

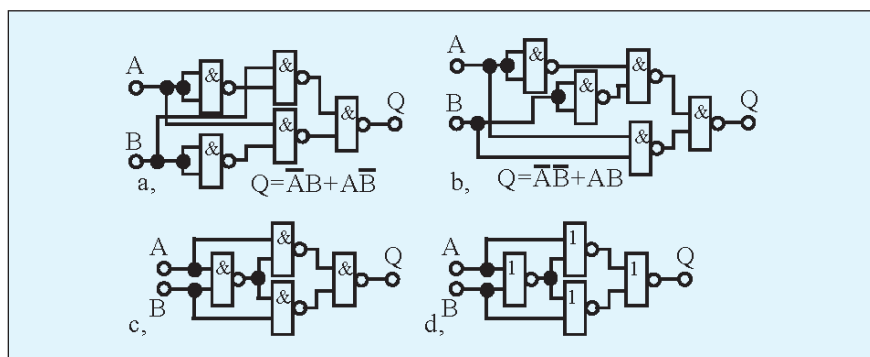
Ekvivalencia, antivalencia NAND, NOR megvalósítása

Dr. Madarász László okl. villamosmérnök, madarasz@3lan.hu

A digitális áramkörökkel foglalkozók számára jó ismerős az antivalencia és az ekvivalencia függvény. Az **1. ábrán** szerepel e két logika függvény műveleti jele és az ún. minterm-alakja, valamint a rajzjele. Az antivalencia (EX-OR, exclusive OR, exkluzív VAGY, kizáró VAGY) akkor ad kimeneti **1** értéket, ha a két bemenete eltérő logikai értékű. Műveleti jele egy körbe zárt összeadásjel, rajzjelében a logikai funkciójel (=1) arra utal, hogy ha a két bemenet közül egy (és csakis egy) logikai **1** értékű, akkor lesz a kimenet is logikai **1**.

Az ekvivalencia függvény az antivalencia negáltja, akkor ad a kimenetén **1**-t, ha a két bemeneti jel logikai értéke azonos. Nincs önálló rajzjele, az antivalencia rajzjelet egy negációra utaló körrel kiegészítve lehet ekvivalencia kaput rajzolni.

A 74-es sorozatokban találunk antivalencia és ekvivalencia kapukat is. A 7486, 74LS86 tokok négy darab kétbemenetű antivalencia kaput tartalmaznak, a 74266, 74LS266 áramkörökben négy darab ekvivalencia kaput alakítottak ki. Ezeket a tokokat nem mindig tartjuk a fiókban, elég ritkán van rájuk szükség. Így előfordul, hogy egy kapcsolás ellenőrzéséhez vagy egy ötlet kipróbálásához szükség lenne



2. ábra

ezekre a kapukra, és éppen nem állnak rendelkezésre. Ilyenkor más kapukból állíthatjuk elő ezeket az elemeket.

A két függvény minterm alakja alapján NAND kapukból megépíthetjük ezeket a különleges kapukat, a **2.a ábrán** egy ilyen antivalencia kapcsolás látható, a **2.b ábrán** egy ekvivalencia. Mindkettő tökéletesen működik. Mivel öt NAND kaput tartalmaznak, a 7400 vagy 74LS00 IC-ből, melyben négy darab kétbemenetű NAND helyezkedik el, két darabot kell felhasználni a megépítésükhöz.

Van azonban olyan kapcsolási megoldás, amelyik csak egyetlen IC-t használ fel. A **2.c ábrán** szereplő kapcsolás is megvalósítja az antivalencia kaput, meghozzá négy darab NAND kapu felhasználásával. Ez a kapcsolás a min-

term alakra épülő szokásos tervezési eljárással nem tervezhető meg, tulajdonképpen a digitális áramkörökkel foglalkozók között terjed, szinte „szájról szájra”. Ha az egyes kapuk kimenetére felírjuk az ott kialakuló logikai függvényt, a Boole-algebra szabályait használva igazolható az antivalencia-működés. De más módon is igazolható a helyes működés: csak össze kell állítani ezt a kis kapcsolást és máris ellenőrizhető.

A **2.d ábra** pedig egy még kevésbé ismert megoldást mutat be. Ha a c kapcsolást nem NAND elemekből, hanem NOR kapukból alakítjuk ki (azaz 7402 vagy 74LS02 IC-t használunk fel), akkor ekvivalencia függvényt fog megvalósítani az áramkör! Ezzel a fogással tehát egyetlen tokkal ez a kapu is megépíthető.

Egy fontos megjegyzés kívánkozik még ide. A kétbemenetű kapukat a Texas Instruments és a másodgyártók a 74-es sorozatokban a **3.a ábra** szerint tokozák, így rendeződnek el az antivalencia kapuk is a 7486, 74LS86 tokokban. A NOR kapukat azonban nem így kötötte be a gyártó! A 7402, 74LS02 lábkiosztása a **3.b ábra** szerint alakul, amire a kísérletek során figyelniünk kell! Az elmondottak az

Antivalencia		Ekvivalencia																															
$Q = \bar{A}B + A\bar{B} = A \oplus B$		$Q = \bar{A}\bar{B} + AB = A \equiv B$																															
Igazságtáblázat:	Rajzjel:	Igazságtáblázat:	Rajzjel:																														
<table border="1"> <tr><th>A</th><th>B</th><th>Q</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	0		<table border="1"> <tr><th>A</th><th>B</th><th>Q</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	1	
A	B	Q																															
0	0	0																															
0	1	1																															
1	0	1																															
1	1	0																															
A	B	Q																															
0	0	1																															
0	1	0																															
1	0	0																															
1	1	1																															

1. ábra