Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Irányítástechnika és Informatika Tanszék

Dr. Pilászy György

Digitális technika 1
07 előadás

(Funkcionális építőelemek – Multiplexerek, komparátorok)

Lektorálta: Dr. Horváth Tamás

Funkcionális építőelemek

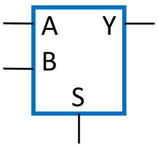
Az alábbi fejezetben röviden ismertetünk néhány olyan építőelemet, amely kapcsolódik a kombinációs hálózatok témaköréhez. Ezeknek az építőelemeknek a használata nagy mértékben csökkenti a felhasznált építő elemek számát a „hagyományos” kapuáramkörökkel végzett megvalósításhoz képest. Bár ezeknek az alkatrészeknek az alkalmazása során elsősorban intuitív tervezési módszerek a meghatározóak, ahol lehet, igyekszünk néhány szisztematikusan is használható módszert bemutatni.

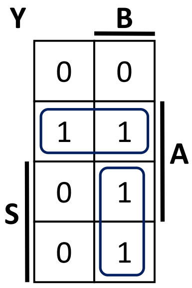
Multiplexerek

A multiplexer áramkörök vezérelt kapcsolóként működnek. A kiválasztó bemeneteikre kapcsolt bináris kombinációval képesek kiválasztani egy jelbemenetet és azt megjeleníteni a kimenetükön. Ez a megfogalmazás első olvasásra bonyolultnak tűnik, ezért a következőkben egy logikai tervezési feladatként készítjük el.

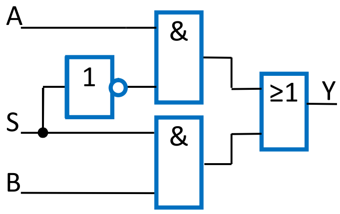
2/1 multiplexer

Egy logikai hálózatnak három bemenete (A,B,S) és egy kimenete (Y) van. S=”0” érték esetén a hálózat kimenete megegyezik az A bemenet értékével Y=A. S=”1” érték esetén a hálózat kimenete megegyezik a B bemenet értékével Y=B. Tervezzük meg ezt a három bemenetű egy kimenetű logikai hálózatot.





$$Y = \overline{S} \cdot A + S \cdot B$$

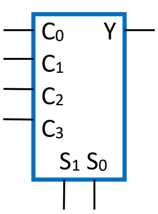


Érdemes megfigyelni a 2/1 multiplexer elvi logikai rajzát. Jól látszik, hogy a két ÉS kapu közül az S bemenettől függően vagy az A, vagy a B bemenet értéke szerint változik az Y.

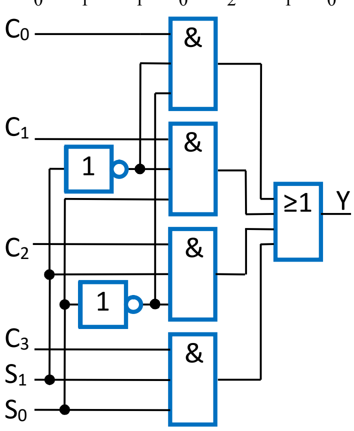
4/1 multiplexer

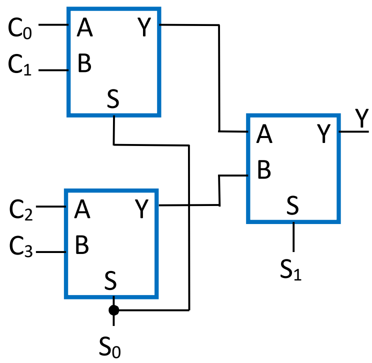
Az előző példa alapján akár intuitívan is elkészíthetjük a 4/1 multiplexer elvi logikai rajzát. Ez a multiplexer a C0…C3 bemenetek közül választ egyet az S1, S0 kiválasztó bemenetekre adott bináris kód szerint. Ha rendelkezésünkre áll 2/1 multiplexer, akkor abból is kialakíthatunk 4/1-eset az alábbi elrendezésben.

$$Y = \overline{S_1} \cdot \overline{S_0} \cdot C_0 + \overline{S_1} \cdot S_0 \cdot C_1 + S_1 \cdot \overline{S_0} \cdot C_2 + S_1 \cdot S_0 \cdot C_3$$



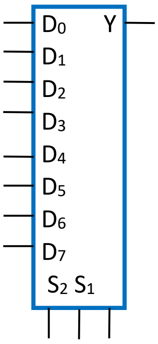
S_1	S_0	Y
0	0	C_0
0	1	C_1
1	0	C_2
1	1	C_3



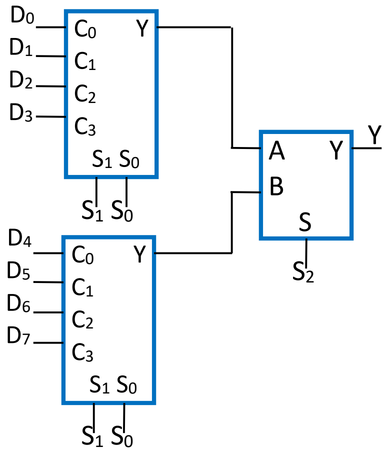


8/1 multiplexer

Az előző két alkatrészből előállíthatunk 8/1-es multiplexert, de értelem szerint a kétszintű kombinációs megvalósítás is megoldható. A 8/1-es multiplexer a D0…D7 bemeneteiből jelenít meg egyet az Y kimenetén az S2..S0 kiválasztó bemenetek értékétől függően.



S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7



A következőkben a multiplexerek gyakorlati felhasználásához mutatunk példát. A működésükből adódóan hazárdmentes kombinációs hálózat kialakítására nem alkalmasak, azonban egyetlen építőelemmel és minimális kiegészítő hálózattal tetszőleges logikai függvény megvalósítására alkalmasak. Egyetlen 8/1 multiplexer kiegészítő alkatrészek nélkül alkalmas tetszőleges három változós logikai függvény megvalósítására. Gondoljunk csak utána, a három változót bekötjük az S_0,S_1,S_2 kiválasztó bemenetekre, majd az igazságtáblázatnak megfelelő bináris értékeket rákötjük a D_0…D_7 bemenetekre. Négy változós függvény is kialakítható, de ilyenkor elképzelhető, hogy szükség lesz egy inverterre is, amint a következő példában látni fogjuk.

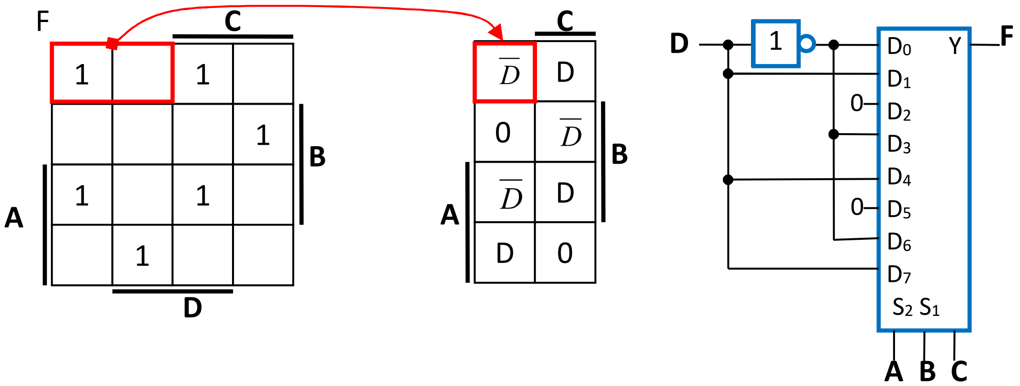
Minterm indexeivel adott az alábbi logikai függvény:

$$F(A,B,C,D)=\sum^{4}(0,3,6,9,12,15)$$

Egyetlen 8/1 multiplexer és minimális kiegészítő hálózat felhasználásával állítsuk elő az F függvényt.

Első lépésként ábrázoljuk F Karnaugh táblázatát. Láthatjuk, hogy nehezen minimalizálható függvénnyel van dolgunk. A multiplexeres megvalósítás alapötlete legyen a következő. Kössük be az A,B,C változókat a multiplexer kiválasztó bemeneteire. Készítsünk egy segéd táblázatot, amely nagyon hasonló a Karnaugh táblázathoz, de minden cellája a multiplexer 1-1 bemenetéhez tartozik. Ez a segédtáblázat fog nekünk segíteni abban, hogy meghatározzuk, a multiplexer bemenetére „0”, „1”konstans értéket vagy a negyedik változót illetve annak negáltját kell-e kapcsolni.

Az alábbiakban felrajzolt elrendezésben az A,B,C=000 bemeneti kombináció esetén az F Karnaugh táblájának bal felső sarkában található két cella tartozik (m^4_0 és m^4_1). Ezeket szögletes téglalappal bejelöltük. Figyeljük meg, hogy D=0 esetén F=1, míg D=1 esetén F=0 függvényérték van előírva ezeken a helyeken, ezért a multiplexer D_0 bemenetére $\overline{D}$ értéket kell kapcsolni. Hasonlóan az alatta lévő két cellában 0 érték van előírva, ezért a multiplexer D_2 bemenetére fix „0” értéket kell kapcsolni.

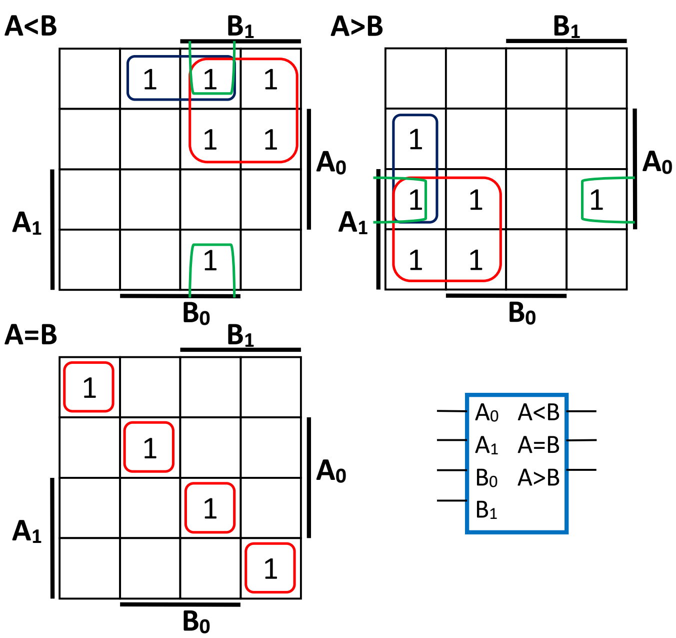


Komparátorok

Amennyiben binárisan ábrázolt számokat szeretnénk összehasonlítani, akkor a két szám közötti reláció megjelenítésére komparátor áramkört használhatunk. Két szám azonos helyi értékének összehasonlításakor az a nagyobb, amelyik „1” értékű. Ha a vizsgált helyi értéken mindkét számjegy azonos, akkor az alacsonyabb helyi érték alapján döntünk. Egyenlőnek csak akkor nyilváníthatjuk a két számot, ha egyik helyi értéken sem találtunk eltérést.

A következőkben tervezzünk meg kétbites bináris számok összehasonlítására szolgáló komparátort. A tervezendő hálózatnak négy bemenete (A_1, A_0, B_1, B_0) és három kimenete (A<B, A=B, A>B) van. A_0, B_0 jelöli a legkisebb helyi értékeket. Adjuk meg a komparátor igazságtáblázatát, majd grafikus minimalizálással állítsuk elő a három kimenet logikai függvényeit.

A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0		1	
0	0	0	1	1		
0	0	1	0	1		
0	0	1	1	1		
0	1	0	0			1
0	1	0	1		1	
0	1	1	0	1		
0	1	1	1	1		
1	0	0	0			1
1	0	0	1			1
1	0	1	0		1	
1	0	1	1	1		
1	1	0	0			1
1	1	0	1			1
1	1	1	0			1
1	1	1	1		1	

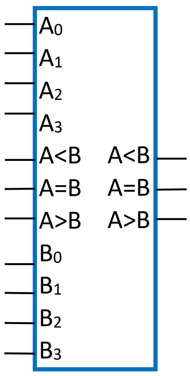


$$A < B \rightarrow B_1 \cdot \overline{A_1} + B_0 \cdot \overline{A_1} \cdot \overline{A_0} + B_1 \cdot B_0 \cdot \overline{A_0}$$

$$A > B \rightarrow A_1 \cdot \overline{B_1} + A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot A_0 \cdot \overline{B_0}$$

$$A = B \rightarrow \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0}$$

A következőkben röviden bemutatunk egy konkrét katalógus áramkört, egy négy bites kaszkádosítható komparátort [5]. A komparátort, mint funkcionális építő elemet az alábbi szimbólummal ábrázoljuk:

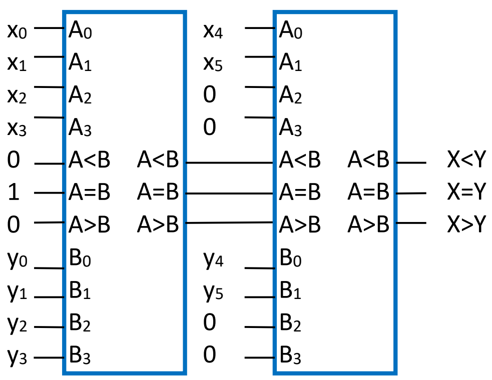


A komparátor négy bites bináris számokat hasonlít össze, melyeket az $A_0..A_3$ és a $B_0…B_3$ bemenetekre kell kapcsolni. Az A_0, B_0 jelöli a legkisebb helyi értékeket. Amennyiben az A és B bemenetre kapcsolt számok különböznek, úgy az áramkör azonnal mutatja a relációnak megfelelő eredményt. Ilyenkor a kaszkádosító bemenetek semmilyen módon nem szólnak bele az összehasonlítás végeredményébe. Amennyiben az A és B bemenetre kapcsolt négy bites bináris számok egyformák, úgy a kaszkádosító bemenetek határozzák meg a kimenetek értékeit. Az A=B kimenet kizárólag akkor ad „1” értéket, ha az A és B szám megegyezik és az A=B kaszkádosító bemeneten is „1” érték van. A részletes működést bemutató táblázat az alábbi [5].

Összehasonlító bemenetek				Kaszkádosító bemenetek			Kimenetek		
A_3,B_3	**A_2,B_2**	**A_1,B_1**	**A_0,B_0**	**A>B**	**A<B**	**A=B**	**A>B**	**A<B**	**A=B**
$A_3>B_3$	-	-	-	-	-	-	1	0	0
$A_3<B_3$	-	-	-	-	-	-	0	1	0
$A_3=B_3$	$A_2>B_2$	-	-	-	-	-	1	0	0
$A_3=B_3$	$A_2<B_2$	-	-	-	-	-	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	-	-	-	-	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	-	-	-	-	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	-	-	-	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	-	-	-	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	**1**	0	0	**1**	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	**1**	0	0	**1**	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	-	-	**1**	0	0	**1**
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	1	0	0	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	0	1	1	0

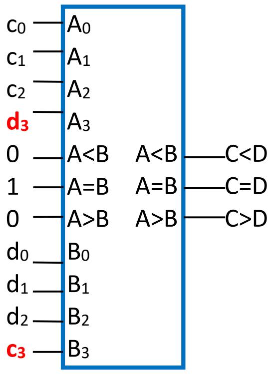
Négybites kaszkádosítható komparátor működési táblázata

A komparátor gyakorlati használatának bemutatására két példát ismertetünk. Először alakítsunk ki a fenti építőelemekből olyan komparátor egységet, amely alkalmas $X(x_0…x_5)$ és $Y(y_0…y_5)$ hatbites előjel nélküli egész számok összehasonlítására. x_0 és y_0 jelöli a legkisebb helyi értékeket. A kisebb helyi értékeket összehasonlító építőelem kaszkádosító bemeneteire az 010 kombinációt kötjük, hogy egyenlőség esetén ezt jelezze. Ugyanennek az építőelemnek a kimeneteit rákötjük a magasabb helyi értékeket összehasonlító építőelemre. Mivel csak hatbites számokat hasonlítunk össze, az „üresen” maradó bemenetekre azonos konstans (0) értéket kapcsolunk. Természetesen más, de egyforma érték is jó lehet a nem használt bemeneteken.



Hatbites bináris számok összehasonlítása

A második feladatunk egy olyan komparátor kialakítása, amely $C(c_0…c_3)$ és $D(d_0…d_3)$ négybites kettes komplemensben ábrázolt számokat hasonlít össze. c_0 és d_0 jelöli a legkisebb helyi értékeket. A komplemens ábrázolás miatt azonos előjelű számok összehasonlításakor helyes eredményt kapunk, azonban különböző előjelek esetén éppen az a legfelső bit különbözik, amely eldönti a végeredményt. Mivel kettes komplemensben a negatív számok előjelbitje „1” értékű, ezért a komparátor bekötésekor fel kell cserélni a két szám előjelbitjeit egymással. Ha azonos előjelűek a számok, akkor a felcserélés után is azonosak maradnak, azonban különböző előjelek esetén így éppen a helyes eredmény irányába változik a komparátor kimenete.



Négybites kettes komplemens számok összehasonlítása