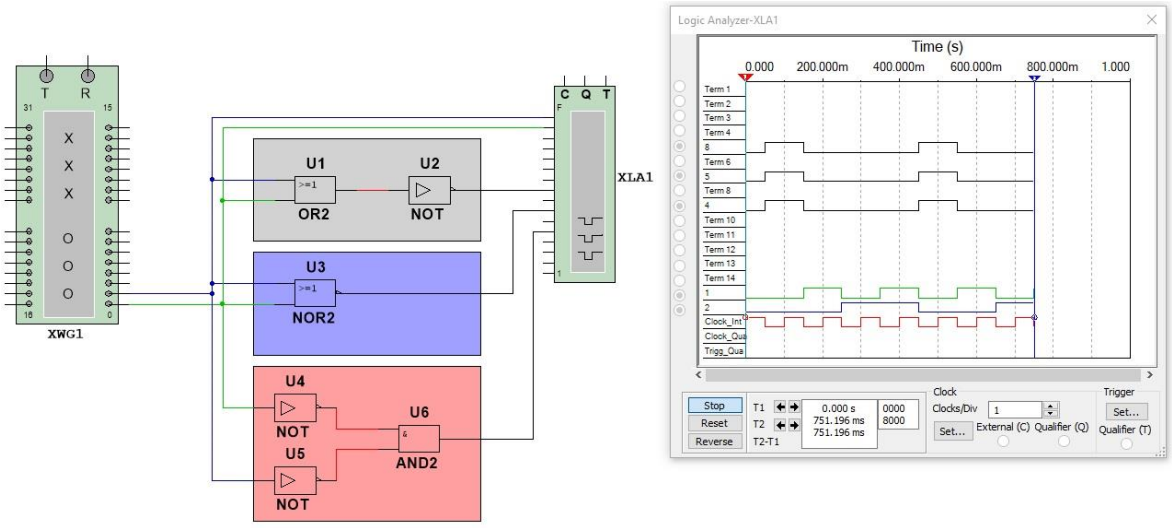
Pravdivostní tabulka AND, NAND			
A	**B**	**AND**	**NAND**
0	**0**	**0**	**1**
0	**1**	**0**	**1**
1	**0**	**0**	**1**
1	**1**	**1**	**0**

Z tabulky číslo 1 je zřetelné, že výsledku lze dosáhnout nejen zapojením logického prvku NAND, ale i zapojením prvku AND a jeho následnou negací. Třetím způsobem, jak dosáhnout daného výsledku je součtem negovaných vstupních hodnot.



Obrázek 7 - Simulace logické funkce NAND (Zdroj: vlastní)

Negovaného logického součtu neboli NOR lze dosáhnout třemi způsoby. Prvním je inverzí vstupních proměnných a jejich následným součinem. Druhý způsob je součet vstupních hodnot a následná negace výsledku, posledním je zapojení prvku NOR. Tuto funkci nazýváme Peirceova funkce. Logickou 1 na výstupu dosáhneme pouze v případě, že všechny vstupní proměnné jsou rovny logické 0. Pokud se na vstupu nachází alespoň jedna logická 1, nezávisle na počtu vstupních proměnných, hodnota výstupní proměnné je pak rovna logické 0.



Obrázek 8 - Simulace logické funkce NOR (Zdroj: vlastní)

Nonekvivalence, též EX-OR, XOR nebo také neshoda, je neshoda vstupních proměnných. V případě dvou vstupních proměnných je výsledek jednoznačně dán a to pokud $a \neq b, pak\ y = 1$**. U tří a více vstupních proměnných se vytvoří dvojice, které již lze dle**

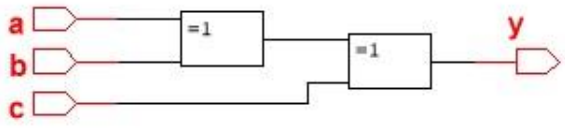
přechozí definice vyjádřit, a následně se v případě lichých vstupních proměnných k výsledku přiřadí zbývající vstupní proměnná.

Jestliže máme tři vstupní proměnné, pak lze vyjádřit postup $y = (a \oplus b) \oplus c$. V případě, že vstoupí další proměnná, pak postupujeme dle $y = (a \oplus b) \oplus (c \oplus d)$. Na následující pravdivostní tabulce a na příkladu zapojení simulujeme funkci nonekvivalence pro tři a čtyři vstupní proměnné. Při více vstupních proměnných postupujeme analogicky jako v níže uvedených příkladech. Nonekvivalence se značí $\oplus$ (Michalík & Semrád, 2007).

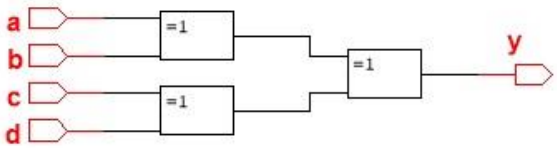
Tabulka 2 - Pravdivostní tabulka nonekvivalence – 3 a 4 vstupní proměnné

XOR - 3 vstupní proměnné				
a	b	c	a ⊕ b	y = (a ⊕ b) ⊕ c
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

XOR - 4 vstupní proměnné						
a	b	c	d	a ⊕ b	c ⊕ d	y = (a ⊕ b) ⊕ (c ⊕ d)
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	0	1	1
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	1	0
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	1	1	0
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	0	0	0



Obrázek 9 - Zapojení nonekvivalence se třemi vstupy (Zdroj: vlastní)



Obrázek 10 - Zapojení nonekvivalence se čtyřmi vstupy (Zdroj: vlastní)

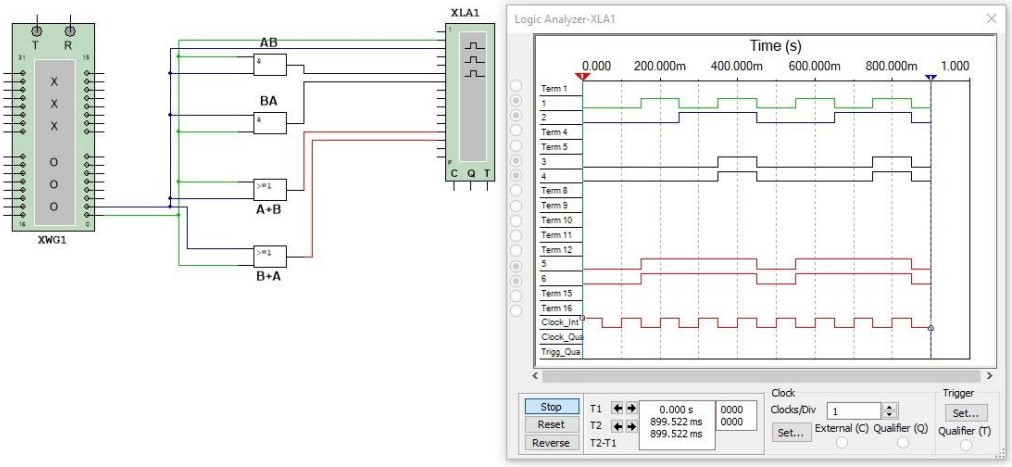
Ekvivalence, nebo také shoda, XNOR, či EX-NOR, je prostou negací nonekvivalence. Matematicky ji lze vyjádřit jako $XNOR = \overline{XOR}$. Ekvivalence se značí ⇔ (Michalík & Semrád, 2007).

2.1.2 ZÁKONY BOOLEOVY ALGEBRY

Zákon komutativnosti stanovuje, že záměnou pozice vstupních proměnných se nemění výstupní proměnná. Tento zákon lze aplikovat nejen na dvě vstupní proměnné, ale na jakýkoliv počet vstupních hodnot.

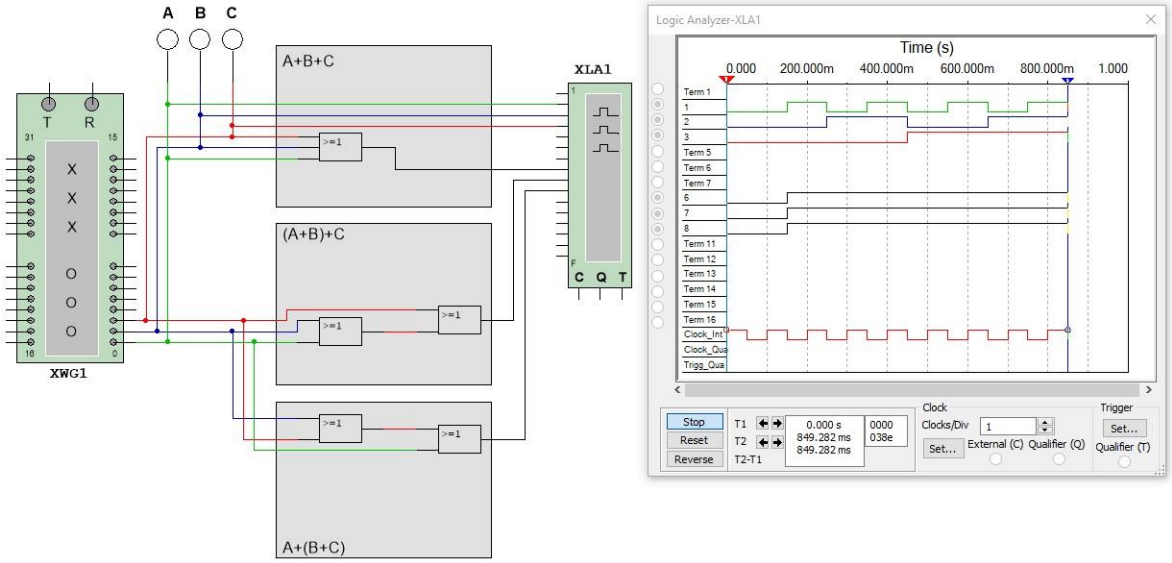
Tabulka 3 - Potvrzení zákonu komutativnosti

Zákon komutativnosti		logický součet		logický součin	
A	B	A+B	B+A	AB	BA
0	0	0	0	0	0
0	0	0	0	0	0
0	1	1	1	0	0
0	1	1	1	0	0
1	0	1	1	0	0
1	0	1	1	0	0
1	1	1	1	1	1
1	1	1	1	1	1



Obrázek 11 - Zákon komutativnosti (Zdroj: vlastní)

Stejně jako zákon komutativnosti, tak i zákon asociativnosti není vázán na počet vstupních proměnných. Jedinou podmínkou je nutnost shody logické operace. V případě, že je shodná logická operace mezi vstupními proměnnými, pak lze kamkoliv umístit závorky bez ohledu na změnu výstupní hodnoty.



Obrázek 12 - Simulace a ověření zákona asociativnosti (Zdroj: vlastní)

Výše uvedená simulace potvrdila zákon asociativnosti v případě logického součtu. Stejně jako v případě logického součtu platí tento zákon i na logický součin. Potvrzení najdeme v následující pravdivostní tabulce.

Tabulka 4 - Potvrzení zákonu asociativnosti

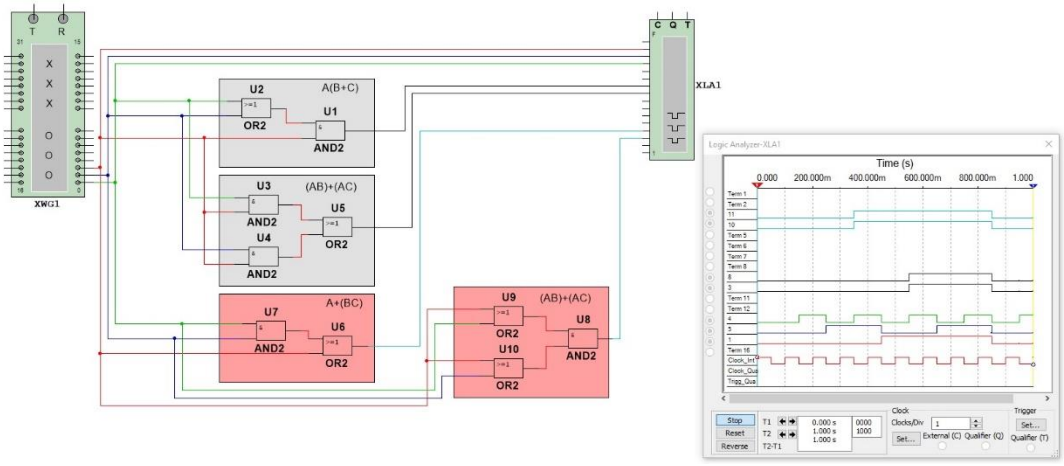
Zákon asociativnosti			logický součet			logický součin		
A	B	C	A+B+C	(A+B)+C	A+(B+C)	ABC	(AB)C	A(BC)
0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0
0	1	0	1	1	1	0	0	0
0	1	1	1	1	1	0	0	0
1	0	0	1	1	1	0	0	0
1	0	1	1	1	1	0	0	0
1	1	0	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1

Na případ logického součinu vstupní proměnné se závorkou, ve které je logickou operací logický součet, aplikujeme zákon distributivnosti. Stejně tak na případ logického součtu se závorkou, kde se nachází logický součin.

Tabulka 5 - Pravdivostní tabulka u zákona distributivnosti

Zákon distribuce						
A	B	C	A(B+C)	(AB)+(AC)	A+(BC)	(A+B)(A+C)
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

V následující simulaci je ověření zákonu distributivnosti. Z neupravené i upravené logické funkce je na výstupu vždy stejná hodnota.



Obrázek 13 - Ověření zákonu distributivnosti (Zdroj: vlastní)

V zákonu o vyloučení třetího stavu platí, že v případě logického součtu hodnoty a negované hodnoty je výsledkem vždy logická 1. V případě logického součinu hodnoty a negované hodnoty platí, že výsledek bude vždy logická 0. Tento zákon platí pro jakoukoliv vstupní hodnotu.

Tabulka 6 - Potvrzení pravdivosti zákona o vyloučení třetího stavu

Vyloučení třetího stavu		
A	A+¬A	A¬A
0	1	0
1	1	0

Zákon o neutrálnosti logické 0 a logické 1 nám stanoví, že v případě logického součtu jakékoliv vstupní proměnné, popř. vstupních proměnných, s logickou hodnotou 0 je výstupní hodnota rovna vždy hodnotě vstupní. V případě logického součinu vstupní proměnné a logické 1 je výstupní hodnota vždy rovna vstupní hodnotě.

Tabulka 7 - Potvrzení pravdivosti zákona o neutrálnosti logické 0 a logické 1

Neutrálnost 0 a 1		
A	A+0	A1
0	0	0
1	1	1

Zákon o agresivnosti logické 0 a logické 1 v mnohém připomíná zákon předchozí. Zde ovšem platí, že při logické operaci OR vstupní hodnoty a logické 1 je výstupný hodnota vždy rovna hodnotě logické 1 nezávisle na hodnotě vstupní. Při logickém součinu vstupní hodnoty s logickou 0 je výstupní logická hodnota vždy rovna logické 0 a to rovněž bez ohledu na zadané hodnotě vstupní.

Tabulka 8 - Potvrzení pravdivosti zákona o agresivnosti logické 0 a logické 1

Agresivnost 0 a 1		
A	A0	A+1
0	0	1
1	0	1

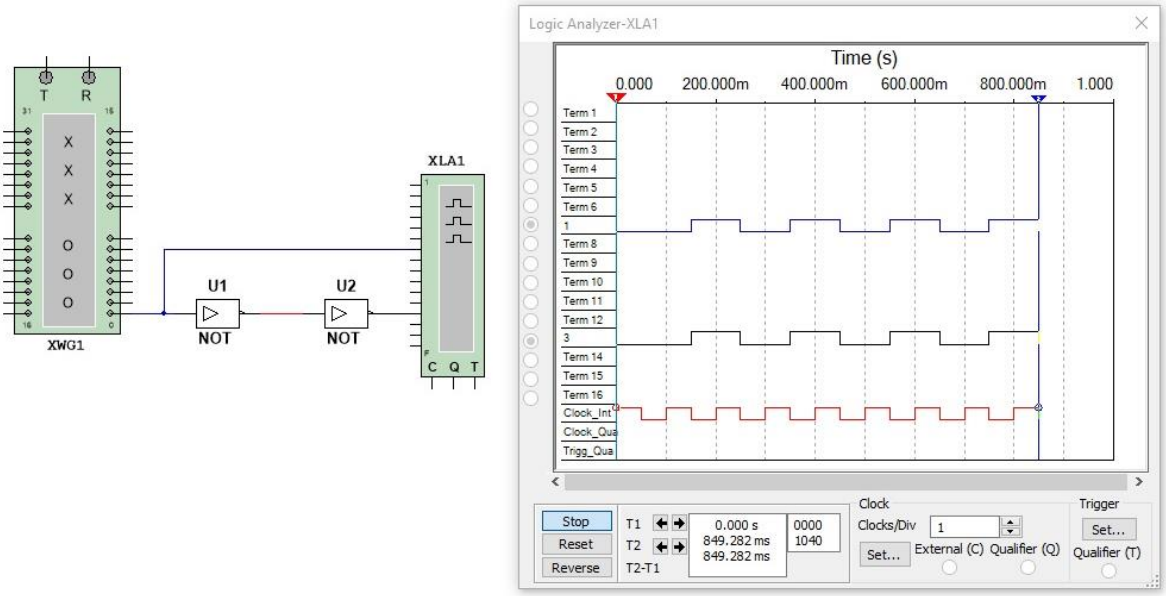
Zákon absorpce udává, že v případech logického součtu nebo logického součinu stejné vstupní proměnné bude výstupní proměnná rovna vstupní proměnné. Tento zákon lze aplikovat na neurčené množství opakujících se vstupních proměnných.

Tabulka 9 - Potvrzení pravdivosti zákona absorpce

Absorpce		
A	AA	A+A
0	0	0
1	1	1

Zákon dvojité negace znamená, že se vstupní hodnota neguje a tato hodnota se neguje znovu. V případě, pokud je vstupní hodnota A, pak po první negaci dostaneme hodnotu

negovaného A, negujeme-li znovu, tak výstupní hodnotou je hodnota A. Dle zákona dvojité negace je jasné, že výstupní hodnota je rovna hodnotě vstupní.



Obrázek 14 – Simulace a potvrzení zákona dvojité negace (Zdroj: vlastní)

De Morganovy zákony říkají, že po nahrazení všech proměnných v logické operaci jejich negacemi a po záměně operací součtu za součin, popř. naopak, získáme výsledek. Na následující simulaci je zřetelně vidět stejné hodnoty výstupních hodnot při úpravě logické funkce dle De Morgana (Michalík & Semrád, 2007). De Morganovy zákony lze zapsat ve tvaru:

$$\overline{A + B} = \bar{A}\bar{B}$$

$$\overline{AB} = \bar{A} + \bar{B}$$

De Morganovy zákony lze uplatnit na libovolný počet hodnot, je nutné však zachovat pozice závorek. Názorně si lze uplatnění De Morganova zákona a nutnosti zachování pozice závorek ukázat na následujícím příkladu.

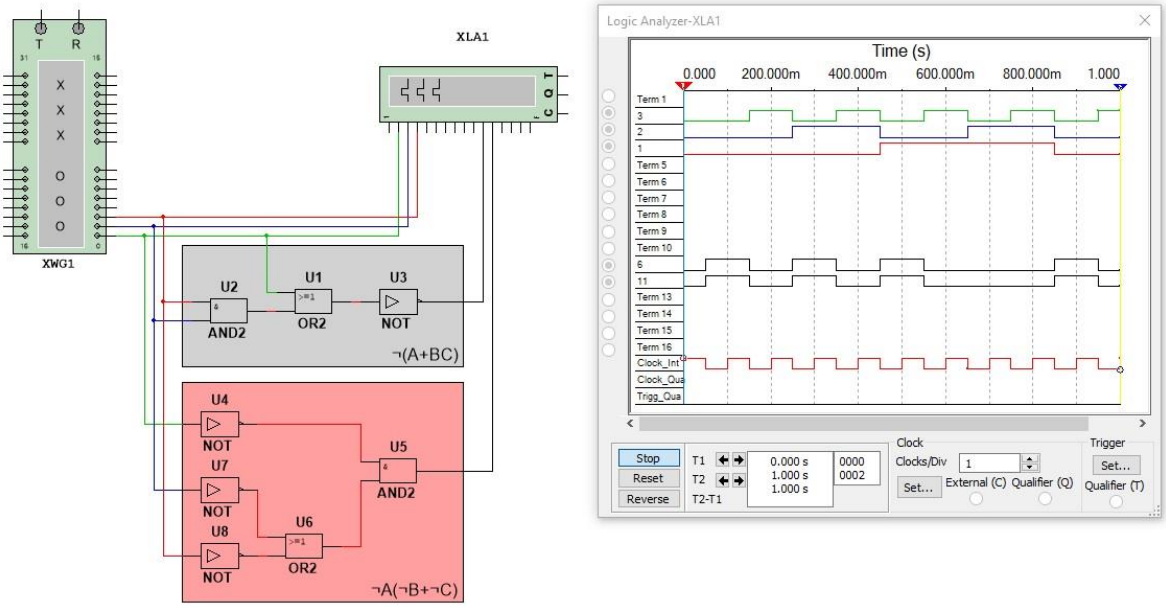
Je dána logická funkce (Kantnerová, 2010):

$$f = \overline{A + BC} => \bar{A}(\overline{BC}) => \bar{A}(\bar{B} + \bar{C})$$

Tabulka 10 - De Morganův zákon

De Morganův zákon				
C	B	A	¬(A+BC)	¬A(¬B+¬C)
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Na příkladu je vidět současné použití logického součinu i logického součtu. Stejně jako v matematice i v logické algebře má součin přednost před součtem, negace má pak stejnou prioritu, jakou má závorka (Kantnerová, 2010).



Obrázek 15 - Simulace a ověření De Morganových zákonů (Zdroj: vlastní)

2.2 MINIMALIZACE LOGICKÝCH FUNKCÍ

2.2.1 MINIMALIZACE LOGICKÝCH FUNKCÍ POMOCÍ ZÁKONŮ BOOLEOVY ALGEBRY

Příklady budou upravovány na základě zákonů Booleovy algebry. Postupně budou zvoleny příklady na procvičení všech zákonů Booleovy algebry, které byly vysvětleny v kapitole předchozí.

Je dána logická funkce (Kantnerová, 2010):

$$f = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$$

Příklad zjednodušíme následujícím způsobem. Za pomoci zákona distributivnosti vytkneme před závorku hodnoty tak, aby nám v závorce vznikl součet vstupní hodnoty s hodnotou negovanou.

$$f = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC = \bar{A}C(B + \bar{B}) + AC(B + \bar{B})$$

Závorku si můžeme nyní upravit dle zákona o vyloučení třetího stavu.

$$f = \bar{A}C(B + \bar{B}) + AC(B + \bar{B}) = \bar{A}C1 + AC1$$

Dále následuje úprava dle zákona o neutrálnosti logické 1. Při součinu logické 1 a vstupní hodnoty je výsledek vždy roven hodnotě vstupní.

$$f = \bar{A}C1 + AC1 = \bar{A}C + AC$$

Upravíme danou funkci vytčením vstupní hodnoty C před závorku.

$$f = \bar{A}C + AC = C(\bar{A} + A)$$

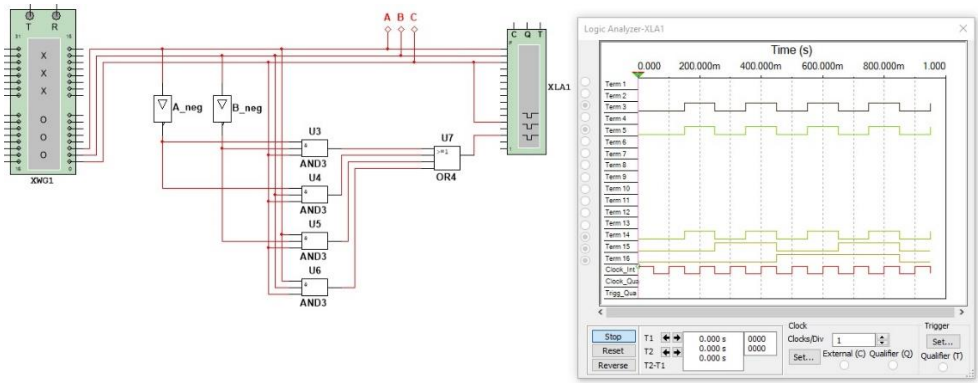
Opět dochází k aplikaci zákona o vyloučení třetího stavu.

$$f = C(\bar{A} + A) = C1$$

A upravujeme stejně jako v předešlém kroku dle zákona o neutrálnosti logické 1. Výsledkem je tedy pouze hodnota C.

$$f = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC = C$$

Na následujícím obrázku provedeme ověření tohoto příkladu v simulaci, zda minimalizace se výsledná minimalizace shoduje se zadanou logickou funkcí.



Obrázek 16 – Simulace minimalizace logické funkce pomocí zákonů Booleovy algebry (Zdroj: vlastní)

Dle simulace, která proběhla je patrné, že výstupní hodnota z neupravené logické funkce přesně kopíruje vstupní hodnotu C. Takto minimalizovaný vzorec je tedy již výsledný.

2.2.2 MINIMALIZACE LOGICKÝCH FUNKCÍ POMOCÍ KARNAUGHOVÝCH MAP

Dalším způsobem minimalizace logických funkcí je pomocí Karnaughových map.

Karnaughova mapa je čtverec nebo obdélník, který je dělen na 2^k polí. Hodnota *k* určuje počet vstupních proměnných. Mapa je v podstatě pravdivostní tabulka převedená do dvourozměrného pole.

Je dána logická funkce (Kantnerová, 2010):

$$f = (A + B)(\bar{A} + B + C)(A + C)(\bar{B} + C)$$

Vytvoříme si pravdivostní tabulku, která je nejběžnějším způsobem zápisu logické funkce. Pravdivostní tabulka ukazuje model chování systému, ale není v ní zakomponována realizace tohoto systému (Antošová & Davídek, 2009).

Pravdivostní tabulka k dané funkci vypadá následovně.

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

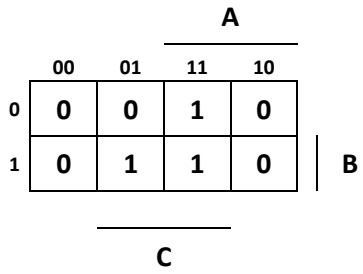
Tuto pravdivostní tabulku převedeme do dvourozměrného pole. Jestliže počet vstupních proměnných je roven 3, pak velikost tabulka bude 2^3=8.

Tvorba Karnaughovy mapy je jednoduchá a opakující se.

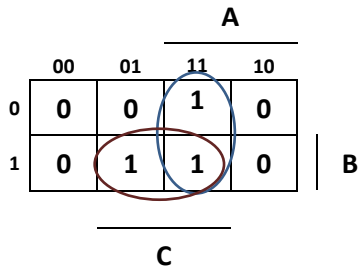
	00	01	11	10
0				
1				

Pomocné hodnoty 0 a 1, které jsou zvýrazněny červeně, se rovnají vstupní hodnotě A. Zelené pomocné hodnoty se rovnají vstupním hodnotám B a C. První pole má tedy dle pomocných hodnot číslo 000. To se rovná prvnímu řádku pravdivostní tabulky, do tohoto pole zapíšeme výstupní hodnotu f z prvního řádku. Druhé pole má již hodnotu 001, což se rovná druhému řádku pravdivostní tabulky, opět zapíšeme výstupní hodnotu f, tentokrát z druhého řádku, analogicky postupujeme dále.

Vytvořená Karnaughova mapa.



Nyní můžeme přejít k samotné minimalizaci. Z mapy přečteme smyčky, jejichž hodnoty jsou pro celou smyčku totožné. Vybíráme hodnoty rovné logické 1.

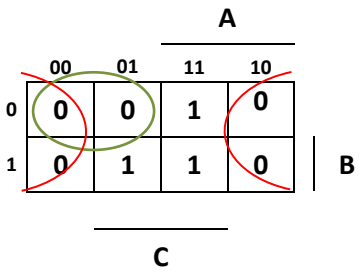


Smyčky plně zasahují do hodnot A a C (modrá smyčka) a hodnot B a C (hnědá smyčka). Výsledná logická funkce je tedy:

$$f_d = AC + BC$$

Toto je minimalizace dané logické funkce v disjunktivním tvaru.

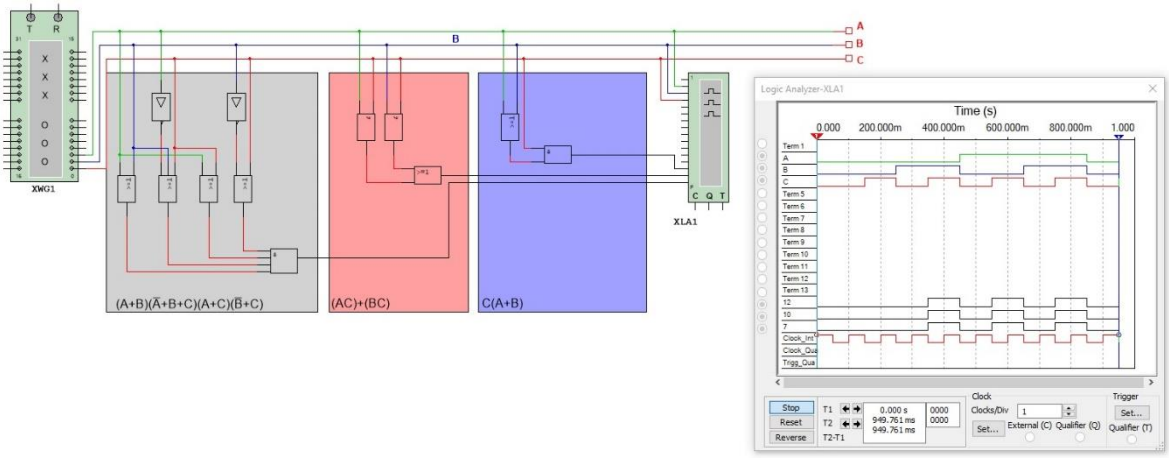
Minimalizaci logické funkce do konjunktivního tvaru provedeme obdobně. Z dané mapy vybereme ovšem hodnoty rovné logické 0.



Smyčky zde plně zasahují hodnoty negativního A a negativního B (zelená smyčka), druhá smyčka do hodnoty negativního C (červená smyčka). Na základě tohoto vytvoříme minimalizovaný tvar dané logické funkce v konjunktivním tvaru:

$$f_k = C(A + B)$$

V simulačním programu Multisim ověříme správnost minimalizace.



Obrázek 17 - Simulace a ověření minimalizace pomocí Karnaughových map (Zdroj: vlastní)

Na výše uvedené simulaci je patrná shodnost v jednotlivých výstupech logických funkcí. Tímto je ověřeno, že zadaná vstupní logická funkce je totožná s konjunktivním i disjunktivním minimalizovaným tvarem dané logické funkce.

Disjunktivní i konjunktivní minimalizované tvary logické funkce jsou si navzájem ekvivalentní. Lze provést zcela jednoduchou zkoušku tohoto tvrzení.

$$f_k = f_d$$

Po dosazení výše uvedené logické funkce by mělo pouhou úpravou minimalizované logické funkce v konjunktivním tvaru dojít k výsledku totožnému s minimalizovaným disjunktivním tvarem logické funkce.

$$C(A + B) = AC + BC$$

Minimalizovanou logickou funkci v konjunktivním tvaru nyní na základě pravidel Booleovy algebry upravíme. Nejdříve použijeme zákon distributivnosti, dle kterého dojde k odstranění závorek a roznásobení logické funkce.

$$C(A + B) = CA + CB$$

Díky zákonu komutativnosti pak upravíme tuto logickou funkci do následujícího tvaru.

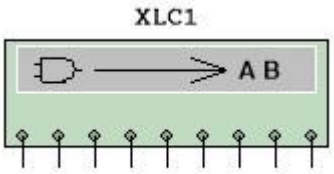
$$CA + CB = AC + BC$$

Tento výsledek je již shodný s předchozím výsledkem logické funkce v minimalizovaném disjunktivním tvaru. Na základě tohoto výpočtu došlo k potvrzení ekvivalence minimalizovaného disjunktivního a konjunktivního tvaru logické funkce.

2.2.3 MINIMALIZACE POMOCÍ SOFTWAROVÉHO ŘEŠENÍ

Mimo uvedené případy minimalizace logické funkce existuje ještě další, pro uživatele nejpohodlnější možnost. Touto možností je softwarové řešení minimalizace.

V rámci simulačního programu Multisim se nachází nástroj, který se nazývá logický konvertor. Tento nástroj umožní převádět schéma zapojení do pravdivostní tabulky, pravdivostní tabulku do logického výrazu nebo do zjednodušeného logického výrazu, dále logický výraz do pravdivostní tabulky a na schéma zapojení se zadanými typy hradel. Minimalizace je zde prováděna Quine-McCluskeyovou metodou (Juránek, 2008).



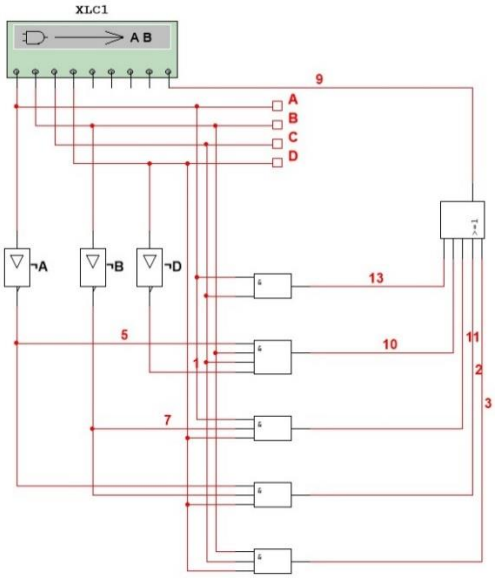
Obrázek 18 - Logický konvertor (Zdroj: vlastní)

Quine-McCLuskeyova metoda vyhledává maximální skupiny sousedních stavů. Tato metoda je vhodná pro minimalizaci většího počtu vstupních proměnných (Kantnerová, 2010).

Je dána logická funkce (Kantnerová, 2010):

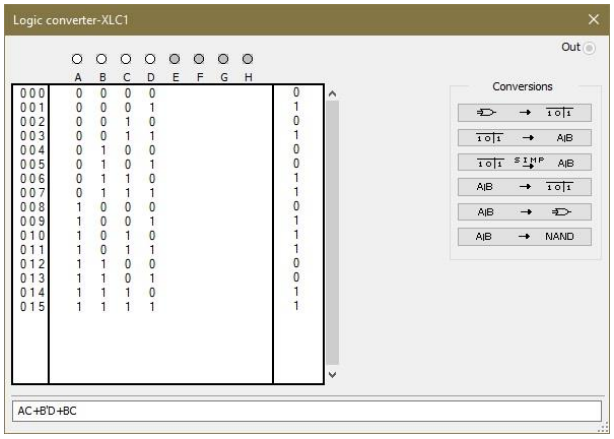
$$f = AC + \bar{A}BC\bar{D} + A\bar{B}D + \bar{A}\bar{B}D + BCD$$

V simulačním programu Multisim je vytvořeno schéma dle zadané logické funkce.



Obrázek 19 - Schéma logické funkce (Zdroj: vlastní)

Následně se v logickém konvertoru použije funkce pro vytvoření pravdivostní tabulky a rovněž i pro minimalizaci dané logické funkce. Výsledkem bude minimalizovaný tvar zadané logické funkce v disjunktním tvaru.



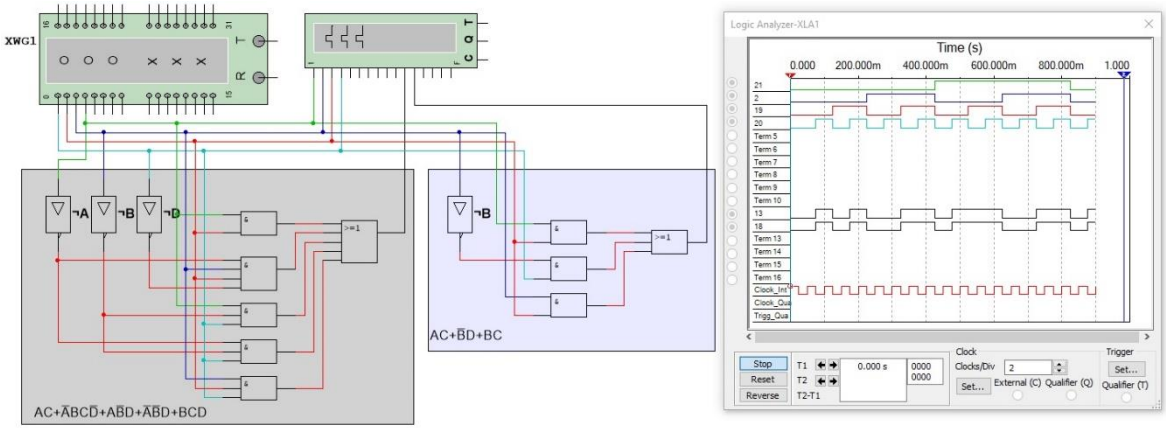
Obrázek 20 - Pravdivostní tabulka a minimalizovaná logická funkce (Zdroj: vlastní)

Další z možností je zadat v logickém konvertoru přímo nám známou pravdivostní tabulku. Na základě této tabulky lze vytvořit logickou funkci a rovněž minimalizovaný tvar logické funkce v disjunktivním tvaru. Logický konvertor umožňuje rovněž vytvořit zapojení s dvouvstupovými prvky.

Výsledek minimalizace zadané logické funkce je:

$$f = AC + \bar{B}D + BC$$

Nyní lze zadanou funkci a výslednou minimalizaci ověřit v simulačním programu.



Obrázek 21 - Simulace a ověření minimalizace pomocí programu Multisim (Zdroj: vlastní)

Výstupy ze zadané logické funkce a z minimalizovaného tvaru jsou totožné. Simulace tak potvrdila správnost výpočtu a zapojení.

3 SADA PŘÍKLADŮ PRO PODPORU VÝUKY LOGICKÉ ALGEBRY

3.1 PŘÍKLAD Č. 1

Je dána logická funkce:

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

Vytvořte minimalizovaný tvar logické funkce, ověřte pravdivost výsledku. Dokažte ekvivalenci disjunktivního a konjunktivního minimalizovaného tvaru logické funkce.

3.1.1 ŘEŠENÍ POMOCÍ ZÁKONŮ BOOLEOVY ALGEBRY

Použití zákonů:

- **zákon distributivnosti,**
- **zákon o agresivnosti 0 a 1,**
- **zákon o neutrálnosti 0 a 1.**

Pomocí zákona distributivnosti si vytkneme vhodné hodnoty před závorky.

$$f = \bar{a}c + \bar{a}bc + a\bar{b}\bar{c} + \bar{b}\bar{c} = \bar{a}c(1 + b) + \bar{b}\bar{c}(1 + a)$$

Na základě použití zákona o agresivnosti logické 0 a 1 odstraníme závorky.

$$f = \bar{a}c(1 + b) + \bar{b}\bar{c}(1 + a) = \bar{a}c1 + \bar{b}\bar{c}1$$

Zákon o neutrálnosti 0 a 1 definuje, že v případě logického součinu vstupní hodnoty s logickou hodnotou 1 bude výsledkem vždy vstupní logická hodnota.

$$f = \bar{a}c1 + \bar{b}\bar{c}1 = \bar{a}c + \bar{b}\bar{c}$$

Toto je již konečný minimalizovaný výsledek zadané logické funkce.

3.1.2 ŘEŠENÍ POMOCÍ KARNAUGHOVÝCH MAP, OVĚŘENÍ EKVIVALENCE

Vytvoříme pravdivostní tabulku, ze které v dalším kroku vytvoříme Karnaughovu mapu. Z této Karnaughovy mapy lze odvodit minimalizovaný tvar logické funkce v disjunktivním i konjunktivním tvaru.

Tabulka 11 - Příklad č. 1 - pravdivostní tabulka

a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Na základě pravdivostní tabulky si vytvoříme Karnaughovu mapu.

Tabulka 12 - Příklad č. 1 - Karnaughova mapa

		c			
				b	
		00	01	11	10
	0	1	1	1	0
a	1	1	0	0	0

Výsledek z této mapy jsou minimalizované tvary zadané funkce v disjunktivním a konjunktivním tvaru.

$$f_k = (\bar{a} + \bar{c})(\bar{b} + c)$$

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

Ověření ekvivalence proběhne úpravou minimalizovaného konjunktivního tvaru logické funkce. Důkaz ekvivalence proběhne tak, že se výsledek úpravy bude shodovat s disjunktivním minimalizovaným tvarem.

Provedeme úpravu minimalizovaného konjunktivního tvaru pomocí zákona distributivnosti.

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

Díky zákonu o vyloučení třetího stavu odstraníme jednu z hodnot.

$$f = \bar{a}\bar{b} + \bar{a}c + \bar{b}\bar{c} + \bar{c}c = \bar{a}\bar{b} + \bar{a}c + \bar{b}\bar{c} + 0$$

Tuto logickou funkci upravíme za pomoci zákona o neutrálnosti logické 0 a 1.

$$f = \bar{a}\bar{b} + \bar{a}c + \bar{b}\bar{c} + 0 = \bar{a}\bar{b} + \bar{a}c + \bar{b}\bar{c}$$

Nyní můžeme za využití zákona o vyloučení třetího stavu pokračovat.

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

Hodnotu $c + \bar{c}$ lze vložit do funkce bez změny výsledku. Tento člen je ve výsledku roven logické 1 a dle zákona o neutrálnosti logické 0 a 1 nebude mít tento člen na výsledek vliv.

Provedeme úpravu dle zákona distributivnosti.

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

Využijeme opět stejný zákon a vytkneme členy $\bar{a}c$ a $\bar{b}\bar{c}$.

$$f = \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}c + \bar{b}\bar{c} = \bar{a}c\left(\bar{b} + 1\right) + \bar{b}\bar{c}(\bar{a} + 1)$$

Provedeme úpravu dle zákona o agresivnosti logické 0 a 1.

$$f = \bar{a}c\left(\bar{b} + 1\right) + \bar{b}\bar{c}(\bar{a} + 1) = \bar{a}c1 + \bar{b}\bar{c}1$$

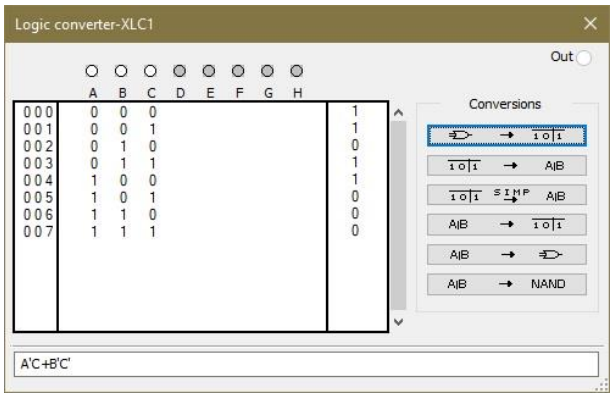
Posledním procesem je úprava dle zákona o neutrálnosti logické 0 a 1.

$$f = \bar{a}c1 + \bar{b}\bar{c}1 = \bar{a}c + \bar{b}\bar{c}$$

Tento výsledek již zcela odpovídá minimalizovanému disjunktivnímu tvaru dané logické funkce. Tímto je tedy ověřena ekvivalence mezi konjunktivním a disjunktivním minimalizovaném tvaru.

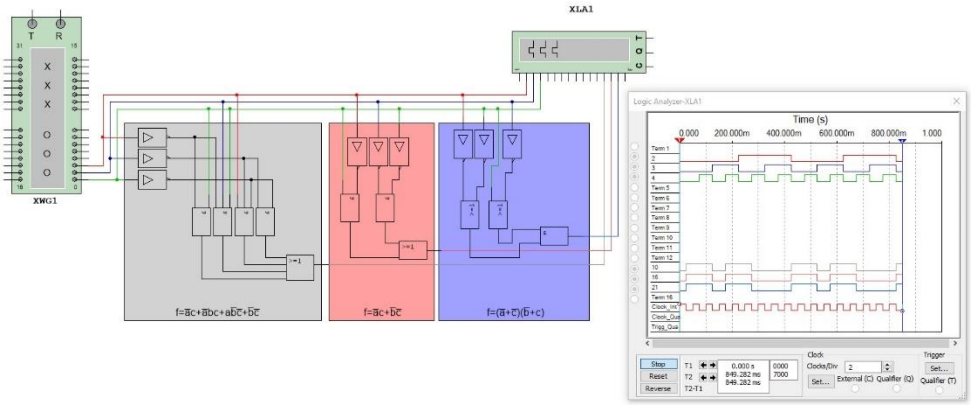
3.1.3 OVĚŘENÍ A SIMULACE PŘÍKLADU Č. 1

Zadanou funkci zpracujeme v prostředí simulačního programu Multisim. Pravdivostní tabulku zadáme do logického konvertoru a výsledkem bude minimalizovaný tvar logické funkce v disjunktivním tvaru.



Obrázek 22 - Příklad č. 1 - Logický konvertor (Zdroj: vlastní)

Minimalizovaný tvar logické funkce v disjunktivním tvaru koresponduje s tvarem, který nám vyšel v předchozích kapitolách.



Obrázek 23 - Příklad č. 1 - Simulace a ověření minimalizace (Zdroj: vlastní)

Simulací dané logické funkce byla zjištěna shoda mezi zadanou logickou funkcí a funkcí v disjunktivním i konjunktivním tvaru.

3.2 PŘÍKLAD Č. 2

Je dána logická funkce

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

Vytvořte minimalizovaný tvar logické funkce, ověřte pravdivost výsledku. Dokažte ekvivalenci disjunktivního a konjunktivního minimalizovaného tvaru logické funkce.

3.2.1 ŘEŠENÍ POMOCÍ ZÁKONŮ BOOLEOVY ALGEBRY

Použití zákonů:

- **De Morganův zákon,**
- **zákon distributivnosti,**

- **zákon absorpce,**
- **zákon o agresivnosti 0 a 1,**
- **zákon o neutrálnosti 0 a 1.**

Ke zjednodušení výpočtu rozdělíme logickou funkci. První závorku označíme jako logickou funkci f_1a za pomoci De Morganova zákona vypočteme.

$$f_1 = (a + \bar{a}b) = \overline{\overline{(a + \bar{a}b)}} = \overline{\bar{a}\overline{(\bar{a}b)}}$$

Na tento výsledek aplikujeme opět De Morganův teorém.

$$f_1 = \overline{\bar{a}\overline{(\bar{a}b)}} = \overline{\bar{a}(\bar{\bar{a}} + \bar{b})}$$

Použijeme-li zákon distribuce, dojde k odstranění závorek.

$$f_1 = \overline{\bar{a}(a + \bar{b})} = \overline{\bar{a}a + \bar{a}\bar{b}}$$

Pokračujeme za užití zákona o vyloučení třetího stavu.

$$f_1 = \overline{\bar{a}a + \bar{a}\bar{b}} = \overline{1 + \bar{a}\bar{b}}$$

V případě použití zákona o agresivnosti logické 0 a 1 dojde k odstranění hodnoty logické 1 z logické funkce.

$$f_1 = \overline{1 + \bar{a}\bar{b}} = \overline{\bar{a}\bar{b}}$$

Nyní opět použijeme De Morganův zákon.

$$f_1 = \overline{\bar{a}\bar{b}} = \bar{\bar{a}} + \bar{\bar{b}}$$

Poslední úpravou logické funkce f_1 bude úprava na základě zákona dvojité negace.

$$f_1 = \bar{\bar{a}} + \bar{\bar{b}} = a + b$$

Nyní se opět vrátíme k výchozí rovnici, kde místo první závorky vložíme výpočet logické funkce f_1.

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

Za pomoci zákona distributivnosti dojde k odstranění závorek.

$$f = (a + b)(a + \bar{c}) = aa + ab + a\bar{c} + b\bar{c}$$

Zákon absorpce nám minimalizuje jednu z hodnot.

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

Použijeme-li zákon distribuce, dojde k vytčení jedné z hodnot před závorku, kterou lze následně zjednodušit.

$$f = a + ab + a\bar{c} + b\bar{c} = a(1 + b + \bar{c}) + b\bar{c}$$

Na hodnoty v závorce aplikujeme zákon o agresivnosti 0 a 1.

$$f = a(1 + b + \bar{c}) + b\bar{c} = a1 + b\bar{c}$$

Poslední úpravou je za pomoci zákona o neutrálnosti 0 a 1.

$$f = a1 + b\bar{c} = a + b\bar{c}$$

Toto je již minimalizovaný tvar v disjunktivním tvaru.

3.2.2 ŘEŠENÍ POMOCÍ KARNAUGHOVÝCH MAP, OVĚŘENÍ EKVIVALENCE

Nejprve si vytvoříme pravdivostní tabulku reprezentující zadanou logickou funkci.

Tabulka 13 - Příklad č. 2 - pravdivostní tabulka

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Na základě této pravdivostní tabulky vytvoříme Karnaughovu mapu a následně z této mapy získáme minimalizovaný tvar zadané funkce v konjunktivním i disjunktivním tvaru.

Tabulka 14 - Příklad č. 2 - Karnaughova mapa

		c			
				b	
		00	01	11	10
a	0	0	0	0	1
	1	1	1	1	1

Z takto vytvořené Karnaughovy mapy již získáme oba tvary minimalizované funkce.

$$f_k = (a + b)(a + \bar{c})$$

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

Nyní provedeme ověření ekvivalence minimalizovaného disjunktivního a konjunktivního tvaru logické funkce. Důkaz ekvivalence provedeme tak, že pomocí zákonů Booleovy algebry upravíme minimalizovaný konjunktivní tvar logické funkce. Pokud bude výsledek této úpravy shodný s disjunktivním tvarem, pak jsme prokázali ekvivalenci obou minimalizovaných tvarů.

Pomocí zákona distribuce provedeme úpravu minimalizovaného konjunktivního tvaru.

$$f = (a + b)(a + \bar{c}) = aa + a\bar{c} + ab + b\bar{c}$$

Následně pomocí zákona absorpce odstraníme duplicitní hodnoty.

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

Použitím zákona distribuce provedeme vytknutí jedné z hodnot.

$$f = a + a\bar{c} + ab + b\bar{c} = a(1 + \bar{c} + b) + b\bar{c}$$

Hodnoty v závorce upravíme dle zákona o agresivnosti logické 0 a 1.

$$f = a(1 + \bar{c} + b) + b\bar{c} = a1 + b\bar{c}$$

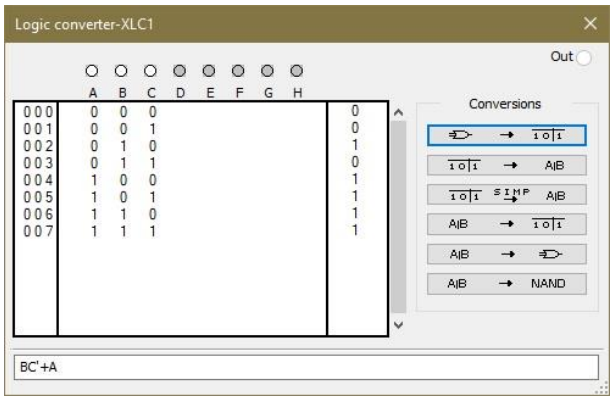
Zákon o neutrálnosti logické 0 a 1 nám pomůže odstranit z funkce hodnotu logické 1.

$$f = a1 + b\bar{c} = a + b\bar{c}$$

Tento výsledek již plně odpovídá disjunktivnímu minimalizovanému tvaru. Tímto je potvrzena ekvivalence obou minimalizovaných tvarů.

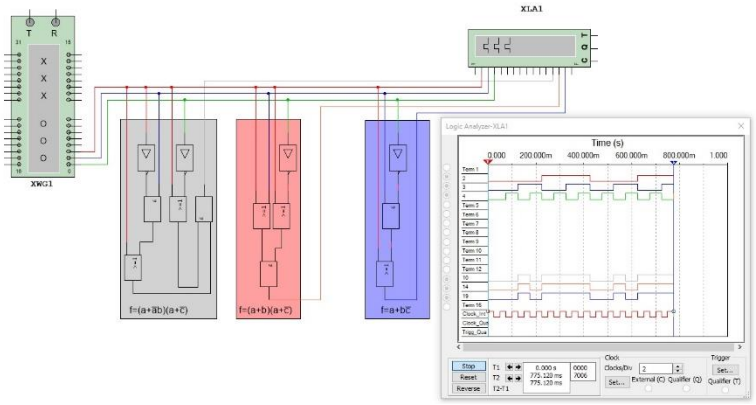
3.2.3 OVĚŘENÍ A SIMULACE PŘÍKLADU Č. 1

Pravdivostní tabulka z předchozího řešení byla zadána do logického konvertoru a pomocí jedné z jeho funkcí byla vytvořena minimalizovaná logická funkce v disjunktivním tvaru.



Obrázek 24 - Příklad č. 2 - Logický konvertor (Zdroj: vlastní)

Výsledný minimalizovaný tvar odpovídá plně výsledku z obou předchozích kapitol.



Obrázek 25 - Příklad č. 2 - Simulace a ověření minimalizace (Zdroj: vlastní)

V rámci simulace byla ověřena pravdivost minimalizovaných tvarů logické funkce v disjunktivním i konjunktivním tvaru na základě ekvivalence vstupu 10, což je zadaná logická funkce, a vstupů 14 a 19, které jsou minimalizovanými podobami zadané logické funkce.

3.3 PŘÍKLAD Č. 3

Je dána logická funkce:

$$f = ab\left(\bar{b} + c\right) + b(\bar{a} + bc) + \bar{a}b\bar{d} + \bar{b}c\bar{d}$$

Vytvořte minimalizovaný tvar logické funkce, ověřte pravdivost výsledku. Ověřte výpočtem ekvivalenci disjunktivního a konjunktivního minimalizovaného tvaru logické funkce.

3.3.1 ŘEŠENÍ POMOCÍ ZÁKONŮ BOOLEOVY ALGEBRY

Použití zákonů:

- **De Morganův zákon,**
- **zákon distributivnosti,**
- **zákon absorpce,**
- **zákon o vyloučení třetího stavu,**
- **zákon o agresivnosti 0 a 1,**
- **zákon o neutrálnosti 0 a 1,**
- **zákon dvojité negace.**

Díky zákonu distributivnosti dojde k odstranění závorek ze zadané logické funkce.

$$f = ab(\bar{b} + c) + b(\bar{a} + bc) + \bar{a}b\bar{d} + \bar{b}c\bar{d} = ab\bar{b} + abc + \bar{a}b + bbc + \bar{a}b\bar{d} + \bar{b}c\bar{d}$$

Za pomoci zákona absorpce se zbavíme dvojitých identických hodnot. Tento zákon nám definuje, že při logickém součinu či logickém součtu identických vstupních hodnot je výstupem vstupní logická hodnota.

$$f = ab\bar{b} + abc + \bar{a}b + bbc + \bar{a}b\bar{d} + \bar{b}c\bar{d} = ab\bar{b} + abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d}$$

Pokračujeme pomocí zákona o vyloučení třetího stavu.

$$f = ab\bar{b} + abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d} = \ a0 + abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d}$$

Zákon o agresivnosti 0 a 1 nám pomůže vyřešit část funkce, kde je součin logické hodnoty logické 0 a vstupní proměnné.

$$f = a0 + abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d} = 0 + abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d}$$

Za pomoci zákona o neutrálnosti logické 0 a 1 odstraníme hodnotu logické 0 z funkce.

$$f = 0 + abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d} = abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d}$$

Díky zákonu distributivnosti pokračujeme v další úpravě.

$$f = abc + \bar{a}b + bc + \bar{a}b\bar{d} + \bar{b}c\bar{d} = bc(a + 1) + \bar{a}b(1 + \bar{d}) + \bar{b}c\bar{d}$$

V případě obou závorek použijeme zákon o agresivnosti 0 a 1.

$$f = bc(a + 1) + \bar{a}b(1 + \bar{d}) + \bar{b}c\bar{d} = bc1 + \bar{a}b1 + \bar{b}c\bar{d}$$

Zákon o neutrálnosti logické 0 a 1 odstraní logické 1 z funkce.

$$f = bc1 + \bar{a}b1 + \bar{b}c\bar{d} = bc + \bar{a}b + \bar{b}c\bar{d}$$

Opět použijeme zákon distributivnosti, na jehož základě si vytkneme před závorku vstupní proměnnou c.

$$f = bc + \bar{a}b + \bar{b}c\bar{d} = c\left(b + \bar{b}\bar{d}\right) + \bar{a}b$$

Pro zjednodušení budeme opět pracovat pouze s částí funkce, kterou si pracovně nazveme logickou funkcí f_1.

$$f_1 = b + \bar{b}\bar{d}$$

Pomocí De Morganova zákona provedeme další úpravu. V úpravě funkce dále využijeme zákon absorpce, zákon dvojité negace, zákon o neutrálnosti logické 0 a 1, zákon o vyloučení třetího stavu a zákon distributivnosti.

$$f_1 = b + \bar{b}\bar{d} = \overline{\overline{b + \bar{b}\bar{d}}} = \overline{\bar{b}\left(\overline{\bar{b}\bar{d}}\right)} = \overline{\bar{b}\left(\bar{\bar{b}} + \bar{\bar{d}}\right)} = \overline{\bar{b}b + \bar{b}d} = 0 + \overline{\bar{b}d} = \overline{\bar{b}d} =$$

$$\bar{\bar{b}} + \bar{d} = b + \bar{d}$$

Pomocný výsledek funkce f_1 vložíme zpět do logické funkce.

$$f = c\left(b + \bar{b}\bar{d}\right) + \bar{a}b = c\left(b + \bar{d}\right) + \bar{a}b$$

Za pomoci zákona distributivnosti roznásobíme závorku.

$$f = c\left(b + \bar{d}\right) + \bar{a}b = bc + c\bar{d} + \bar{a}b$$

Toto je již minimalizovaný tvar logické funkce v disjunktivním tvaru.

3.3.2 ŘEŠENÍ POMOCÍ KARNAUGHOVÝCH MAP, OVĚŘENÍ EKVIVALENCE

Nyní si vytvoříme pravdivostní tabulku. V návaznosti na tuto tabulku poté vytvoříme Karnaughovu mapu.

Tabulka 15 - Příklad č. 3 - pravdivostní tabulka

a	b	c	d	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Z takto vytvořené pravdivostní tabulky lze již realizovat Karnaughovu mapu.

Tabulka 16 - Příklad č. 3 - Karnaughova mapa

		d			
				c	
		00	01	11	10
b	00	0	0	0	1
	01	1	1	1	1
a	11	0	0	1	1
	10	0	0	0	1

Z Karnaughovy mapy vytvoříme minimalizovanou funkci v konjunktivním i disjunktivním tvaru.

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

$$f_d = \bar{a}b + bc + c\bar{d}$$

Minimalizovaný tvar logické funkce v disjunktivním tvaru se plně shoduje s výsledkem z předchozí kapitoly.

Nyní provedeme ověření ekvivalence. Minimalizovaný tvar v konjunktivním tvaru roznásobíme za pomoci zákona o distributivnosti. Nejprve ale za pomoci zákona asociativnosti oddělíme dvojici pro snadnější výpočet.

$$f = (\bar{a} + c)(b + c)\big(b + \bar{d}\big) = \big((\bar{a} + c)(b + c)\big)\big(b + \bar{d}\big)$$

Nyní již přistoupíme za pomoci zákona distributivnosti k roznásobení.

$$f = \big((\bar{a} + c)(b + c)\big)\big(b + \bar{d}\big) = (\bar{a}b + \bar{a}c + bc + cc)\big(b + \bar{d}\big)$$

Zákon absorpce odstraní dvojici proměnné c.

$$f = (\bar{a}b + \bar{a}c + bc + cc)\big(b + \bar{d}\big) = (\bar{a}b + \bar{a}c + bc + c)\big(b + \bar{d}\big)$$

Pomocí zákona distributivnosti zjednodušíme první závorku.

$$f = (\bar{a}b + \bar{a}c + bc + c)\big(b + \bar{d}\big) = (\bar{a}b + c(\bar{a} + b + 1))\big(b + \bar{d}\big)$$

Zákon o agresivnosti 0 a 1 dále zjednoduší závorku.

$$f = (\bar{a}b + c(\bar{a} + b + 1))\big(b + \bar{d}\big) = (\bar{a}b + c1)\big(b + \bar{d}\big)$$

Díky zákonu o neutrálnosti logické 0 a 1 dojde ke konečné úpravě první závorky.

$$f = (\bar{a}b + c1)\big(b + \bar{d}\big) = (\bar{a}b + c)\big(b + \bar{d}\big)$$

Opět využijeme zákon distributivnosti a roznásobíme závorku.

$$f = (\bar{a}b + c)\big(b + \bar{d}\big) = \bar{a}bb + \bar{a}b\bar{d} + bc + c\bar{d}$$

Zákon absorpce upraví první hodnotu logické funkce.

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

Za pomoci zákona distributivnosti dále upravíme logickou funkci.

$$f = \bar{a}b + \bar{a}b\bar{d} + bc + c\bar{d} = \bar{a}b(1 + \bar{d}) + bc + c\bar{d}$$

Zákon o agresivnosti 0 a 1 pomůže odstranit závorku.

$$f = \bar{a}b\big(1 + \bar{d}\big) + bc + c\bar{d} = \bar{a}b1 + bc + c\bar{d}$$

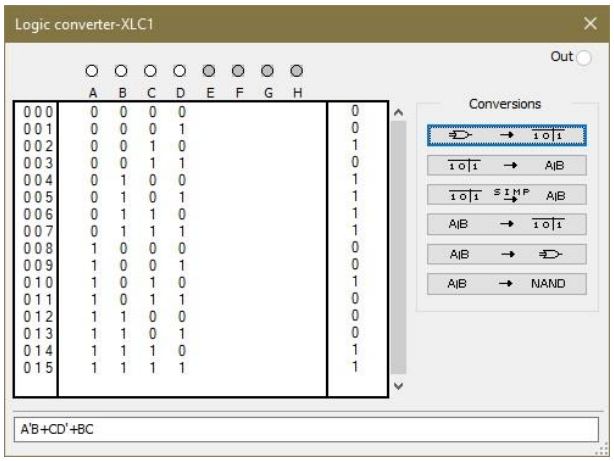
Poslední úprava pomocí zákona o neutrálnosti logické 0 a 1.

$$f = \bar{a}b1 + bc + c\bar{d} = \bar{a}b + bc + c\bar{d}$$

Z tohoto lze zřetelně vidět, že úpravou logické funkce v minimalizovaném konjunktivním tvaru jsme dospěli k podobě minimalizovaného disjunktivního tvaru logické funkce. Na základě tohoto je ověřena ekvivalence konjunktivního a disjunktivního tvaru logické funkce.

3.3.3 OVĚŘENÍ A SIMULACE PŘÍKLADU Č. 3

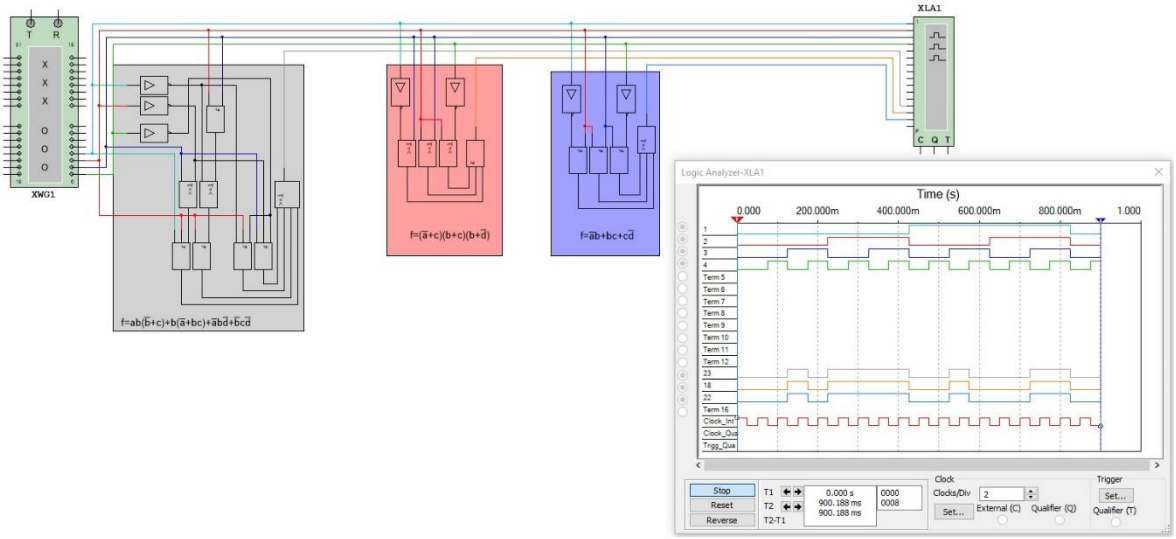
Do logického konvertoru zadáme pravdivostní tabulku, dle které simulační program vytvoří minimalizovanou podobu logické funkce v disjunktivním tvaru.



Obrázek 26 - Příklad č. 3 - Logický konvertor (Zdroj: vlastní)

Výsledek logického konvertoru se plně shoduje s výpočtem z předchozích kapitol.

Nyní provedeme zapojení v simulačním programu Multisim, a to jak zadané logické funkce, tak jejich minimalizovaných tvarů. Výstupní průběh by měl být naprosto totožný.



Obrázek 27 - Příklad č. 3 - Simulace a ověření minimalizace (Zdroj: vlastní)

Simulací jednotlivých výsledků a zadané logické funkce byla ověřena pravdivost jednotlivých výsledků.

3.4 PŘÍKLAD Č. 4

Je dána logická funkce:

$$f = a(\bar{b}c + c) + b\left((c + \bar{d})(a + \bar{b})\right) + a\bar{b}c\bar{d}$$

Proveďte minimalizaci zadané logické funkce. Ověřte správnost výsledků. Proveďte důkaz ekvivalence konjunktivního a disjunktivního minimalizovaného tvaru logické funkce.

3.4.1 ŘEŠENÍ POMOCÍ ZÁKONŮ BOOLEOVY ALGEBRY

Použití zákonů:

- **zákon distributivnosti,**
- **zákon o vyloučení třetího stavu,**
- **zákon o agresivnosti 0 a 1,**
- **zákon o neutrálnosti 0 a 1.**

Na logickou funkci aplikujeme zákon distributivnosti, a to na všechny závorky.

$$f = a(\bar{b}c + c) + b\left((c + \bar{d})(a + \bar{b})\right) + a\bar{b}c\bar{d}$$
$$= a\bar{b}c + ac + b(ac + \bar{b}c + a\bar{d} + \bar{b}d) + a\bar{b}c\bar{d}$$

$$f = a\bar{b}c + ac + b(ac + \bar{b}c + a\bar{d} + \bar{b}d) + a\bar{b}c\bar{d}$$
$$= a\bar{b}c + ac + abc + b\bar{b}c + ab\bar{d} + b\bar{b}d + a\bar{b}c\bar{d}$$

Na takto upravenou logickou funkci nyní aplikujeme zákon o vyloučení třetího stavu.

$$f = a\bar{b}c + ac + abc + b\bar{b}c + ab\bar{d} + b\bar{b}d + a\bar{b}c\bar{d}$$
$$= a\bar{b}c + ac + abc + 0c + ab\bar{d} + 0d + a\bar{b}c\bar{d}$$

Díky zákonu o agresivnosti logické 0 a 1 zjednodušíme další členy logické funkce.

$$f = a\bar{b}c + ac + abc + 0c + ab\bar{d} + 0d + a\bar{b}c\bar{d}$$
$$= a\bar{b}c + ac + abc + 0 + ab\bar{d} + 0 + a\bar{b}c\bar{d}$$

Zákon o neutrálnosti logické 0 a 1 nám pomůže odstranit logické 0 z funkce.

$$f = a\bar{b}c + ac + abc + 0 + ab\bar{d} + 0 + a\bar{b}c\bar{d} = a\bar{b}c + ac + abc + ab\bar{d} + a\bar{b}c\bar{d}$$

Pomocí zákona distributivnosti vytkneme před závorku člen ac.

$$f = a\bar{b}c + ac + abc + ab\bar{d} + a\bar{b}c\bar{d} = ac\left(\bar{b} + 1 + b + \bar{b}\bar{d}\right) + ab\bar{d}$$

Zákon o agresivnosti logické 0 a 1 nám pomůže zjednodušit závorku.

$$f = ac\left(\bar{b} + 1 + b + \bar{b}\bar{d}\right) + ab\bar{d} = ac1 + ab\bar{d}$$

Poslední úpravou logické funkce bude za pomoci zákona o neutrálnosti logické 0 a 1.

$$f = ac1 + ab\bar{d} = ac + ab\bar{d}$$

Výsledkem je již minimalizovaný disjunktivní tvar zadané logické funkce.

3.4.2 ŘEŠENÍ POMOCÍ KARNAUGHOVÝCH MAP, OVĚŘENÍ EKVIVALENCE

Prvním krokem je vytvoření pravdivostní tabulky.

Tabulka 17 - Příklad č. 4 - pravdivostní tabulka

a	b	c	d	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Z takto vytvořené pravdivostní tabulky realizujeme Karnaughovu mapu.

Tabulka 18 - Příklad č. 4 - Karnaughova mapa

		d			
				c	
		00	01	11	10
b	00	0	0	0	0
	01	0	0	0	0
a	11	1	0	1	1
	10	0	0	1	1

Z Karnaughovy mapy vytvoříme minimalizovanou funkci v konjunktivním i disjunktivním tvaru.

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

$$f_d = ab\bar{d} + ac$$

Minimalizovaný tvar logické funkce v disjunktivním tvaru odpovídá výsledku úpravy zadané logické funkce v předchozí kapitole.

Důkaz ekvivalence konjunktivního a disjunktivního minimalizovaného tvaru logické funkce provedeme, že minimalizovaný konjunktivní tvar logické funkce upravíme dle zákonů Booleovy algebry. Důkaz ekvivalence proběhne díky shodnému výsledku s minimalizovaným disjunktivním tvarem logické funkce.

Provedeme prvotní úpravu dle zákona distributivnosti. Nejprve roznásobíme závorky, následně dojde k roznásobení výsledku s hodnotou a.

$$f = a\left(c + \bar{d}\right)(b + c) = a(bc + cc + b\bar{d} + c\bar{d})$$

$$f = a(bc + cc + b\bar{d} + c\bar{d}) = abc + acc + ab\bar{d} + ac\bar{d}$$

Nyní využijeme zákon absorpce.

$$f = abc + acc + ab\bar{d} + ac\bar{d} = abc + ac + ab\bar{d} + ac\bar{d}$$

Znovu využijeme zákon distributivnosti.

$$f = abc + ac + ab\bar{d} + ac\bar{d} = ac\left(b + 1 + \bar{d}\right) + ab\bar{d}$$

Zákon o agresivnosti logické 0 a 1 pomůže s odstraněním závorky.

$$f = ac\left(b + 1 + \bar{d}\right) + ab\bar{d} = ac1 + ab\bar{d}$$

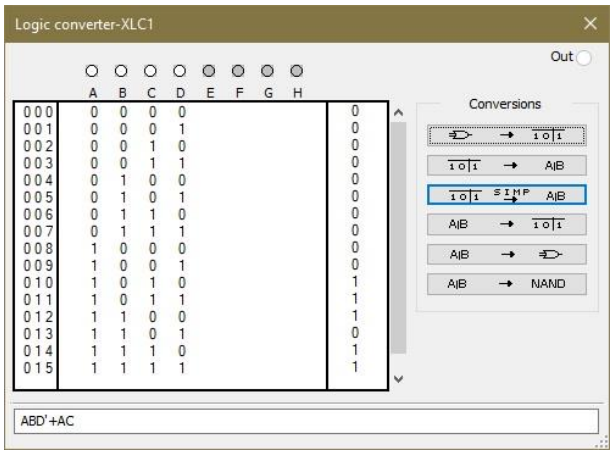
Díky zákonu o neutrálnosti logické 0 a 1 dojde k poslední úpravě logické funkce.

$$f = ac1 + ab\bar{d} = ac + ab\bar{d}$$

Na základě shody výsledku s minimalizovaným disjunktivním tvarem logické funkce můžeme konstatovat, že došlo k ověření ekvivalence mezi konjunktivním a disjunktivním minimalizovaném tvaru logické funkce.

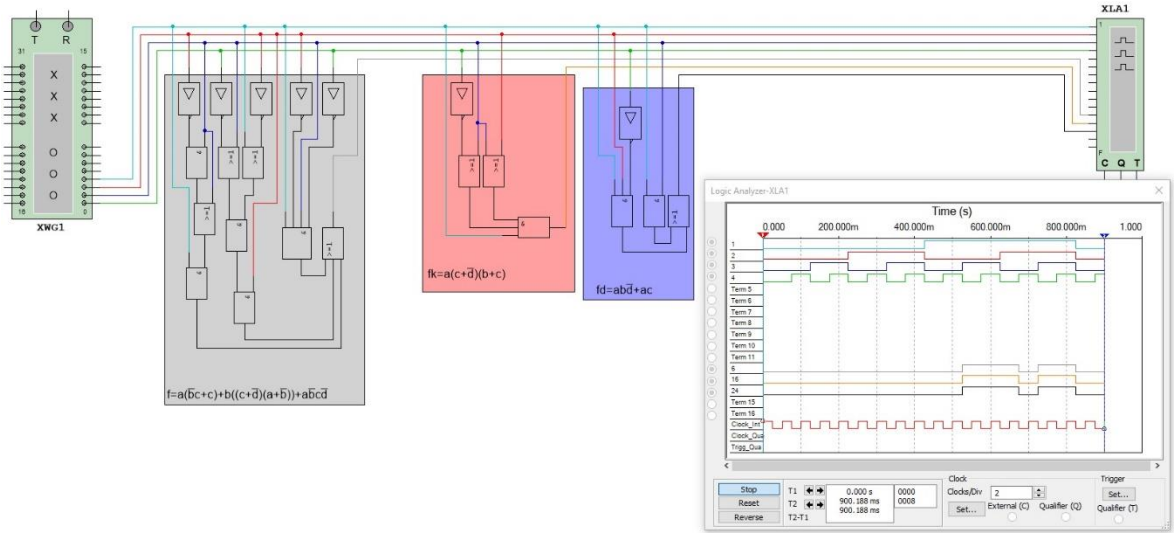
3.4.3 OVĚŘENÍ A SIMULACE PŘÍKLADU Č. 4

Nejprve zadáme pravdivostní tabulku do logického konvertoru a následně porovnáme výsledek minimalizace provedené logickým konvertorem s námi vypočítanými hodnotami.



Obrázek 28 - Příklad č. 4 - Logický konvertor (Zdroj: vlastní)

Logická funkce v minimalizovaném tvaru plně odpovídá přechozím výpočtům, stejně jako pravdivostní tabulka.



Obrázek 29 - Příklad č. 4 - Simulace a ověření minimalizace (Zdroj: vlastní)

V rámci simulace proběhlo ověření výsledků disjunktivního i konjunktivního minimalizovaného tvaru zadané logické funkce.

3.5 PŘÍKLAD Č. 5

Je dána logická funkce:

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

Proveďte minimalizaci zadané logické funkce. Ověřte správnost výsledků. Ověřte ekvivalenci minimalizovaných tvarů.

3.5.1 ŘEŠENÍ POMOCÍ ZÁKONŮ BOOLEOVY ALGEBRY

Použití zákonů:

- **zákon asociativnosti,**
- **zákon distributivnosti,**
- **zákon absorpce,**
- **zákon o vyloučení třetího stavu,**
- **zákon o agresivnosti 0 a 1,**
- **zákon o neutrálnosti 0 a 1.**

Za pomoci zákona asociativnosti si rozdělíme funkci na menší celky z důvodu usnadnění výpočtu. Pomocné funkce si označíme f_1 a f_2.

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

$$f = f_1 f_2$$

$$f_1 = (a + b)(\bar{c} + d)$$

$$f_2 = (a + d)(c + a + b)$$

Postupně si vypočítáme pomocné funkce. Za pomoci zákona distributivnosti roznásobíme závorky.

$$f_1 = (a + b)(\bar{c} + d) = a\bar{c} + ad + b\bar{c} + bd$$

Tento tvar již nemůžeme dále nijak upravit. Přejdeme k upravení druhé pomocné funkce za využití stejného zákona.

$$f_2 = (a + d)(c + a + b) = ac + aa + ab + cd + ad + bd$$

Za využití zákona o absorpci zjednodušíme člen aa.

$$f_2 = ac + aa + ab + cd + ad + bd = ac + a + ab + cd + ad + bd$$

Nyní opětovně využijeme zákon distributivnosti a díky němu vytkneme člen a před závorku.

$$f_2 = ac + a + ab + cd + ad + bd = a(c + 1 + b + d) + cd + bd$$

Díky zákonu o agresivnosti 0 a 1 lze upravit závorku.

$$f_2 = a(c + 1 + b + d) + cd + bd = a1 + cd + bd$$

Poslední úpravou pomocné funkce je pomocí zákona o neutrálnosti 0 a 1.

$$f_2 = a1 + cd + bd = a + cd + bd$$

Nyní vynásobíme výsledky pomocných funkcí f_1 a f_2 za pomoci zákona distributivnosti.

$$\begin{aligned} f &= (a\bar{c} + ad + b\bar{c} + bd)(a + cd + bd) \\ &= aa\bar{c} + a\bar{c}cd + ab\bar{c}d + aad + acdd + abcd + ab\bar{c} + b\bar{c}cd + bb\bar{c}d \\ &\quad + abd + bcdd + bbdd \end{aligned}$$

Nejprve k zjednodušení využijeme zákon absorpce.

$$\begin{aligned} f &= aa\bar{c} + a\bar{c}cd + ab\bar{c}d + aad + acdd + abcd + ab\bar{c} + b\bar{c}cd + bb\bar{c}d + abd + bcdd \\ &\quad + bbdd \\ &= a\bar{c} + a\bar{c}cd + ab\bar{c}d + ad + acd + abcd + ab\bar{c} + b\bar{c}cd + b\bar{c}d + abd \\ &\quad + bcd + bd \end{aligned}$$

Dalším krokem je využití zákona o vyloučení třetího stavu.

$$\begin{aligned} f &= a\bar{c} + a\bar{c}cd + ab\bar{c}d + ad + acd + abcd + ab\bar{c} + b\bar{c}cd + b\bar{c}d + abd + bcd + bd \\ &= a\bar{c} + 0 + ab\bar{c}d + ad + acd + abcd + ab\bar{c} + 0 + b\bar{c}d + abd + bcd \\ &\quad + bd \end{aligned}$$

Za pomoci zákona o neutrálnosti logické 0 a 1 odstraníme z funkce hodnotu logické 0.

$$\begin{aligned} f &= a\bar{c} + 0 + ab\bar{c}d + ad + acd + abcd + ab\bar{c} + 0 + b\bar{c}d + abd + bcd + bd \\ &= a\bar{c} + ab\bar{c}d + ad + acd + abcd + ab\bar{c} + b\bar{c}d + abd + bcd + bd \end{aligned}$$

Opětovně využijeme znalosti o zákonu distributivnosti.

$$\begin{aligned} f &= a\bar{c} + ab\bar{c}d + ad + acd + abcd + ab\bar{c} + b\bar{c}d + abd + bcd + bd \\ &= a\bar{c}(1 + bd + b) + ad(1 + c + bc + a) + bd(1 + \bar{c} + c) \end{aligned}$$

Na úpravu závorek použijeme zákon o agresivnosti 0 a 1.

$$f = a\bar{c}(1 + bd + b) + ad(1 + c + bc + a) + bd(1 + \bar{c} + c) = a\bar{c}1 + ad1 + bd1$$

Poslední úpravou bude užití zákona o neutrálnosti logické 0 a 1.

$$f = a\bar{c}1 + ad1 + bd1 = a\bar{c} + ad + bd$$

Nyní jsme dosáhli minimalizovaného disjunktivního tvaru zadané logické funkce.

3.5.2 ŘEŠENÍ POMOCÍ KARNAUGHOVÝCH MAP, OVĚŘENÍ EKVIVALENCE

Na základě pravdivostní tabulky si vytvoříme Karnaughovu mapu, ze které lze vyčíst minimalizovaný disjunktivní a minimalizovaný konjunktivní tvar zadané logické funkce.

Tabulka 19 - Příklad č. 5 - pravdivostní tabulka

a	b	c	d	f
0	0	0	0	0
0	0	0	1	0
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	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Z této pravdivostní tabulky vytvoříme Karnaughovu mapu, ze které poté dostaneme zadanou logickou funkci v minimalizovaném tvaru v disjunktivním i konjunktivním tvaru.

Tabulka 20 - Příklad č. 5 - Karnaughova mapa

		d			
				c	
		00	01	11	10
b	00	0	0	0	0
	01	0	1	1	0
a	11	1	1	1	0
	10	1	1	1	0

Následují výsledky minimalizace pomocí Karnaughových map v disjunktivním i konjunktivním tvaru.

$$f_k = (a + b)(a + d)(\bar{c} + d)$$
$$f_d = a\bar{c} + ad + bd$$

Výsledný minimalizovaný tvar logické funkce v disjunktivním tvaru zcela odpovídá výsledku z předchozí kapitoly.

Ekvivalence se prokáže úpravou konjunktivního tvaru. V případě, že dojde k shodě výsledku s disjunktivním tvarem, pak proběhl důkaz ekvivalence obou logických funkcí.

Nejprve pomocí zákona distributivnosti provedeme roznásobení první a druhé závorky.

$$f = (a + b)(a + d)(\bar{c} + d) = (aa + ad + ab + bd)(\bar{c} + d)$$

Na základě zákona absorpce odstraníme duality v členu aa.

$$f = (aa + ad + ab + bd)(\bar{c} + d) = (a + ad + ab + bd)(\bar{c} + d)$$

Nyní využijeme zákon distributivnosti.

$$f = (a + ad + ab + bd)(\bar{c} + d) = (a(1 + d + b) + bd)(\bar{c} + d)$$

Zákon o agresivnosti logické 0 a 1 následovaný zákonem o neutrálnosti logické 0 a 1 pomůže k další úpravě.

$$f = (a(1 + d + b) + bd)(\bar{c} + d) = (a1 + bd)(\bar{c} + d) = (a + bd)(\bar{c} + d)$$

Opětovně využijeme zákon distributivnosti a roznásobíme závorku.

$$f = (a + bd)(\bar{c} + d) = a\bar{c} + ad + b\bar{c}d + bdd$$

Další úprava logické funkce je na základě zákona absorpce.

$$f = a\bar{c} + ad + b\bar{c}d + bdd = a\bar{c} + ad + b\bar{c}d + bd$$

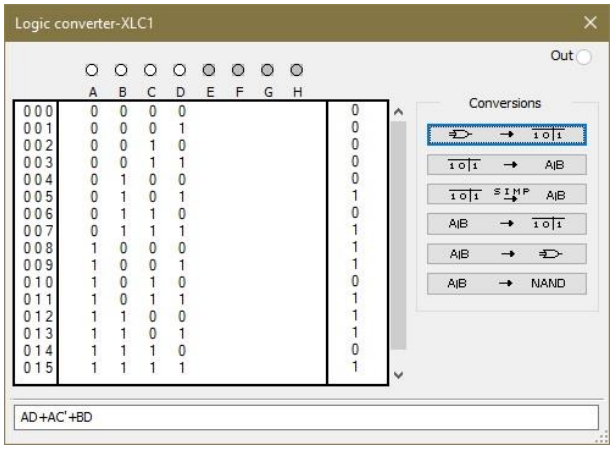
Poslední úpravou je za pomoci zákona distributivnosti, následovaný zákonem o agresivnosti logické 0 a 1 a zákonem o neutrálnosti logické 0 a 1.

$$f = a\bar{c} + ad + b\bar{c}d + bd = a\bar{c} + ad + bd(1 + \bar{c}\,) = a\bar{c} + ad + bd1 = a\bar{c} + ad + bd$$

Tento výsledek je již shodný s minimalizovaným disjunktivním tvarem logické funkce. Tím je ověřena ekvivalence mezi oběma tvary.

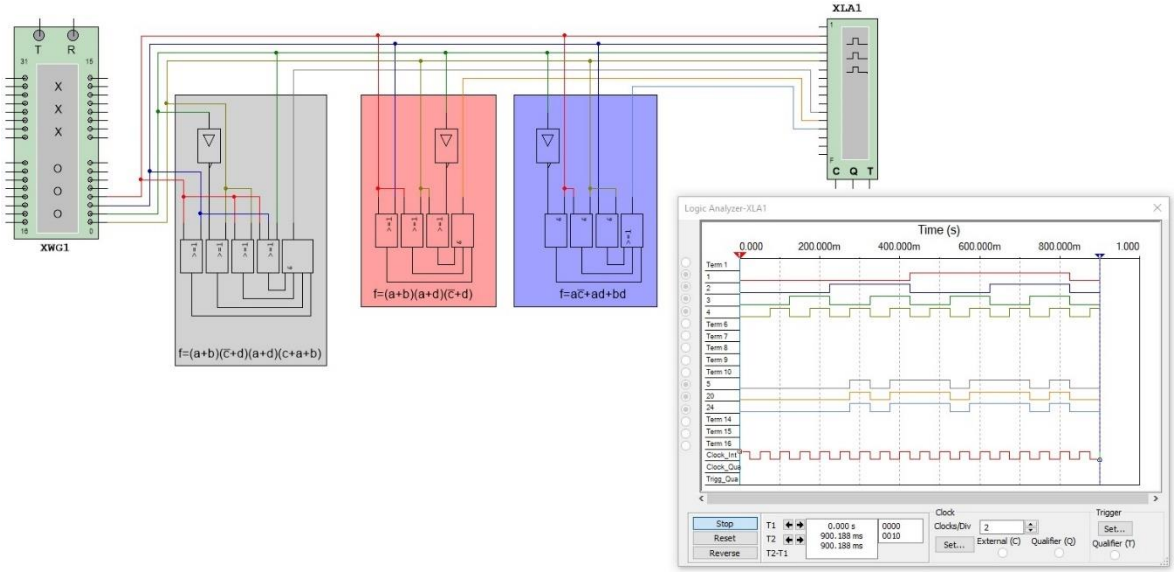
3.5.3 OVĚŘENÍ A SIMULACE PŘÍKLADU Č. 5

Pravdivostní tabulku převedeme do logického konvertoru. Výsledkem by měl být minimalizovaný disjunktivní tvar logické funkce.



Obrázek 30 - Příklad č. 5 - Logický konvertor (Zdroj: vlastní)

Minimalizovaný disjunktivní tvar logické funkce odpovídá předchozím výsledkům



Obrázek 31 - Příklad č. 5 - Simulace a ověření minimalizace (Zdroj: vlastní)

Na základě simulace došlo k ověření pravdivosti výsledků. Všechny výstupní hodnoty jsou identické, to značí, že jak zadaná logická funkce, tak i minimalizované tvary jsou si navzájem ekvivalentní.

3.6 PŘÍKLAD Č. 6

Je dána logická funkce:

$$f = \Big(\big(a + \bar{b}\big)(c + b)\Big)\big(\bar{a} + \overline{b + d}\big) + \big(\overline{(\bar{a} + d)(b + \bar{c})}\big)$$

Proveďte minimalizaci zadané logické funkce. Dokaž ekvivalenci minimalizovaného disjunktivního a minimalizovaného konjunktivního tvaru logické funkce. Proveďte ověření správnosti výsledků.

3.6.1 ŘEŠENÍ POMOCÍ ZÁKONŮ BOOLEOVY ALGEBRY

Použití zákonů:

- **De Morganův zákon,**
- **zákon dvojité negace,**
- **zákon distributivnosti,**
- **zákon absorpce,**
- **zákon o vyloučení třetího stavu,**
- **zákon o agresivnosti 0 a 1,**
- **zákon o neutrálnosti 0 a 1.**

V rámci ulehčení výpočtu lze výpočty provádět postupně. Vytvoříme si tak pomocné funkce f_1 a f_2, které po výpočtu dosadíme zpět do vstupní funkce.

$$f = \Big(\big(a + \bar{b}\big)(c + b)\Big)\big(\bar{a} + \overline{b + d}\big) + \big(\overline{(\bar{a} + d)(b + \bar{c})}\big)$$

Začneme vytvořením pomocné funkce f_1.

$$f_1 = \big(a + \bar{b}\big)(c + b)$$

Pomocí zákona distributivnosti roznásobíme závorky.

$$f_1 = \big(a + \bar{b}\big)(c + b) = ac + ab + \bar{b}c + \bar{b}b$$

Dle zákona o vyloučení třetího stavu můžeme tuto funkci dále upravit.

$$f_1 = ac + ab + \bar{b}c + \bar{b}b = ac + ab + \bar{b}c + 0$$

Poslední úpravou této funkce je za použití zákona o neutrálnosti logické 0 a 1.

$$f_1 = ac + ab + \bar{b}c + 0 = ac + ab + \bar{b}c$$

Pokračujeme vytvořením pomocné funkce f_2.

$$f_2 = \overline{(\bar{a} + d)(b + \bar{c})}$$

Nyní použijeme De Morganův zákon.

$$f_2 = \overline{(\bar{a} + d)(b + \bar{c})} = \overline{(\bar{a} + d)} + \overline{(b + \bar{c})}$$

Postup budeme nyní opakovat za užití stejného zákona.

$$f_2 = \overline{(\bar{a} + d)} + \overline{(b + \bar{c})} = \bar{\bar{a}}\bar{d} + \bar{b}\bar{\bar{c}}$$

Poslední úpravou této pomocné funkce je za využití zákona o dvojité negaci.

$$f_2 = \bar{\bar{a}}\bar{d} + \bar{b}\bar{\bar{c}} = a\bar{d} + \bar{b}c$$

Tyto vzniklé vypočítané pomocné funkce vložíme zpět do zadané logické funkce.

$$f = \left(ac + ab + \bar{b}c\right)\left(\bar{a} + \overline{b + d}\right) + \ a\bar{d} + \bar{b}c$$

V druhé závorce využijeme znalosti De Morganova zákona.

$$f = \left(ac + ab + \bar{b}c\right)\left(\bar{a} + \overline{b + d}\right) + \ a\bar{d} + \bar{b}c = \left(ac + ab + \bar{b}c\right)\left(\bar{a} + \bar{b}\bar{d}\right) + \ a\bar{d} + \bar{b}c$$

Nyní již můžeme využít zákon distributivnosti a roznásobit závorky v dané logické funkci.

$$\begin{aligned} f &= \left(ac + ab + \bar{b}c\right)\left(\bar{a} + \bar{b}\bar{d}\right) + \ a\bar{d} + \bar{b}c \\ &= a\bar{a}c + a\bar{b}c\bar{d} + a\bar{a}b + ab\bar{b}\bar{d} + \bar{a}\bar{b}c + \bar{b}\bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c \end{aligned}$$

Upravíme logickou funkci za využití zákona absorpce.

$$\begin{aligned} f &= a\bar{a}c + a\bar{b}c\bar{d} + a\bar{a}b + ab\bar{b}\bar{d} + \bar{a}\bar{b}c + \bar{b}\bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c \\ &= a\bar{a}c + a\bar{b}c\bar{d} + a\bar{a}b + ab\bar{b}\bar{d} + \bar{a}\bar{b}c + \bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c \end{aligned}$$

Dalším krokem bude využití zákona o vyloučení třetího stavu.

$$\begin{aligned} f &= a\bar{a}c + a\bar{b}c\bar{d} + a\bar{a}b + ab\bar{b}\bar{d} + \bar{a}\bar{b}c + \bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c \\ &= 0c + a\bar{b}c\bar{d} + 0b + a0\bar{d} + \bar{a}\bar{b}c + \bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c \end{aligned}$$

Zákon o agresivnosti logické 0 a 1 nám pomůže minimalizovat danou logickou funkci.

$$\begin{aligned} f &= 0c + a\bar{b}c\bar{d} + 0b + a0\bar{d} + \bar{a}\bar{b}c + \bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c \\ &= 0 + a\bar{b}c\bar{d} + 0 + 0 + \bar{a}\bar{b}c + \bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c \end{aligned}$$

Nyní využijeme zákon o neutrálnosti logické 0 a 1.

$$f = 0 + a\bar{b}c\bar{d} + 0 + 0 + \bar{a}\bar{b}c + \bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c = a\bar{b}c\bar{d} + \bar{a}\bar{b}c + \bar{b}c\bar{d} + \ a\bar{d} + \bar{b}c$$

Na základě zákona distributivnosti vytkneme před závorku hodnotu $\bar{b}c$.

$$f = a\bar{b}c\bar{d} + \bar{a}\bar{b}c + \bar{b}c\bar{d} + a\bar{d} + \bar{b}c = \bar{b}c\left(a\bar{d} + \bar{a} + \bar{d} + 1\right) + a\bar{d}$$

Zjednodušíme závorku za pomoci zákona o agresivnosti logické 0 a 1.

$$f = \bar{b}c\left(a\bar{d} + \bar{a} + \bar{d} + 1\right) + a\bar{d} = \bar{b}c1 + a\bar{d}$$

Poslední úpravou je využití zákona o neutrálnosti logické 0 a 1.

$$f = \bar{b}c1 + a\bar{d} = \bar{b}c + a\bar{d}$$

Toto je již minimalizovaný disjunktivní tvar zadané logické funkce.

3.6.2 ŘEŠENÍ POMOCÍ KARNAUGHOVÝCH MAP, OVĚŘENÍ EKVIVALENCE

Přistoupíme k vytvoření pravdivostní tabulky a z ní následně vytvoříme Karnaughovu mapu. Na základě této mapy pak lze určit disjunktivní a konjunktivní tvar logické funkce.

Tabulka 21 - Příklad č. 6 - pravdivostní tabulka

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

Tuto pravdivostní tabulku pak zcela jednoduše přetvoříme do Karnaughovy mapy, ze které následně vytvoříme minimalizovanou logickou funkci v disjunktivním i konjunktivním tvaru.

Tabulka 22 - Příklad č. 6 - Karnaughova mapa

		d			
				c	
		00	01	11	10
b	00	0	0	1	1
	01	0	0	0	0
a	11	1	0	0	1
	10	1	0	1	1

Z této Karnaughovy mapy již lze odvodit minimalizované logické funkce v obou tvarech. Následně provedeme proces ověření ekvivalence mezi minimalizovaným disjunktivním a konjunktivním tvarem logické funkce.

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

$$f_d = a\bar{d} + \bar{b}c$$

Minimalizovaný disjunktivní tvar logické funkce je totožný s výsledkem z předchozí kapitoly.

Minimalizovaný konjunktivní tvar upravíme za pomoci zákonů Booleovy algebry a ověření proběhne na základě shody upravené logické funkce s minimalizovaným disjunktivním tvarem.

Pomoci zákona o distributivnosti provedeme roznásobení závorek. Díky zákonu asociativnosti provedeme postupné roznásobení po jednotlivých závorkách

$$f = (a + c)\left(\bar{b} + \bar{d}\right)\left(c + \bar{d}\right)\left(a + \bar{b}\right) = (a\bar{b} + a\bar{d} + \bar{b}c + c\bar{d})(ac + \bar{b}c + a\bar{d} + \bar{b}\bar{d})$$

$$\begin{aligned} f = \left(a\bar{b} + a\bar{d} + \bar{b}c + c\bar{d}\right)\left(ac + \bar{b}c + a\bar{d} + \bar{b}\bar{d}\right) \\ = aa\bar{b}c + a\bar{b}\bar{b}c + aa\bar{b}\bar{d} + a\bar{b}\bar{b}\bar{d} + aac\bar{d} + a\bar{b}c\bar{d} + aa\bar{d}\bar{d} + a\bar{b}\bar{d}\bar{d} \\ + a\bar{b}cc + \bar{b}\bar{b}cc + a\bar{b}c\bar{d} + \bar{b}\bar{b}c\bar{d} + acc\bar{d} + \bar{b}cc\bar{d} + ac\bar{d}\bar{d} + \bar{b}c\bar{d}\bar{d} \end{aligned}$$

Za pomoci zákona absorpce upravíme logickou funkci.

$$\begin{aligned} f = aa\bar{b}c + a\bar{b}\bar{b}c + aa\bar{b}\bar{d} + a\bar{b}\bar{b}\bar{d} + aac\bar{d} + a\bar{b}c\bar{d} + aa\bar{d}\bar{d} + a\bar{b}\bar{d}\bar{d} + a\bar{b}cc + \bar{b}\bar{b}cc \\ + a\bar{b}c\bar{d} + \bar{b}\bar{b}c\bar{d} + acc\bar{d} + \bar{b}cc\bar{d} + ac\bar{d}\bar{d} + \bar{b}c\bar{d}\bar{d} \\ = a\bar{b}c + a\bar{b}c + a\bar{b}\bar{d} + a\bar{b}\bar{d} + ac\bar{d} + a\bar{b}c\bar{d} + a\bar{d} + a\bar{b}\bar{d} + a\bar{b}c + \bar{b}c \\ + a\bar{b}c\bar{d} + \bar{b}c\bar{d} + ac\bar{d} + \bar{b}c\bar{d} + ac\bar{d} + \bar{b}c\bar{d} \end{aligned}$$

Znovu využijeme zákon absorpce k odstranění duplicit.

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

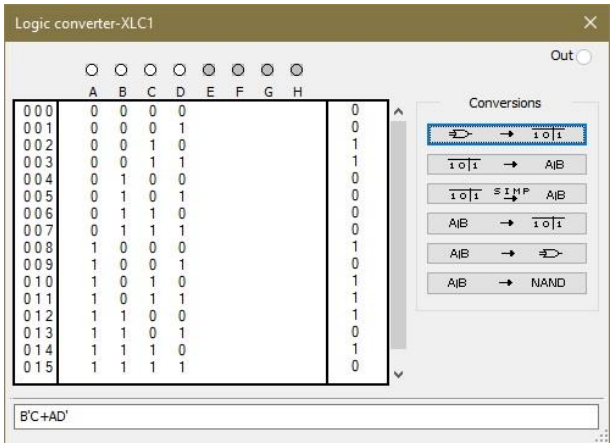
Nyní za pomoci zákona o distributivnosti, ve spojení se zákonem o agresivnosti logické 0 a logické 1, následovaný zákonem o neutrálnosti logické 0 a 1, upravíme logickou funkci.

$$f = a\bar{b}c + a\bar{b}\bar{d} + ac\bar{d} + a\bar{b}c\bar{d} + a\bar{d} + \bar{b}c + \bar{b}c\bar{d}$$
$$= \bar{b}c\left(a + a\bar{d} + 1 + \bar{d}\right) + a\bar{d}\left(\bar{b} + c + 1\right) = \bar{b}c1 + a\bar{d}1 = \bar{b}c + a\bar{d}$$

Tento výsledek již odpovídá minimalizovanému disjunktivnímu tvaru logické funkce. Tímto je tedy dokázána ekvivalence mezi konjunktivním a disjunktivním tvarem logické funkce.

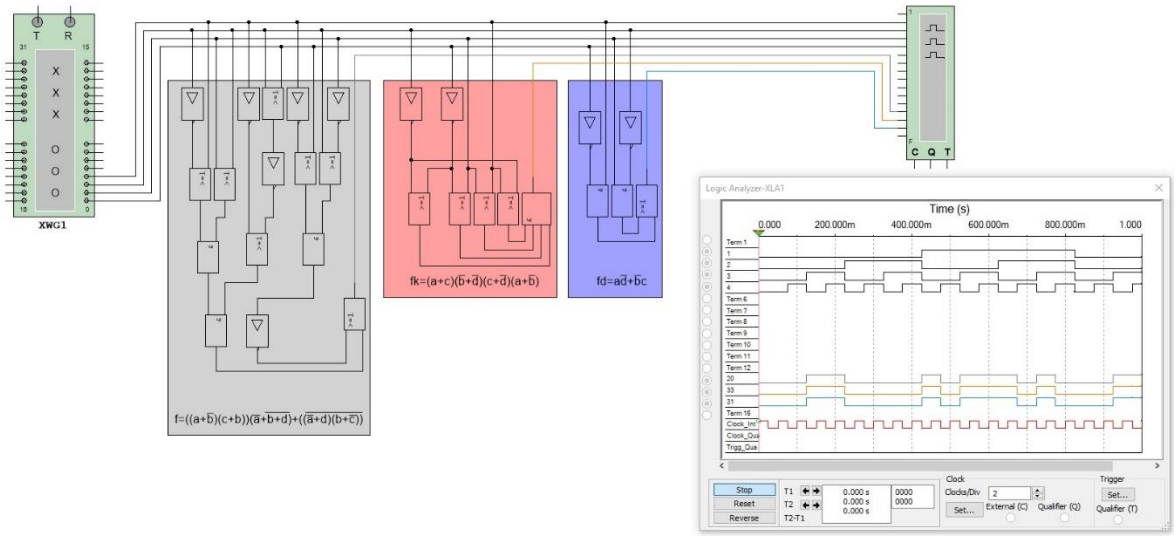
3.6.3 SIMULACE A OVĚŘENÍ PŘÍKLADU Č. 6

Do logického konvertoru zadáme pravdivostní tabulku, díky které vytvoří simulační program minimalizovaný disjunktivní tvar logické funkce.



Obrázek 32 - Příklad č. 6 - Logický konvertor (Zdroj: vlastní)

Z výše uvedeného vidíme shodu v pravdivostní tabulce. Minimalizovaný tvar logické funkce v disjunktivním tvaru je $f = \bar{b}c + a\bar{d}$, což je výsledek totožný s předchozím výsledkem $f_d = a\bar{d} + \bar{b}c$, jelikož dle zákona asociativnosti lze jednotlivé členy v rámci dané logické funkce přesouvat bez vlivu na výsledek logické funkce.



Obrázek 33 - Příklad č. 6 - Simulace a ověření minimalizace (Zdroj: vlastní)

V rámci simulace došlo k ověření výsledků minimalizace jak v disjunktivním, tak i v konjunktivním tvaru logické funkce. Dle výsledků minimalizace je patrné, že vstupní logická funkce je totožná s jejími minimalizovanými tvary.

ZÁVĚR

Hlavním cílem mé práce bylo představit sadu příkladů, které by sloužily k podpoře výuky logické algebry. Práce je rozdělena na dva bloky, kdy první z těchto bloků stručně představuje a charakterizuje logickou algebru jako celek a dále se zabývá zákony logické algebry. Tento teoretický blok není a nemá být výukovým materiálem k pochopení zákonitostí logické algebry, ale jde pouze o stručný přehled, který následně slouží pro vypracování příkladů. Druhým stěžejním blokem je sada příkladů, která má primárně za úkol podpořit výuku logické algebry. Složitost těchto příkladů má vzestupnou tendenci. Snažil jsem se postupně volit vyšší obtížnost výpočtů, i když téměř celé spektrum zákonů logické algebry obsahuje již první příklad. Gradující složitost sady je dána především postupem, kde je nutno častěji řetězit použití zákonů logické algebry.

Dílčím cílem byla snaha ukázat možnosti při řešení jednotlivých příkladů. Po zadání logické funkce následovalo řešení za pomoci zákonů logické algebry. Právě v této části je zřetelná gradace složitosti řešení zadaných logických funkcí. Další z možností řešení byla minimalizace za použití Karnaughovy mapy. Nejprve jsem si vždy vytvořil pravdivostní tabulku, jejíž vzor se nachází na přiloženém médiu. Pravdivostní tabulka byla vypočtena v tabulkovém procesoru, následně byla přenesena do této práce. Vytvoření pravdivostní tabulky bylo nutné ze dvou důvodů. Prvním je nutnost znalosti pravdivostní tabulky a její následné přenesení do Karnaughovy mapy. Z této mapy lze pak vyčíst minimalizované tvary zadaných logických funkcí. Další využití pravdivostní tabulky je přenesení do simulačního programu, kterým simuluji jednotlivé logické funkce, analyzuji vstupní a výstupní hodnoty. Právě díky simulačnímu programu, ve kterém je implementován logický konvertor, lze jednoduše odvodit minimalizovaný tvar logické funkce v disjunktivním tvaru. Díky logickému konvertoru lze také vytvořit zapojení, ale pouze s dvouvstupovými členy, která pro účely této práce nejsou příliš vhodná. Chtěl jsem zapojením přesně simulovat zadanou logickou funkci tak, aby byla vidět shoda mezi použitými prvky a členy v logické funkci. Dále jsem v rámci řešení příkladu zadal ověření ekvivalentního vztahu mezi konjunktivním a disjunktivním minimalizovaným tvarem logické funkce. Toto zadání bylo řešení úpravou konjunktivního tvaru a následný výsledek jsem porovnával s disjunktivním tvarem. V případě shody je prokázána ekvivalence. Posledním úkolem byl ověření řešení pomocí simulačního programu Multisim. V tomto

programu jsem simuloval jednotlivé logické funkce, tedy zadanou logickou funkci bez úpravy, dále minimalizovaný disjunktivní a minimalizovaný konjunktivní tvar. Analýzou vstupních a výstupních signálů pak došlo k ověření správnosti výsledků, a tedy i celkového postupu při výpočtech.

Tato práce, která se zabývá jednou z elementárních znalostí v oblasti informatiky, pro mě byla velmi poučným nástrojem, díky kterému jsem ještě více pronikl do hloubky tohoto tématu. V rámci vypracování jsem pracoval nejen s tabulkovým procesorem, ale i se simulačním programem Multisim, kde jsem realizoval zapojení jednotlivých příkladů.

Možné rozšíření této práce bych spatřoval v prezentaci dalších možností při řešení příkladů, stejně tak jako přidání další sady příkladů. Mezi jinými by se mohlo jednat o softwarové řešení výpočtů a online nástroje. Dále by byla možnost do větší hloubky propracovat teoretickou část této práce s důrazem na hlubší vysvětlení zákonů Booleovy algebry.

Tato práce přináší sadu, která může podpořit výuku logické algebry a je určena především pro studenty středních škol zabývajících se hlouběji daným tématem a také pro studenty vysokých škol, kde sada příkladů může být pomocným materiálem při studiu.

RESUMÉ

The thesis submits a set of examples for logical algebra support. In the theoretical part I have dealt with a brief description of logical algebra and laws of logical algebra. After that I have followed up the options of example solutions of asked logical functions which include the solution by usage of Boole's algebra laws and the solutions in assistance of Karnaugh's maps as well. The last option was to rework the truth-table into the logical convertor where one of the choices is the automatic minimization of asked logical function.

Within the practical part I have introduced six examples also with successive solution and the equivalence between the individual simulation results of all the logical functions was demonstrated here. I have tried to make the complexity of the set of examples with increasing tendency that a student drawing on my thesis could have a possibility gradually come through the principles and lawfulness of algebra.

The main task of the set of examples stated in the theses is to support teaching logical algebra in high schools and universities where could be used as a helpful teaching material.

SEZNAM LITERATURY

ANTOŠOVÁ, Marcela a Vratislav DAVÍDEK. *Číslicová technika: [učebnice]*. 4., aktualiz. vyd. České Budějovice: Kopp, 2009. ISBN 978-80-7232-394-4.

JURÁNEK, Antonín. *MultiSIM-elektronická laboratoř na PC: schémata a zapojení*. Praha: BEN-technická literatura, 2008. ISBN 978-80-7300-194-0.

KANTNEROVÁ, Ivana. *Sbírka příkladů z číslicové techniky*. V Praze: Idea servis, 2010. ISBN 978-80-85970-66-1.

LEBROVÁ, Dobromila. George Boole, britský matematik a logik - 145. výročí úmrtí. In: *Pozitivní noviny* [online]. 2019, 08. 12. 2009 [cit. 2019-03-26]. Dostupné z: https://www.pozitivni-noviny.cz/cz/clanek-2009120040

MAREŠ, Milan. *Příběhy matematiky: stručná historie královny věd*. Příbram: Pistorius & Olšanská, 2008. ISBN 978-80-87053-16-4.

MATOUŠEK, David. *Číslicová technika: základy konstruktérské praxe*. Praha: BEN, 2001. ISBN 80-7300-025-3.

PETR, Michalík a Semrád PETR. Číslicové prvky a systémy. *E-kurz: Číslicové prvky a systémy* [online]. Západočeská univerzita v Plzni, 2007, 08. 03. 2007 [cit. 2019-03-26]. Dostupné z: https://www.kvd.zcu.cz/cz/materialy/cps/cps/cps/

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

PŘÍLOHY

Příloha I. - digitální datové optické médium s elektronickou verzí práce, s vytvořenou sadou příkladů včetně řešení a zdrojových dat