

Cvičenia z princípov počítačov

Michal Ďurian

Cvičenie 2 – 1-bitový komparátor

2.1 Logické obvody

Pod logickými obvodmi chápeme také zapojenia, ktoré podľa určitých pravidiel pracujú s logickou informáciou, zakódovanou pomocou napäťových úrovní. Vo všeobecnosti môžeme pre ľubovoľnú, nie len logickú funkciu, písať:

$$y = f(a, b, c, \dots)$$

kde a, b, c sú vstupné parametre funkcie f a y predstavuje výstup z funkcie.

Z hľadiska funkcionality poznáme dva typy logických obvodov:

- **Kombinačné obvody** – Ich výstup závisí iba od vstupných hodnôt v danom okamihu. Sú to poväčšine obvody, ktoré realizujú matematické logické funkcie.
- **Sekvenčné obvody** – Ich výstup závisí okrem aktuálnych vstupov aj od predošlého stavu obvodu. Napríklad klopné a pamäťové obvody. Často, ale nie nevyhnutne, sa vyznačujú tým, že sú časované.

Integrované obvody

Pod pojmom **integrovaný obvod** sa dnes schováva ohromné množstvo súčiastok a elektronických komponentov rôznej veľkosti a komplexnosti. Integrované obvody, ktoré realizujú logické funkcie, patria medzi tie najjednoduchšie. V závislosti od technológie výroby sú zadelené do rodín, z ktorých každá má svoje označenie. Konkrétna súčiastka s logickou funkciou má v rámci rodiny zase svoje označenie. Napríklad označenie *MH7400* čítame nasledovne:

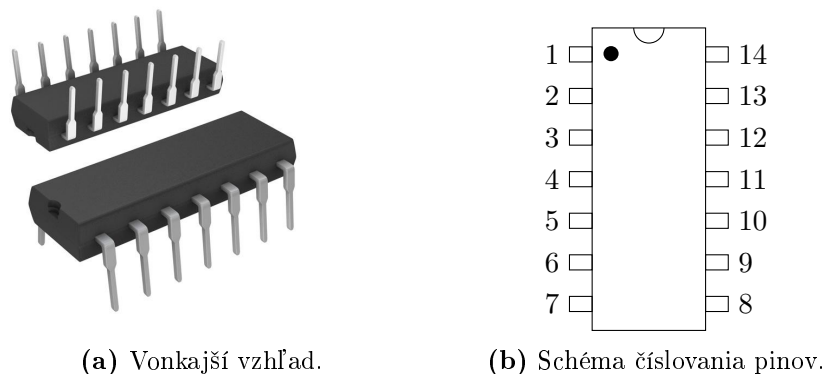
- Prvé dve miesta sú kód výrobcu. *MH* – Tesla, *SN* – Texas Instruments, atď...
- Ďalšie dve miesta označujú rodinu obvodov. *74* – logické obvody TTL, *40* – logické obvody CMOS, atď...
- Posledné dve miesta označujú konkrétny integrovaný obvod v rámci rodiny. *00* – Súčiastka obsahujúca štyri dvojvstupové hradlá NAND.

Postupom času, ako narastal počet rôznych logických integrovaných obvodov sa prispôbovalo aj značenie, ktoré často obsahuje ešte doplňujúce miesta v označovacom kóde. Napríklad, existuje súčasne vyrábaná súčiastka od firmy Texas Instruments obsahujúca 4 hradlá NAND, nesúca označenie *SN74LS00N*. Pôvodné prvky značenia sú zachované a rozpoznateľné. Nový údaj *LS* vyjadruje, že daná súčiastka vie pracovať s maximálnym

napájacím napätím 7V. Kód *N* je zase výrobcovo označenie puzdra (viď ďalší odsek), v ktorom sa logický obvod nachádza.

Samotné hradlá sú vyleptané na tzv. **jadre**, ktoré predstavuje polovodičový substrát, na ktorom boli technologickými postupmi vytvorené polovodičové prvky. Jadro je schované v **puzdre**, pričom kontakty jadra sú pripojené na nožičky puzdra. Obvody, s ktorými budeme pracovať, sú v puzdách typu **DIP14**¹, ktoré je na obrázku 2.1a. Číslo 14 v označení puzdra hovorí o počte nožičiek – **pinov**.

Kvôli tomu, aby sme vedeli na piny súčiastky priviesť správne signály, potrebujeme sa v nich orientovať. Na toto slúži ich číslovanie, ktoré je štandardizované a ukážeme si to na schéme Orientačným bodom na puzdre je výrez, na jednom z jeho koncov. Tento výrez nazývame **kľúč** (obr. 2.1a). Ak sa pozeráme na súčiastku zhora, tak krajnému pinu vľavo od kľúča prislúcha číslo 1. Často sa vedľa pinu číslo 1 nachádza aj bodka, ktorá niekedy nahrádza kľúč. Ostatné piny sú číslované proti smeru hodinových ručičiek (pozri obrázok 2.1b), takže pin s najvyšším číslom bude vpravo od kľúča. Funkcionalitu konkrétneho pinu možno zistiť z výrobcovej dokumentácie ku danej súčiastke.



Obrázok 2.1: DIP puzdro.

2.2 deMorganove a ďalšie pravidlá

Pojmy logickej funkcie a pravdivostnej hodnoty sú základné kamene **Booleovej algebry**. Pre potreby zápisu Booleovských (t.j. logických) funkcií bol vytvorený jednoduchý systém, v ktorom každá logická funkcia má svoj symbol.

- NOT, Negácia – vodorovná čiara nad premennou, $\bar{a}$
- AND, Logický súčin – bodka, $a.b$
- OR, Logický súčet – plus, $a + b$
- XOR, Exkluzívny súčet – plus v krúžku $a \oplus b$

¹angl. dual inline package. Niekedy ozn. aj DIL.

Pod deMorganovými pravidlami sa myslí dvojica logických rovníc, ktoré vyjadrujú vzťah medzi logickým súčtom a súčinom. Pravdivosť prvého pravidla dokážeme pomocou pravdivostnej tabuľky na obrázku 2.2. Vidno, že posledný a predposledný stĺpec, ktoré predstavujú ľavú a pravú stranu prvého deMorganovho pravidla, sa rovnajú. Overenie druhého pravidla nechávame na čitateľa.

- $\overline{a.b} = \bar{a} + \bar{b}$
- $\overline{a + b} = \bar{a}.\bar{b}$

b	a	$\bar{b}$	$\bar{a}$	$a.b$	$\overline{a.b}$	$\bar{a} + \bar{b}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

Obrázok 2.2: Overenie prvého deMorganovho pravidla.

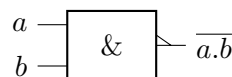
Okrem deMorganových pravidiel existujú v aj ďalšie užitočné výrazy. Ich overenie a osvojenie znova ponechávame na samostatnú prácu.

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

2.3 Logická funkcia NAND

V tejto časti uvedieme niektoré informácie o logických funkciách, ktoré budú pre toto cvičenie relevantné. Ako základný stavebný diel budeme neskôr používať logickú funkciu NAND, ktorej pravdivostnú tabuľku a schematickú značku uvádzame na obrázku 2.3.

b	a	$a.b$
0	0	1
0	1	1
1	0	1
1	1	0

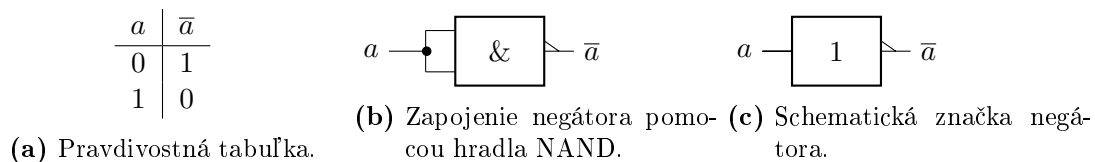


Obrázok 2.3: Pravdivostná tabuľka a schematická značka hradla NAND.

Funkcia NAND je v praxi veľmi dôležitá, pretože kombináciou hradiel NAND vieme poskladať ľubovoľnú inú logickú funkciu. V terminológii Booleovej algebry o tejto vlastnosti hovoríme, že funkcia NAND tvorí **minimálny úplný súbor** logických funkcií. Ďalej uvedieme príklady realizácie funkcií NOT, AND a OR pomocou hradiel NAND.

Funkcia NOT pomocou hradla NAND

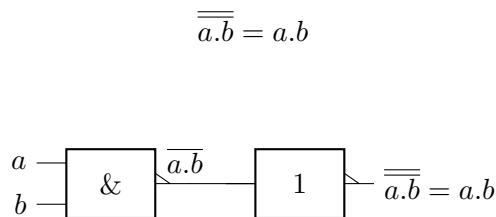
Ako prvý a najjednoduchší príklad môžeme uviesť funkciu negácie. Pozrime sa na tabuľku pravdivostných hodnôt funkcie NAND (obr. 2.3) a predpokladajme, že $a = b$. Z tabuľky nám teda zostane prvý a posledný riadok, kde vidíme, že takto zapojené hradlo NAND realizuje funkciu negácie (obr. 2.4a). Keďže vstupné hodnoty sa rovnajú, tak namiesto dvoch vstupov a, b , môžeme zaviesť len jeden vstup a , čo je znázornené na obrázku 2.4b. Schematická značka negátora bez ohľadu na to, ako je realizovaný, je na obrázku 2.4c.



Obrázok 2.4: Negátor, resp. invertor.

Funkcia AND pomocou hradiel NAND

Druhý, a tiež pomerne jednoduchý príklad, je realizovanie funkcie AND pomocou NAND hradiel. Využitím vzťahu o dvojnásobnej negácii zo sekcie 2.2 vidíme, že môžeme použiť hradlo NAND a jeho výstup potom znegovať. Na obrázku 2.5 sme negátor nakreslili schematickou značkou, ale máme na mysli NANDový negátor z predchádzajúcej podsektie.



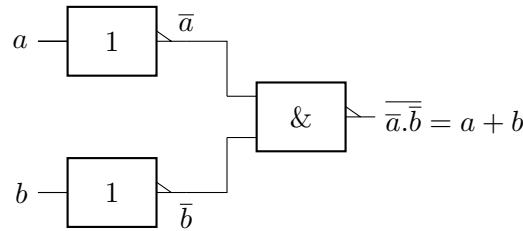
Obrázok 2.5: Realizácia funkcie AND pomocou hradiel NAND.

Funkcia OR pomocou hradla NAND

O niečo zaujímavejšia je realizácia hradla OR. Všimnime si, že chceme realizovať logický súčet, pomocou logického súčinu. Vzťahy medzi logickým súčtom a súčinom ukazujú práve deMorganove pravidlá zo sekcie 2.2. Použijeme druhé z nich, pričom znegujeme obidve strany. Vtedy dostaneme:

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

Teda vidíme, že logický súčet je rovný negovanému logickému súčinu negácií vstupných hodnôt. Negovaný logický súčin nie je nič iné, ako naša funkcia NAND. Na vstupy tohto NANDu pomocou negátorov privedieme hodnoty $\bar{a}$ a $\bar{b}$. Výsledná schéma teda bude:



Obrázok 2.6: Realizácia funkcie OR pomocou hradíel NAND.

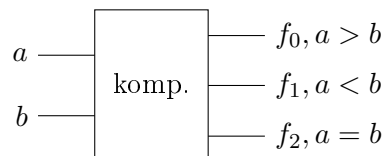
2.4 1-bitový komparátor

Základná funkcionality

Logický komparátor je sústava logických obvodov, ktoré porovnávajú dva vstupy. Výstupom z komparátora je informácia o tom, ktorý vstup je väčší (t.j. 1), prípadne že vstupy sa rovnajú. Ideme teraz takúto sústavu logických obvodov navrhnuť, pričom začneme tým, že si pomocou pravdivostnej tabuľky (obr. 2.7) zhrnieme, akú funkcionality od nášho komparátora požadujeme.

b	a	f_0	f_1	f_2	
0	0	0	0	1	$a > b \Rightarrow f_0 = 1$
0	1	1	0	0	$a < b \Rightarrow f_1 = 1$
1	0	0	1	0	$a = b \Rightarrow f_2 = 1$
1	1	0	0	1	

Obrázok 2.7: Pravdivostná tabuľka komparátora.



Obrázok 2.8: Schematické znázornenie fungovania komparátora.

Všimnime si, že celkovú funkcionality komparátora sme rozdelili na tri jednoduchšie podfunkcie. To je prirodzené, pretože ak máme dva vstupy, môžu nám pri ich porovnávaní nastať práve tri rôzne možnosti. Tie sú zapísané pomocou rovníc na obrázku 2.7 a schematicky na obrázku 2.8. Vďaka rozdeleniu problému na menšie môžeme nad každou z funkcií f_0 , f_1 a f_2 rozmýšľať ako o separátnom bloku.

Funkcia f_0

Od funkcie f_0 požadujeme, aby nám dala výstupnú pravdivostnú hodnotu 1 ak je $a > b$ (t.j. $a = 1$ a $b = 0$) a hodnotu 0 vo všetkých ostatných prípadoch. Ak by sme to chceli

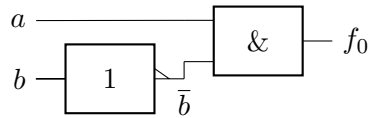
povedať jazykom výrokovej logiky, znelo by to nasledovne:

Výrok (f_0) je pravdivý, ak je pravda "a", a zároveň nie je pravda "b".

Ak by sme tento výrok chceli zapísať symbolicky, musíme si všimnúť niektoré kľúčové slová. Výraz "a zároveň" znamená logický súčin, teda funkciu AND. Druhý dôležitý výraz je "nie je pravda", ktorý označuje negáciu. Symbolický zápis teda bude:

$$f_0 = a \cdot \bar{b}$$

Zo symbolického zápisu už ľahko vidíme, aká môže byť obvodová realizácia funkcie f_0 , znázornená na obrázku 2.9.



Obrázok 2.9: Schéma funkcie f_0 .

Ku schéme na obrázku 2.9 by sme mohli dôjsť aj inou úvahou, ktorá prakticky zodpovedá analýze nakresleného obvodu. Od funkcie f_0 požadujeme, aby bol len jeden prípad, kedy máme na výstupe hodnotu 1. Ak si spomenieme na pravdivostnú tabuľku funkcie AND z predošlého cvičenia, tak vidíme, že aj AND má výstupnú hodnotu 1 práve v jednom prípade. Preto môžeme funkciu f_0 realizovať práve pomocou funkcie AND.

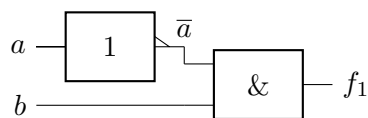
Keďže v prípade ANDu je výstup rovný 1 iba v prípade $a = b$, musíme vstupy ešte upraviť, pokiaľ chceme mať logickú jednotku v prípade, kedy $a > b$. Vtedy aj na vstupoch funkcie AND, ktorú sme použili, musia byť dve logické jednotky. Lenže, na vstupoch máme situáciu $a = 1$ a $b = 0$. Z toho vyplýva, že vstup b musíme ešte pred vstupom do ANDu znegovať, čím dostávame obvod z obrázka 2.9.

Funkcia f_1

Funkcia f_1 má byť pravdivá iba v prípade, ak $a < b$. Svojím obsahom je rovnaká ako funkcia f_0 , len otočená, a preto identickou analýzou² by sme dospeli ku zápisu:

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

To následne možno realizovať podľa schémy:



Obrázok 2.10: Schéma funkcie f_1 .

²Nechávame na premyslenie čitateľovi.

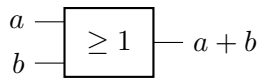
Funkcia f_2

Od funkcie f_2 požadujeme, aby bola pravdivá vtedy, ak $a = b$. To je splnené očividne v dvoch prípadoch: $a = 0, b = 0$ a $a = 1, b = 1$. Na prvý pohľad sa to môže zdať zložitejšie, ale rovnako ako v predošlých prípadoch si môžeme pomôcť výrokovou logikou:

Výrok (f_2) je pravdivý, ak je pravda "a" a zároveň je pravda "b", alebo ak nie je pravda "a" a zároveň nie je pravda "b".

Okrem už známych formuliek na logický súčin a negáciu tu máme nový výraz "alebo". Intuitívne tušíme, že výrok obsahujúci "alebo" bude pravdivý, ak bude pravdivá aspoň jedna z jeho častí. To je vlastnosť logického súčtu, teda funkcie OR, ktorej pravdivostnú tabuľku a schematickú značku uvádzame na obrázku 2.11.

b	a	$a.b$
0	0	0
0	1	1
1	0	1
1	1	1

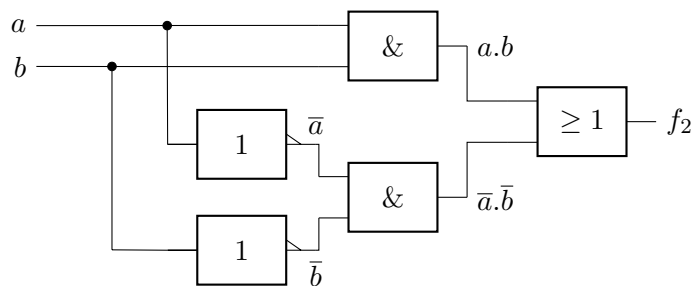


Obrázok 2.11: Pravdivostná tabuľka a schematická značka hradla OR.

Podme teraz symbolicky zapísať vyššie uvedený výrok. Vidíme, že obsahuje logický súčet, ktorý má na oboch stranách logické súčiny.

$$f_2 = a.b + \bar{a}.\bar{b}$$

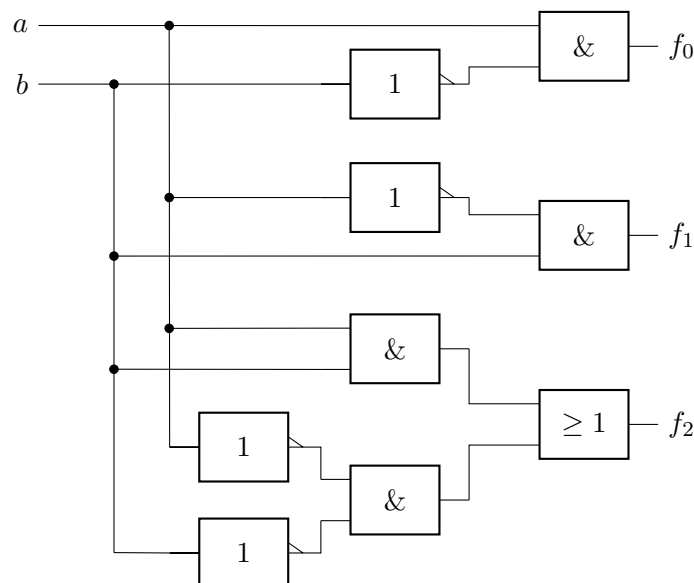
Tento výraz už vieme ľahko zakresliť schematicky:



Obrázok 2.12: Schéma funkcie f_2 .

Spojenie funkcií f_0 , f_1 a f_2 a realizácia pomocou NANDov

Aby sme dostali komparátor, spojíme teraz funkcie f_0 , f_1 a f_2 do jednej schémy na obrázku 2.13. Všetky tri funkcie totiž súčasne operujú na tých istých vstupoch a a b , tak ako sme to navrhli a znázornili na obrázkoch 2.7 a 2.8.



Obrázok 2.13: Schéma komparátora vzniknutá spojením f_0 , f_1 a f_2 .

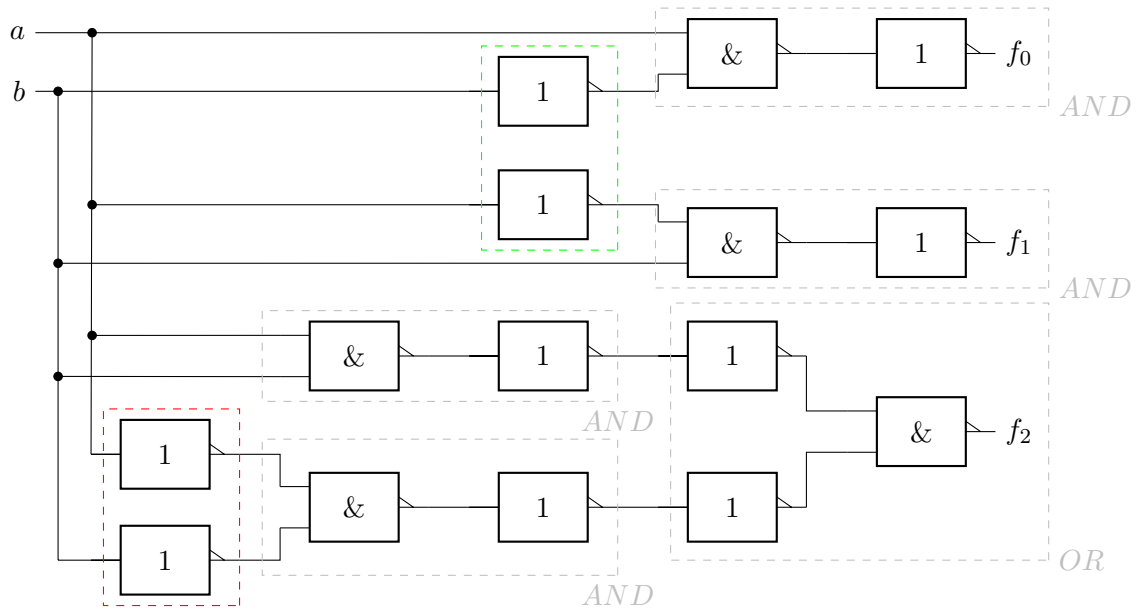
Po počiatočnom návale radosti zo splnenej úlohy nás hneď schladí zistenie, že v praxi nemáme k dispozícii obvody, ktoré by realizovali logické funkcie NOT, AND a OR. Máme k dispozícii iba funkciu NAND, ktorá našťastie má tú príjemnú vlastnosť, že pomocou nej je možné realizovať všetky ostatné logické funkcie. V sekcii 2.3 sme uviedli a rozanalyzovali realizácie funkcií NOT, AND aj OR, pomocou hradieľ NAND. Tým pádom môžeme všetky elementy v schéme 2.13 nahradiť tak, aby boli všade použité iba hradlá NAND. Takto dostaneme schému na obrázku 2.14. Pod schematickými značkami negátorov máme opäť na mysli ich realizáciu pomocou hradla NAND.

Ako vidíme, schéma sa nám trochu rozrástla, pričom na jej zostavenie v tomto prevedení by bolo potrebných až 15 hradieľ NAND. Ako ukážeme, toto je krajne neefektívne, nehovoriac o tom, že v praxi máte k dispozícii iba dva integrované čipy po 4 hradlá NAND.

Prvá redukcia počtu hradieľ

Na zapojení z obrázku 2.14 možno vykonať hneď dve optimalizácie. Prvou očividnou optimalizáciou je zrušenie sériovo zapojených dvojíc negátorov vo funkcii f_2 , pretože platí výraz o dvojitej negácii zo sekcii 2.2.

Druhé zlepšenie sa týka negátorov, vyznačených na obrázku 2.14 červeným obdĺžnikom. Sú prítomné preto, aby na vstupoch nasledovného NANDu boli hodnoty $\bar{a}$ a $\bar{b}$. Všimnime si ale negátory v zelenom obdĺžniku, ktoré sú súčasťou funkcií f_0 a f_1 . Tieto rovnako vyrábajú negované hodnoty vstupov a a b . Preto negátory v červenom obdĺžniku vynecháme a namiesto nich si hodnoty $\bar{a}$ a $\bar{b}$ vezmeme z výstupov negátorov v zelenom obdĺžniku. Keď obidve tieto optimalizácie zakreslíme, dostaneme obrázok 2.15.



Obrázok 2.14: Schéma komparátora realizovaná hradlami NAND.

Druhá redukcia počtu hradíel – Výsledná schéma

V predošlom kroku sme sa zbavili šiestich hradíel, ale stále ich máme deväť, teda o jedno viac, než máme k dispozícii. Treba teda ešte jednu optimalizáciu. Z toho, ako sme si zadefinovali funkcie f_0 , f_1 a f_2 vidíme (pozri obrázok 2.7), že platí:

$$f_0 + f_1 + f_2 = 1$$

Využijeme asociatívnosť logického súčtu:

$$(f_0 + f_1) + f_2 = 1$$

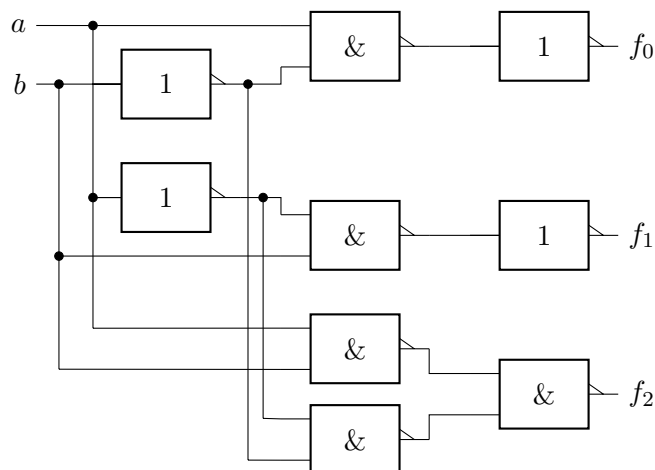
Zo sekcie 2.2 použijeme vzťah $a + \bar{a} = 1$. Keď ho porovnáme z predchádzajúcim výrazom, dostaneme:

$$a = f_0 + f_1 \quad \bar{a} = f_2$$

Dosadením za a tým pádom dostaneme:

$$f_2 = \overline{f_0 + f_1}$$

To je užitočná informácia! Funkciu f_2 teda môžeme realizovať pomocou výstupov funkcií f_0 a f_1 vykonaním logického súčtu, ktorý potom znegujeme. Ešte predtým, ako zakreslíme toto zistenie do novej schémy sa v krátkosti zamyslime. Na funkciu OR by sme potrebovali tri NAND hradlá a na negáciu ešte jedno. Tým pádom by sme mali vo výsledku desať hradíel NAND, čo je o jedno viac ako pred tým. Je teda na mieste otázka: Kde sa stala chyba?



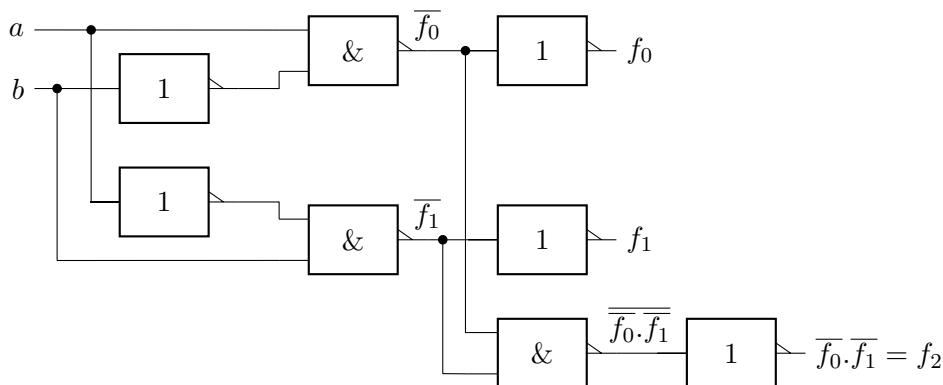
Obrázok 2.15: Schéma komparátora po prvej redukcii počtu hradíel.

Odpovieme na ňu tým, že budeme pokračovať v optimalizácii. Aplikujme na výraz pre f_2 deMorganove pravidlo:

$$f_2 = \overline{f_0 + f_1} = \overline{f_0} \cdot \overline{f_1}$$

To znamená, že namiesto negovania logického súčtu potrebujeme urobiť logický súčin negácií. To je pre nás aj pohodlnejšie kvôli tomu, že máme k dispozícii len hradlá NAND, ktoré predstavujú práve súčin. Zdá sa teda, že plán je jasný: Vezmeme výstupy funkcií f_0 a f_1 , znegujeme ich a vykonáme funkciu AND pomocou NANDu a negátora.

Buďme ale zdržanliví a pozrime sa najprv na schému 2.15. Ak na výstupe pravého horného negátora má signál hodnotu f_0 , tak na jeho vstupe musí byť hodnota $\overline{f_0}$. Rovnako to platí aj pre funkciu f_1 . Tým pádom namiesto opätovného negovania funkcií f_0 a f_1 môžeme použiť už existujúce hodnoty ich negácií, ktoré sa v obvode nachádzajú. Stačí ich len vodičmi priviesť tam, kam potrebujeme, teda na vstup NANDu spolutvoriaceho funkciu f_2 . Výsledná schéma len z ôsmich hradíel je na obrázku 2.16.



Obrázok 2.16: Finálna schéma komparátora po druhej optimalizácii.

2.5 Úloha na cvičenie

Zostavte 1-bitový komparátor podľa schémy 2.16. Budete potrebovať poznať aj číslovanie pinov na vašich integrovaných obvodoch a aj to, aká funkcionality je jednotlivým pinom priradená. Tieto informácie máte k dispozícii na každom pracovnom stole. Na schéme to kvôli prehľadnosti nakreslené nie je, ale na výstup každej funkcie si kvôli praktickosti ako indikátor zapojte LED diódu. Nezabudnite na ochranný rezistor! Odporúčaná postup riešenia úlohy:

1. Zostavte najprv funkciu f_0 a otestujte ju privedením príslušných hodnôt signálov a a b (t.j. $a > b$).
2. Zostavte funkciu f_1 a otestujte ju privedením príslušných hodnôt signálov a a b (t.j. $a < b$).
3. Nakoniec zostavte funkciu f_0 a otestujte ju privedením príslušných hodnôt signálov a a b (t.j. $a = b$, oba prípady).