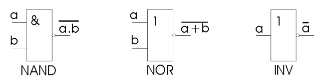
5 Realizace logických funkcí

Nyní se konečně seznámíme s tím, jak navrhnout zařízení, které se bude chovat podle zadané logické funkce. Ukážeme si i některé praktické příklady. Základní prvky, které budeme používat budou: hradlo NAND, hradlo NOR a invertor (viz obr. 5.1).



Obr. 5.1 Schematická značka hradel NAND, NOR, INV

Víme, že funkce NAND (resp. NOR) tvoří úplný soubor logických funkcí, což pro nás znamená, že vystačíme s jedním typem hradla pro realizaci všech funkcí. Nejprve se budeme zabývat metodami realizace logických funkcí pomocí jednoho typu hradla (NAND nebo NOR) bez ohledu na to, jaké skutečné součástky (integrované obvody - dále jen IO) máme k dispozici.

Poznámka. Ve skutečnosti je důležité, kolik použijeme hradel např. dvouvstupových, třívstupových atp., protože na tom záleží, kolik součástek (IO) musíme použít. Např. v IO 7400 jsou 4 dvouvstupová hradla NAND a v IO 7410 3 třívstupová hradla NAND. Jestliže navrhneme zapojení s pěti dvouvstupovými a dvěma třívstupovými hradly NAND, bude výhodnější použít k realizaci jen dva IO - dvouvstupové hradlo lze nahradit třívstupovým s jedním nepřipojeným vstupem (v technologii TTL se nepřipojený vstup chová jako by byl zapojen na log.1) nebo se dvěma vstupy spojenými.

5.1 Realizace pomocí hradel NAND

Mějme funkci zadanou pomocí MNDF, např. $f = a + \bar{b}c + c\bar{d}$. Naším cílem je získat místo všech operací negované logické součiny. Na základě minulých kapitol si dokážeme poradit pomocí dvojí negace funkce f:

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

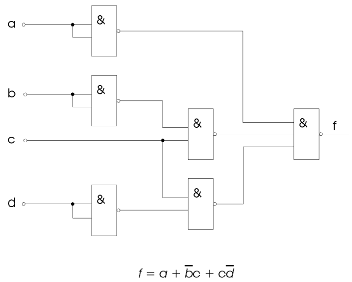
Získali jsme to, co jsme potřebovali, výsledné schema je na obr. 5.2. Když si výsledný tvar dobře prohlédneme zjistíme, že si můžeme ušetřit práci s negováním, způsob realizace snadno určíme přímo z tvaru zadané funkce. Můžeme vyslovit následující pravidlo:

Pro realizaci logických operací + i . používáme hradlo NAND, proměnné vstupující přímo do hradla na liché úrovni od konce musí být negované.

5.2 Realizace pomocí hradel NOR

Zde bude situace velmi podobná; jestliže bude funkce zadaná v MNKF vlastně úplně stejná (až na použitá hradla). Mějme funkci zadanou v MNKF: $f' = b(\bar{a} + c)(\bar{c} + d)$, tedy

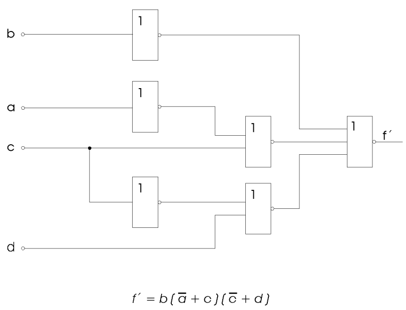
$$\overline{\overline{f'}} = \overline{\overline{b(\bar{a} + c)(\bar{c} + d)}} = \overline{\bar{b} + \overline{(\bar{a} + c)} + \overline{(\bar{c} + d)}}$$



Obr. 5.2 Realizace funkce f pomocí hradel NAND

Výsledné schema je na obr. 5.3. Můžeme též vyslovit podobné pravidlo jako pro hradla NAND:

Pro realizaci logických operací + i . používáme hradlo NOR, proměnné vstupující přímo do hradla na liché úrovni od konce musí být negované.



Obr. 5.3 Realizace funkce f' pomocí hradel NOR

Jestliže je funkce zadaná v MNDF a použijeme hradla NAND (resp. MNKF a NOR), je náš úkol celkem snadný. O něco horší situace nastane v ostatních případech. Vždy je možné použít vícenásobné negace (de Morganovy zákony), ale při algebraických úpravách se snadno udělá chyba. Existuje pomůcka, která je použitelná stejným způsobem pro hradla NAND i NOR, je výhodné ji použít i při opačném úkolu (analýze schematu), její použití je velmi názorné při hledání statických a dynamických hazardů. Jedná se o tzv. Rottovy mřížky.

5.3 Cvičení

1. Navrhněte převodník kódů z kódu 2 z 5 (čti dva z pěti) do kódu BCD. Pro realizaci použijte a)hradla NAND, b)hradla NOR, c)vhodný typ multiplexoru nebo dekodéru, d)paměť PROM 74188.

Řešení

Kód 2 z 5 je bezpečnostní kód, kde číslice 0 až 9 jsou zakódovány pomocí pěti bitů, z nichž vždy dva jsou jedničkové a tři nulové (pro 5 bitů je to právě 10 možností). Bezpečnostní kód znamená, že případou chybu je možné rozpoznat (říkáme detekovat) a někdy dokonce i opravit. Jedna chyba znamená buď tři nebo jednu jedničku, což není správné číslo (říkáme kódové slovo). Dále uveďme dvě varianty kódu 2 z 5 (Tab. 5.1). Číslice v symbolech 74210 a 84210 mají význam vah jednotlivých proměnných s tím, že musíme dodržet počet nul a jedniček.

Desítkové číslo	kód BCD $DCBA$	kód 2z5 typ 74210 $e\,d\,c\,b\,a$	kód 2z5 typ 84210 $e\,d\,c\,b\,a$
0	0 0 0 0	1 1 0 0 0	1 0 1 0 0
1	0 0 0 1	0 0 0 1 1	0 0 0 1 1
2	0 0 1 0	0 0 1 0 1	0 0 1 0 1
3	0 0 1 1	0 0 1 1 0	0 0 1 1 0
4	0 1 0 0	0 1 0 0 1	0 1 0 0 1
5	0 1 0 1	0 1 0 1 0	0 1 0 1 0
6	0 1 1 0	0 1 1 0 0	0 1 1 0 0
7	0 1 1 1	1 0 0 0 1	1 1 0 0 0
8	1 0 0 0	1 0 0 1 0	1 0 0 0 1
9	1 0 0 1	1 0 1 0 0	1 0 0 1 0

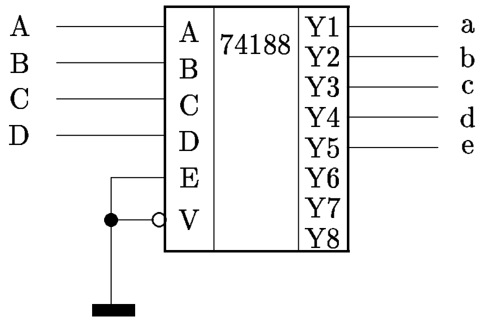
Tab. 5.1 Dvě varianty kódu 2 z 5.

Jedná se o návrh kombinačního obvodu se čtyřmi vstupy (A, B, C, D) a pěti výstupy (a, b, c, d, e), tedy o minimalizaci skupiny logických funkcí. Při realizaci pomocí hradel použijeme 5 Karnaughových map pro minimalizaci funkcí a, b, c, d, e. Podle použitého typu hradel najdeme buď MNDF nebo MNKF s tím, že se budeme snažit nalézt skupinové termy. Jestliže použijeme paměť PROM 74188, je situace ještě jednodušší - není možná minimalizace (paměť obsahuje adresní dekodér, přístup do paměti je pouze prostřednictvím úplné adresy). Paměť 74188 má paměťovou matici 32 x 8 bitů, tedy 5 adresních vstupů a 8 výstupů a programujeme v podstatě zadanou tabulku kódu, jeden vstup zůstává nevyužitý (obr. 5.12).

2. Navrhněte převodník kódů z kódu 2 z 5 do BCD. Realizujte a) pomocí hradel NAND, b) pomocí hradel NOR, c) pomocí multiplexorů nebo dekodérů, d) pomocí paměti PROM s tím, že umožníte vstup obou variant kódu 2 z 5, pro vstupní proměnnou $X = 0$ vstupuje varianta 74210 pro $X = 1$ varianta 84210.

Řešení

Máme navrhnout opačný převodník než v příkladu 1. Jedná se tedy o návrh kombinačního obvodu s pěti vstupy (a, b, c, d, e) a čtyřmi výstupy (A, B, C, D) tedy pro zadání a) až c)



Obr. 5.12 Realizace převodníku pomocí paměti PROM 74188

o minimalizaci skupiny logických funkcí. Budeme vytvářet nyní jen čtyři mapy, ale tentokrát pro 5 proměnných. Jedno z možných řešení v MNDF je :

$$A = \bar{e}b + ea + ec$$
$$B = \bar{e}c + ea$$
$$C = \bar{e}d + ea$$
$$D = eb + \qquad ec$$

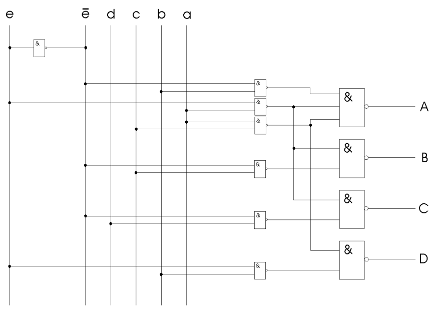
Realizace pomocí hradel NAND (s tím, že skupinové termy - zde imlikanty - jsou ea a ec) je na obr. 5.13 a vychází z předchozích znalostí - viz odst. 5.1 nebo pomocí Rottových mřížek - viz odst. 5.3.

Při použití multiplexorů vycházíme z popisu funkce příslušného typu multiplexoru a záleží na tom, zda chceme použít pouze multiplexory nebo jestě další hradla, jaký typ multiplexoru se nám hodí (jde např. o velikost pouzdra). Pro IO 74153 platí (vstupy jsou označené $0, 1, 2, 3$, výstupy $1Y$ a $2Y$ a adresové vstupy A a B, viz obr. 5.14):

$$2Y = 1Y = \overline{AB}0 + A\overline{B}1 + \overline{A}B2 + AB3$$

Jedno z možných řešení pomocí dvou multiplexorů 74153 a vycházející z výše uvedených rovnic je na obr. 5.14. Zde je uvedena varianta, kdy použijeme pro oba mutiplexory stejnou adresu e. Pak ovšem nezbývá než použít ještě další hradla pro realizaci funkcí A a D. Je možné nalézt i takovou realizaci, kde použijeme pouze dva multiplexory 74153 a žádná další hradla (zavedeme jiné proměnné do adresních vstupů multiplexorů); viz obr. 5.15. Dokažte, že realizace uvedené na obr. 5.14 a 5.15 odpovídají výše uvedenému řešení.

Pro řešení d) použijeme dva IO 74188, vstup X vedeme do výběrových vstupů IO (obr. 5.16). První paměť bude obsahovat na adresách podle varianty kódu 2 z 5 74210 kód BCD, druhá paměť bude mít kódem BCD obsazena paměťová místa podle varianty 84210. (Schema na obr. 5.16 neodpovídá skutečnosti; je třeba mít na paměti, že paměťový obvod 74188 má výstupy s otevřeným kolektorem, takže pro správně fungující zapojení je třeba zapojit výstupy přes odpory na napájení.) Dále je možné využít volné výstupy pro další vhodné funkce takovéhoto převodníku, např. pro výstup určující zda přijaté slovo je kódové či nekódové.



Obr. 5.13 Realizace převodníku kódu 2z5 do BCD pomocí hradel NAND

3. Navrhněte jeden blok binární sčítačky.
a) pomocí hradel NAND
b) pomocí bloků tzv. půlsčítačky.

Řešení

a) Blok binární sčítačky je kombinační obvod, který má tři vstupy (dva sčítance a, b a přenos z nižšího řádu p) a dva výstupy (součet s a přenos do vyššího řádu q). Binární sčítání je vyjádřeno pravdivostní tabulkou v Tab. 5.2.

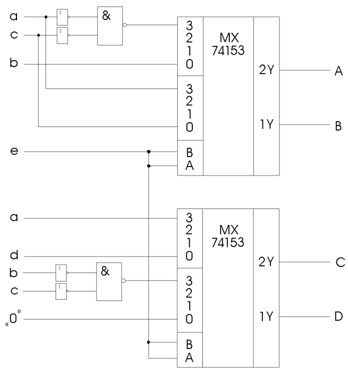
$a\ b\ p$	$s\ q$
0 0 0	0 0
0 0 1	1 0
0 1 0	1 0
0 1 1	0 1
1 0 0	1 0
1 0 1	0 1
1 1 0	0 1
1 1 1	1 1

Tab. 5.2 Tabulka funkce jednoho bloku binární sčítačky

Pomocí Karnaughových map získáme následující výrazy pro součet s a přenos q:

$$s = \bar{a}\bar{b}p + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp$$

$$q = ap + bp + ab$$



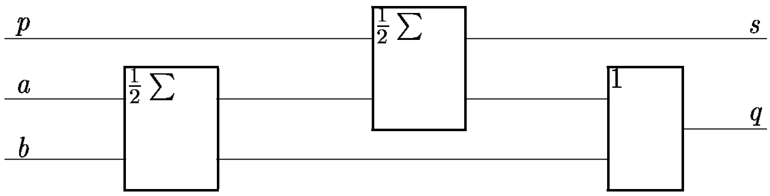
Obr. 5.14 Realizace převodníku kódu 2z5 do BCD pomocí IO 74153 a hradel

Bohužel binární sčítačka jako základní stavební prvek aritmeticko-logické jednotky počítače není právě nejjednodušší obvod. Součet nelze minimalizovat, žádné dva stavy nejsou sousední. Při realizaci musíme počítat se zpožděním způsobeným průchodem signálu přes tři hradla (včetně negací). Matematickými úpravami je možné vyjádřit součet pomocí funkce XOR a přenos pomocí majority:

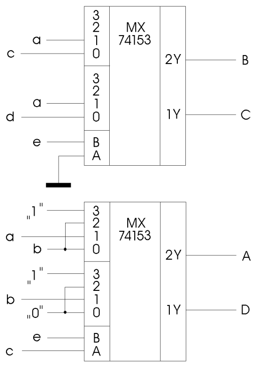
$$s = p(\bar{a}\bar{b} + ab) + \bar{p}(\bar{a}b + a\bar{b}) = p\overline{(a \oplus b)} + \bar{p}(a \oplus b) = p \oplus a \oplus b$$

$$q = ap + bp + ab = M_3(a, b, p)$$

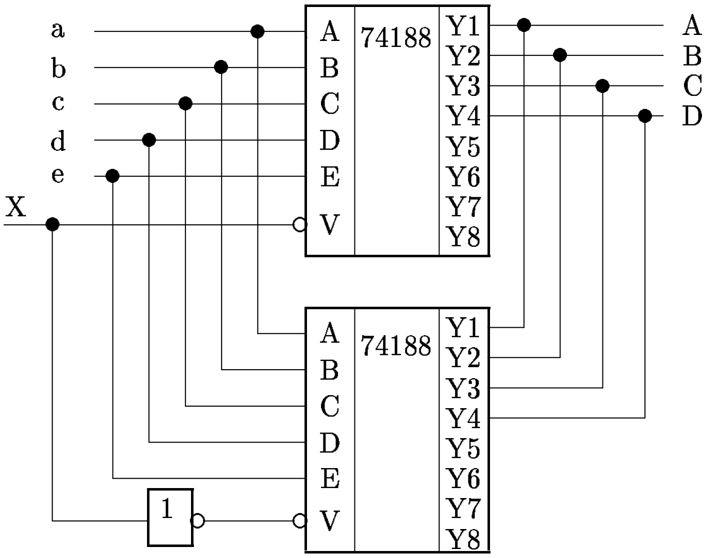
b) Půlsčítačka je kombinační obvod se dvěma vstupy a, b a dvěma výstupy s, r, kde $s = a \oplus b$ a $r = ab$. Je možné sestavit úplnou sčítačku podle obr. 5.17. Dokažte to.



Obr. 5.17 Realizace jednoho bloku úplné binární sčítačky pomocí půlsčítaček



Obr. 5.15 Realizace převodníku kódu 2z5 do BCD pomocí IO 74153



Obr. 5.16 Realizace převodníku dvou variant kódu 2z5 do BCD pomocí IO 74188