

4. hét Az ideális és a valódi építőelemek

4.1. Az ideális és valódi építőelemek

Most ismerkedjünk meg a rendszereket felépítő építőelemekkel. Előtte azonban célszerű néhány alapfogalmat tisztázni.

4.1.1. Alapfogalmak

Ideális helyzet

Az ideális helyzet az, amikor egy kombinációs hálózat kimeneti jele csak a bemeneti jelek aktuális, pillanatnyi értékétől függ. A hálózat a bemeneti jelre időkésés nélkül reagál.

Valóságos helyzet azonban az, hogy az előző állítás nem igaz, a rendszerek a bemeneti jelekre csak késéssel reagálnak. Ennek több oka lehet, s ezek az okok külön – külön és együttesen is okozhatják a rendszer ideális helyzettől való eltérését. Ezek a lehetséges okok a következők:

- ❖ A jelek terjedési sebessége véges, mert
 - Az elektromágneses hullámok terjedési sebessége véges, és
 - Szórt kapacitások és induktivitások okozta késleltetés lép fel.
- ❖ Az egyes hálózati elemek (pl. kapuk) kimeneti jele csak késéssel reagál a bemenet változására, mert
 - a késleltető hatások átmenetileg hibás kimeneti kombinációkat hozhatnak létre.
- ❖ A hibák előfordulása a környezeti változóktól:
 - hőmérséklet,
 - öregedés,
 - stb. függhet

Az ilyen véletlenszerű, rendszertelen hibajelenség neve **hazardjelenség**. Ezt a jelenséget – jellegéből adódóan – **káros jelenségnek** tekinthetjük. Ezért tervezéskor arra kell törekedni, hogy a kombinációs hálózat működése a lehető legnagyobb mértékben független legyen a késleltetési viszonyok alakulásától. Megfelelő tervezési módosításokkal ezt a jelenséget természetesen meg kell szüntetni. Ezt az eljárást „Hazardmentesítés”-nek nevezzük.

4.1.2. Építőelemek

Fontos kérdés, hogy milyen építőelemekből építhetjük meg ezeket a valós rendszereket. Mi az, ami számunkra rendelkezésre áll. Ezen építőkövek ismerete azért is fontos, mert ezek segítségével nagyobb és bonyolultabb rendszereket is fel tudunk építeni.

Ezek az építőelemek már nem tekinthetők „alap építőkövek”-nek, hiszen *a korábban megismert logikai kapuk megfelelő kombinációjából épülnek fel.*

Az építőelemek típusai a következők:

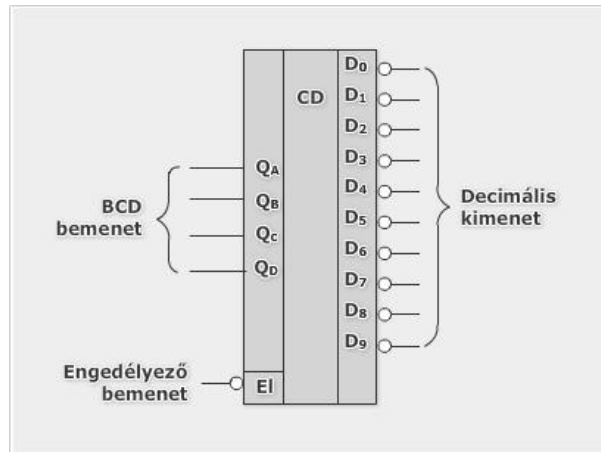
- ❖ Kódoló, Dekódoló
- ❖ Adatút-választók
- ❖ Aritmetikai egységek

4.1.2.1. Kódoló

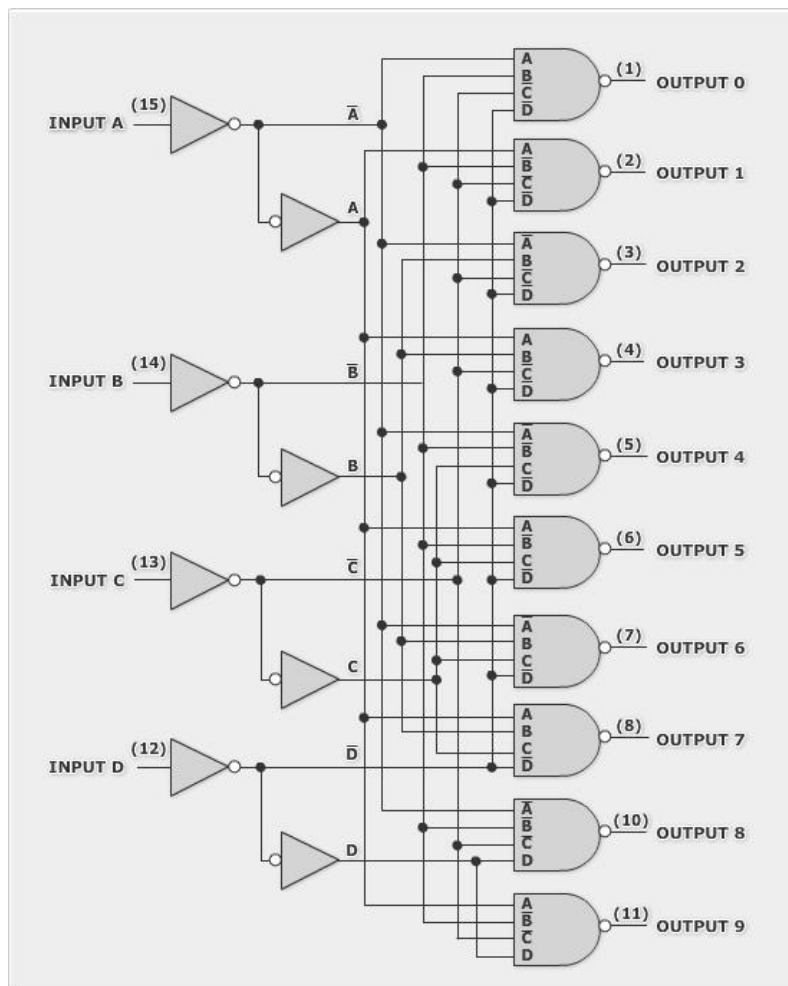
Kódolóknak, dekódolóknak vagy kódátalakítóknak azokat a kapuáramkörökből felépített logikai hálózatokat nevezzük, amelyek az információt az egyik kódrendszerből a másik kódrendszerbe alakítják át.

4.1.2.1.1. Kódoló

Ilyen például a BCD decimális kódoló-dekódoló áramkör. Mint tudjuk, a bináris rendszerben 4 helyiértékkel 16-ig lehet a számokat leírni. A BCD decimális dekódolóban nem használjuk ki mind a 16 lehetőséget (ezt a hexadecimális rendszer teszi meg), mert csak 10 decimális számjegy ábrázolására van szükség. Az áramkör úgy működik, hogy a megfelelő BCD-kódra az áramkör kimeneti oldalán a megfelelő decimális kimeneten logikai 1 (igen) jelenik meg, míg a többi decimális szám kimenete zérus (logikai nem) lesz. Az áramkör része szokott lenni az engedélyező bemenet, ami azt jelenti, hogy az áramkör csak akkor lesz aktív, ha erre a bemenetre logikai igent kapcsolunk. Az ábrán látható engedélyező bemenetnél megjelenő kis körszimbólum egy jelinvertálást jelent, ezért az engedélyezés csak akkor történik meg, ha erre a bemenetre logikai nem értéket kapcsolunk.

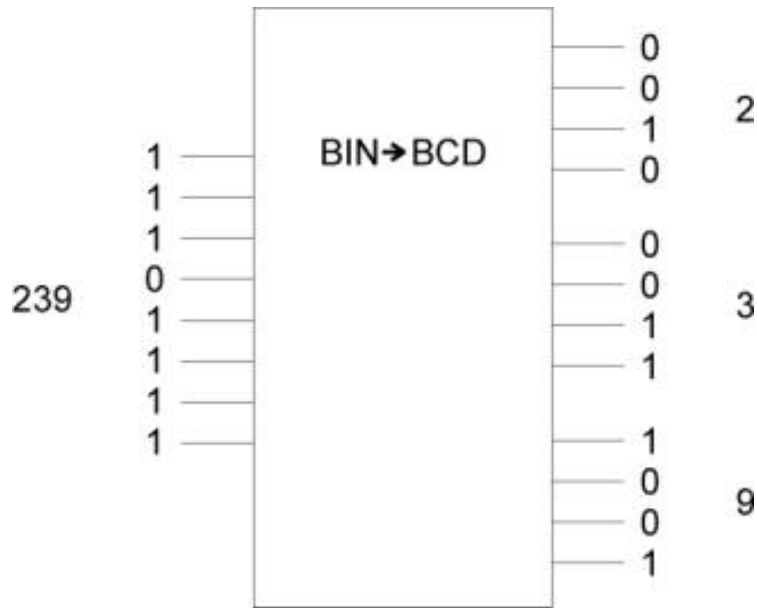


1. ábra BCD - Decimális konverter



2. ábra BCD - Decimális konverter

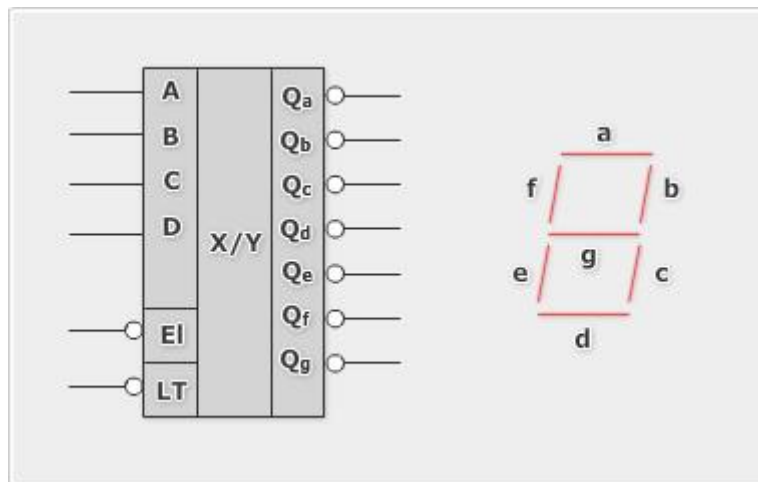
Jó példa ugyanerre az átalakításra a Bináris – BCD átalakító, melynek fontos szerepe van a binaris számok megjelenítésében például. Segítségével lehet a 239-es szám binaris alakját 3db 4 bites tárolású számmá alakítani.



3. ábra Bináris - BCD konverter

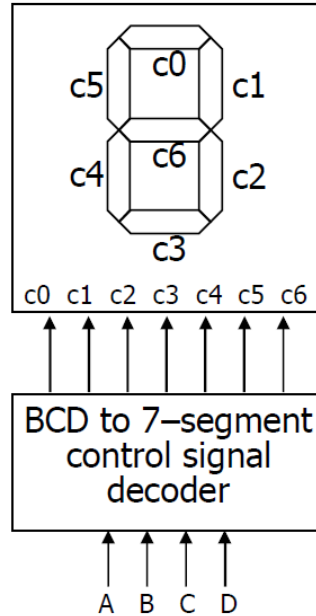
4.1.2.1.2. Dekódolók

Következő példa a kódoló-dekódoló áramkörökre a BCD hétszegmenses kijelzőt meghajtó áramkör. A hétszegmenses kijelzőket a digitális technikában sok helyen használjuk, elsősorban a decimális számok kijelzésére.



4. ábra 7 szegmenses kijelző - dekódoló

A kódátalakító sémáját az ábra mutatja. A 4 bemenethez itt 7 kimenet tartozik. A kimenetekre akkor jut logikai igen, amit a kimeneteknél ábrázolt a kis körszimbólum, ami itt is invertálás műveletet jelent logikai nem értékre változtat, ha az aktuális szegmensnek világítania kell. A hétszegmenses kijelző állapotait az ábra mutatja.



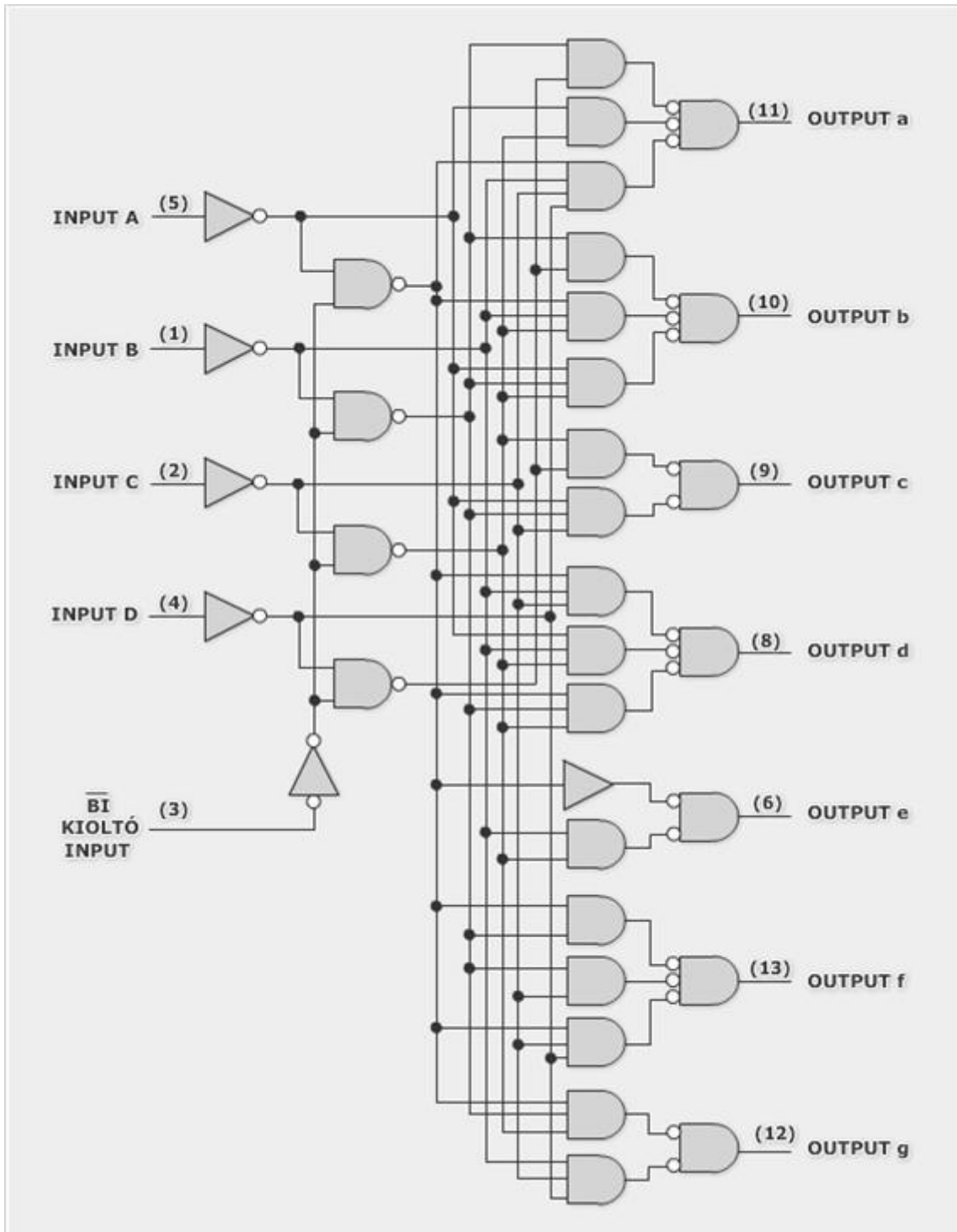
5. ábra 7 szegmenses kijelző

Az igazságtáblázatot az láthatjuk. Például a 0 kijelzésénél a g szegmensén kívül mindegyik szegmensnek világítania kell. Megjegyezzük, hogy az X jelölés azt jelenti, hogy ekkor a logikai változó értéke közömbös, nincs befolyása a kimenetre.

Funkcionális táblázat (T3)												
decimális szám, vagy funkció						kimenet						
	D	C	B	A	\overline{BI}	a	b	c	d	e	f	g
0	L	L	L	L	H	H	H	H	H	H	H	L
1	L	L	L	H	H	L	H	H	L	L	L	L
2	L	L	H	L	H	H	H	L	H	H	L	H
3	L	L	H	H	H	H	H	H	H	L	L	H
4	L	H	L	L	H	L	H	H	L	L	H	H
5	L	H	L	H	H	H	L	H	H	L	H	H
6	L	H	H	L	H	L	L	H	H	H	H	H
7	L	H	H	H	H	H	H	H	L	L	L	L
8	H	L	L	L	H	H	H	H	H	H	H	H
9	H	L	L	H	H	H	H	H	L	L	H	H
10	H	L	H	L	H	L	L	L	H	H	L	H
11	H	L	H	H	H	L	L	H	H	L	L	H
12	H	H	L	L	H	L	H	L	L	L	H	H
13	H	H	L	H	H	H	L	L	H	L	H	H
14	H	H	H	L	H	L	L	L	H	H	H	H
15	H	H	H	H	H	L	L	L	L	L	L	L
BI	X	X	X	X	X	L	L	L	L	L	L	L

1.2.1.6. ábra Forrás: Texas Instruments

Az, hogy ez az áramkör is kombinációs hálózat, a következő ábrából jól látható, az áramkör kapukból épül fel, és memóriaelemet nem tartalmaz.



Forrás: Texas Instruments

6. ábra A 7 segmenses kijelző kapcsolási rajza

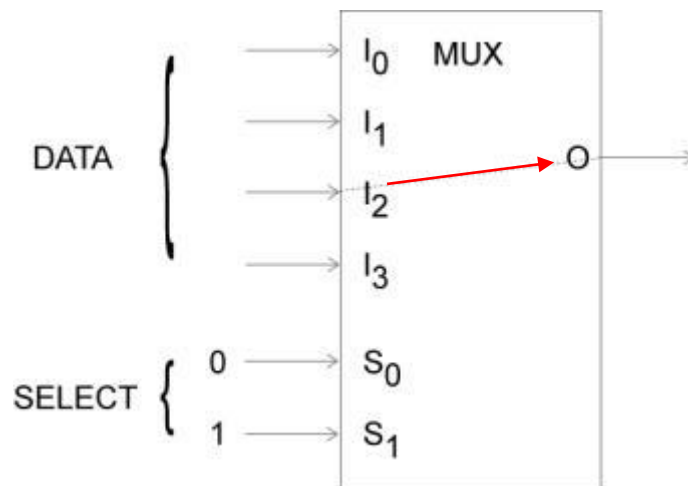
A kijelzés csak akkor működik, ha a „blanking” bemenetre logikai igen értékű jelet kapcsolunk.

4.1.2.2. Adatút-választók

Ezek az áramkörök kapuáramkörökből megvalósított kombinációs hálózatok. IC tokokban hozzák forgalomba és közepes bonyolultságú feladatok ellátására alkalmasak.

4.1.2.2.1. Multiplexer

Egy multiplexer (vagy mux vagy ritkán muldex) egy speciális logikai eszköz, amely két vagy több bemenő jel közül a címező (vezérlő) jel alapján egyet a kimenetére ad. Egy adott pillanatban igaz ez az állítás, de hosszabb időt tekintve képes a jelek egyesítésére (idő eltolás használatával). Az analóg és digitális jelek multiplexálására külön analóg és digitális multiplexer áramkörök szolgálnak.

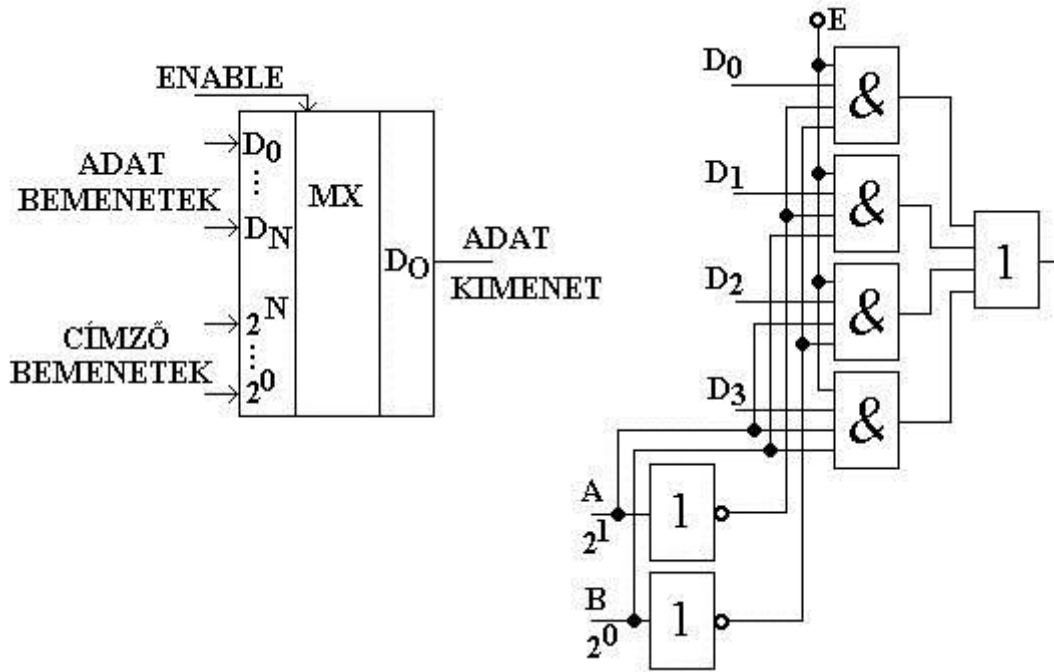


7. ábra A multiplexer adattválasztása

Ezek olyan digitális áramkörök, amelyek **több adatbemenettel** (2, 4, 8, 16) és **egy adatkimenettel rendelkeznek**. Vannak címző bemeneteik, amelyekre adott bináris címek segítségével ki lehet választani, hogy melyik adatbemeneten lévő információ kerüljön át a kimenetre. Ezeken kívül van egy engedélyező bemenete is, amely általában 1-re vagy 0-ra aktív .

Tehát:

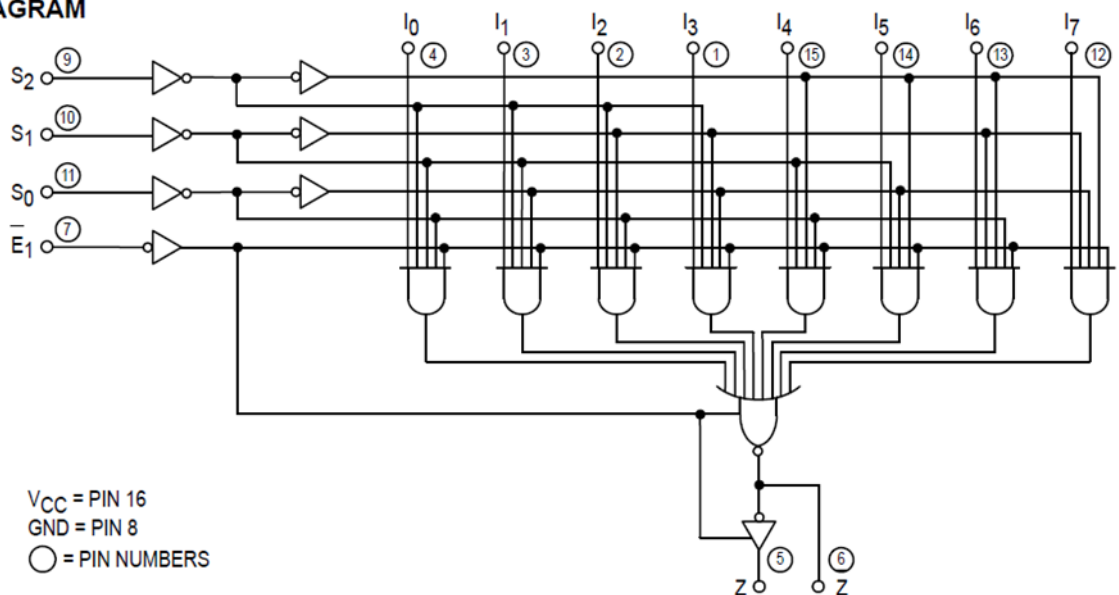
- ❖ A kiválasztó-vezérlő jel (SELECT) függvényében a bemeneti jelek (DATA) közül az egyiket a kimenetre irányítja
- ❖ n számú SELECT jel értelem szerűen 2^n adatbemenet közül választhat
- ❖ Az egység általában a működést engedélyező bemenettel is rendelkezik



8. ábra Egy multiplexer elvi rajza és felépítése

Pl.: az ábrán látható 2 címző bemenettel rendelkező multiplexer címző vezetékeire 00 kódot adva a D_0 -ra adott információ kerül a kimenetre, ha 01-et kapcsolunk a címvezetékekre, akkor a D_1 -re adott információ kerül a kimenetre...stb.

LOGIC DIAGRAM



9. ábra Egy konkrét multiplexer (74LS251)

TRUTH TABLE

E ₀	S ₂	S ₁	S ₀	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	Z	Z
H	X	X	X	X	X	X	X	X	X	X	X	(Z)	(Z)
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	H	L
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
L	H	H	H	X	X	X	X	X	X	X	L	H	L
L	H	H	H	X	X	X	X	X	X	X	H	L	H

H = HIGH Voltage Level
 L = LOW Voltage Level
 X = Don't Care
 (Z) = High impedance (Off)

10. ábra A multiplexer igazságtáblázata

Szükség esetén a multiplexerek összekapcsolhatók, bővíthetők. Ezzel az adatbemenetek száma többszörözhető.

Felépítésénél fogva elsősorban **adatkiválasztásra, logikai függvény megvalósítására** (MSI kombinációs hálózat) és **párhuzamos-soros adat átalakításra** használható.

Feladatok, melyeket a multiplexer el tud végezni:

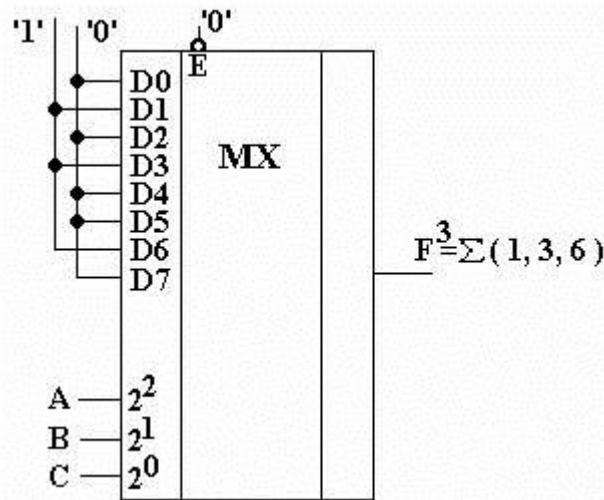
1) Párhuzamos-soros adat átalakítás

Egy kódszó minden bitje egyszerre áll rendelkezésre (pl.: egy számítógép adatvezetékein), és ezt a multiplexer adatbemeneteire kötjük. A címző bemenetekre adott címek hatására **az adatbemenetekről órajellel ütemezve mindig egy-egy bit kerül át a kimenetre**. Egy kódszó átviteléhez annyi órajel szükséges, ahány bites az információ.

2) Kombinációs hálózat megvalósítása

Adott a logikai függvény **diszjunktív szabályos alakja számjegyes formában**. Ebből látható, hogy hány változós a függvény. A legegyszerűbb megvalósítási mód az, hogy ugyanennyi címző bemenettel rendelkező multiplexert használunk a megvalósításhoz. A címző bemenetekre kötjük a függvény változóit (A, B, C...).

Azokra az adatbemenetekre 0-t kötünk, amelyek nem szerepelnek a függvényben, azokra pedig 1-t, amelyek szerepelnek a függvényben. Egy logikai függvény megvalósítása a 8/3/1-es multiplexerrel a 3. ábrán látható.



11. ábra Egy logikai függvény megvalósítása multiplexer felhasználásával

Megvalósítható a kombinációs hálózat eggyel kevesebb címző bemenettel rendelkező multiplexerrel is, de akkor szükség van egy inverterre is, és az egyik változót ennek a bemenetére adjuk.

3) Adatkiválasztás multiplexerrel

Adott négy mérőkészülékünk (érezkelőnk) telepítve különböző helyeken, és tételezzük fel, hogy mindegyik folyamatosan szolgáltatja az információt. Ezek egy multiplexer 1-1 bemenetére kerülnek, és a címző bemenetekre adott kód segítségével ki lehet jelölni, hogy melyikről kerüljön tovább az információ a kimenetre (pl.: rögzítésre, megjelenítésre ...).

A multiplexer nem képes a jeleket egyesíteni, mert a több bemenetből egyet kiválaszt és a kimenetre adja (a kép baloldalán látható- a képen multiplexálás és demultiplexálás látható). Működési elve korai telefonközpontokhoz hasonlítható, maga a multiplexer a telefonos kisasszonyok voltak. Némely írásokat ezzel a magyarázattal megcáfolhatjuk, ugyanis ha telefonálunk, akkor nem halljuk egyesítve az összes telefonközpontba beérkező beszédet.

Egy példa a multiplexerre: a telefonközpont. A telefonközpont a hívót és hívottat összekötve egy időbeli multiplexálást végez. Szintén multiplexer egy műsorszóró műhold.

A multiplexereket, működési módjuk alapján a következő típusokba sorolják:

- ❖ időosztásos multiplexerek
- ❖ frekvenciaosztásos multiplexerek
- ❖ hullámhossz-osztásos multiplexerek.
- ❖ kód-osztásos multiplexerek

Időosztásos multiplexálás esetén a bemeneti jelek felváltva egyetlen közös kimenetre kapcsolódnak. Ilyen például az ábrán látható mux forgó kapcsolóval történő megvalósítása, vagy egy elektromechanikus telefonközpont.

Frekvenciaosztásos multiplexer a bemeneti jeleket egyidőben továbbítja ugyanazon az átviteli úton, azok az elfoglalt frekvencia tartomány alapján különíthetők el egymástól, mint például az analóg kábeltévé rendszer. Hullámhossz-osztásos multiplexer például az optikai átvitel esetében használatos (lényegét tekintve ez is frekvenciaosztás).

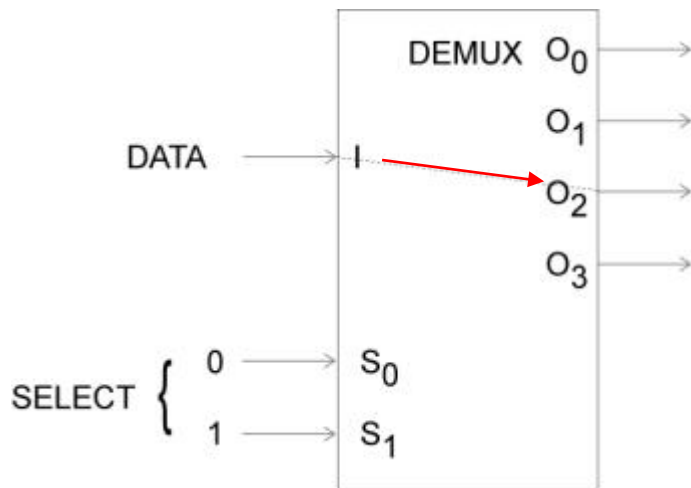
A frekvenciaosztásos és időosztásos multiplexerek használata a "klasszikus" telefon technikában fejlődött ki, a kód-osztásos multiplexer technikák pedig a digitális átvitel elterjedésével nyertek egyre nagyobb teret (szigorúan véve a kód-osztásos multiplexer is időbeli multiplexálást végez).

Egyértelműen analóg technológiák a frekvencia- és hullámhossz-osztásos multiplexálás, míg digitális technológia a kód-osztásos multiplexálás. Az időosztásos technológia mindkét csoportba besorolható. Nagyobb rendszerek felépítése érdekében ezeket az elveket egymás mellett egyidejűleg is alkalmazhatják.

Gyakran építik egybe a multiplexert és a demultiplexert egy berendezésbe, amit szintén "multiplexer"-nek neveznek. Azonban az egybeépített egységen belül mindkét résznek megvan a saját be- és kimenete, mivel egy multiplexált jelet csak egy demultiplexer tud visszaalakítani, két multiplexer hiába van egymással ki- és bemeneten összekötve, a megoldás működésképtelen.

4.1.2.2.2. Demultiplexer

A vételi oldalon szükség van egy olyan egységre, amely elvégzi a visszaalakítás műveletét, ez a demultiplexer vagy demux. Ilyen funkciók lehetnek a zajszűrés, jelhelyreállítás stb. A demultiplexáláshoz szükség van ugyanarra a vezérlő jelre, amit a multiplexer is használt. Ez a vezérlő jel mondja meg, hogy melyik be- kimenetet használja.

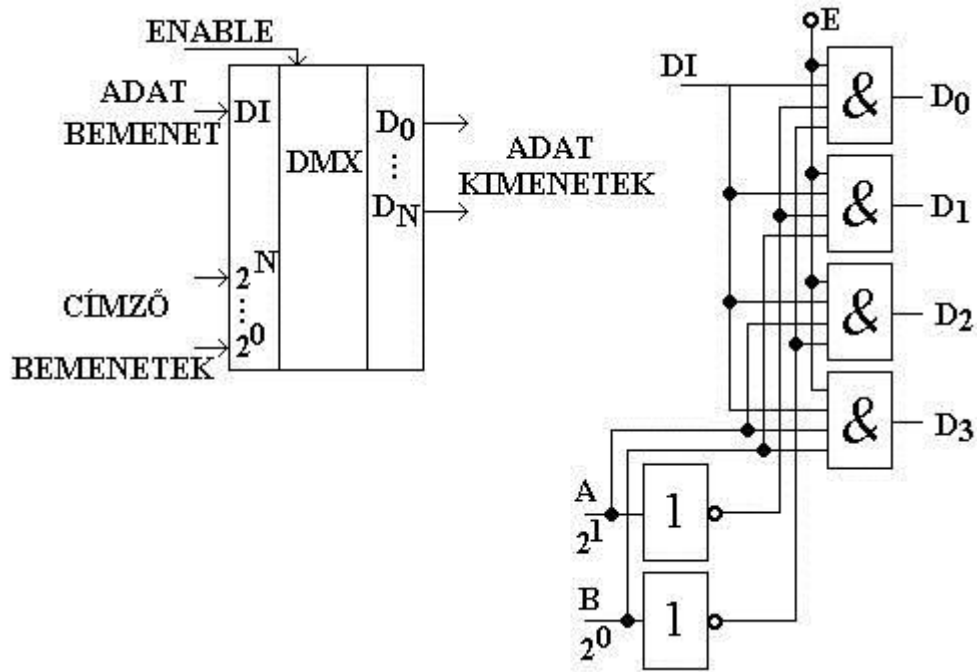


12. ábra A demultiplexer

Ezek olyan digitális áramkörök, amelyek **egy adatbemenettel és több adatkimenettel** (2, 4, 8, 16) **rendelkeznek**. Vannak címző bemeneteik, amelyekre adott bináris címek segítségével ki lehet választani, hogy melyik adatkimenetre kerüljön át az információ a bemenetről. Ezen kívül van egy engedélyező bemenete is, amely lehet 0-ra vagy 1-re aktív.

Tehát:

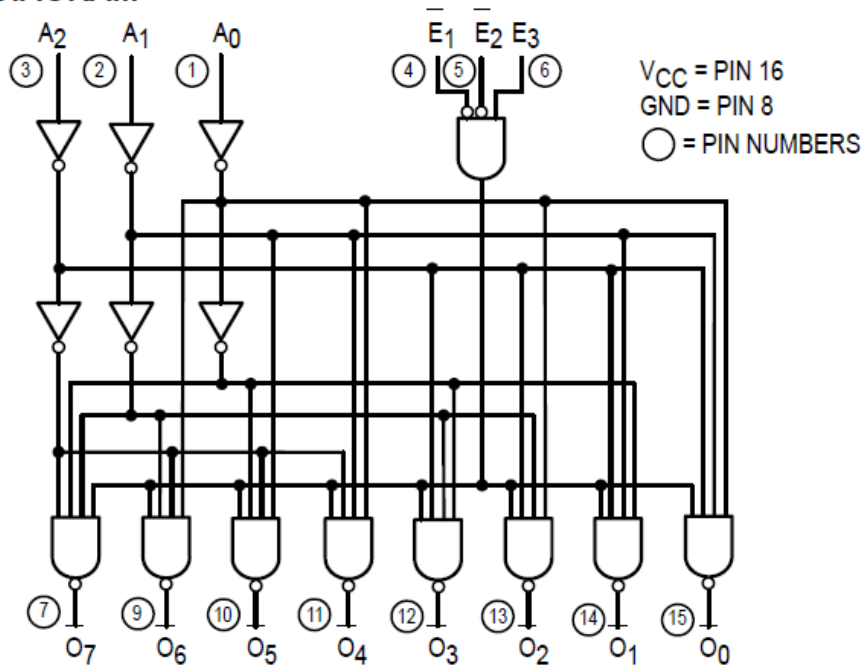
- ❖ A kiválasztó-vezérlő jel (SELECT) függvényében a bemeneti jelet (DATA) a kimenetek egyikére irányítja
- ❖ n számú SELECT jel értelem szerűen 2^n adatkimenet közül választhat
- ❖ Az egység általában a működést engedélyező bemenettel is rendelkezik



13. ábra Egy demultiplexer jelölése és felépítése

Pl.: egy 2 db címző bemenettel rendelkező DMX címző bemeneteire 00-t kapcsolva a DI-re adott információ a D_0 kimenetre kerül át. Ha 10-t kapcsolunk rá, akkor a D_2 kimenetre kerül át. A DMX-eket negált kimenetekkel és negált bemenettel is gyártják, amely annyit jelent, hogy ezek 0-ra aktívak (hatásosak).

LOGIC DIAGRAM



14. ábra Egy konkrét demultiplexer (74LS138)

TRUTH TABLE

INPUTS						OUTPUTS							
E ₁	E ₂	E ₃	A ₀	A ₁	A ₂	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
 L = LOW Voltage Level
 X = Don't Care

15. ábra Egy konkrét demultiplexer (74LS138)

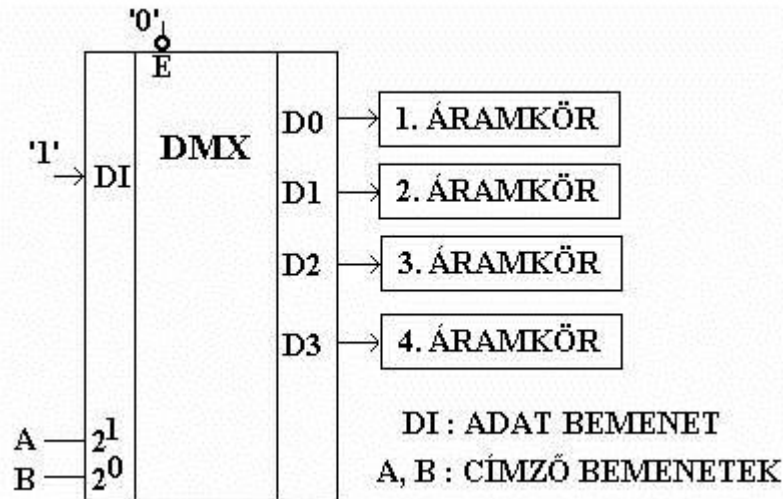
Felépítésénél fogva elsősorban **adatelosztásra, kód átalakításra és soros-párhuzamos adat átalakításra** használhatjuk.

1) Soros-párhuzamos adat átalakítás

Egy kódszó bitjei sorosan állnak rendelkezésre, pl. egy soros kimenetű átmeneti tárban és ütemezve egyenként kerülhetnek a DMX adatbemenetére. A címző bemenetekre adott bináris címek hatására **az adatbemenetről minden ütemre a megcímzett kimenetre kerül 1-1 bitnyi információ.** Egy kódszó átviteléhez annyi órajel szükséges, ahány bites az információ.

2) Adatok elosztása

Az adatbemenetre állandóan 1-et adunk és a címző bemenetekkel választhatjuk ki, hogy melyik kimenetre kerüljön át ez az 1-es, ezzel az erre kötött áramkört engedélyezzük. Mivel az áramkörök tok engedélyezői többnyire 0-ra aktívak, ezért itt inverz kimenettel rendelkező demultiplexereket használnak. Pl. így történik a memória modulok kiválasztása (címezése) és a portok kiválasztása stb.



16. ábra Áramkörök kiválasztása demultiplexerrel

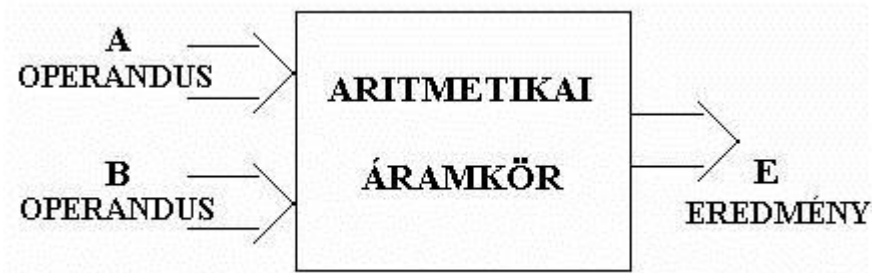
3) Kódátalakítás

Egy megadott kódrendszerben lévő kódszót alakít át egy másik kódrendszerbeli kódszóvá. Pl.: egyszerűen megvalósítható egy 1/4/16-os DMX-rel egy bináris-decimális vagy egy bináris-hexadecimális kódátalakító. A címző bemenetre kapcsolt bináris számnak megfelelő decimális kimenet lesz az aktív.

4.1.2.3. Aritmetikai egységek

Ebbe a csoportba olyan digitális áramkörök tartoznak, amelyekkel összeadási, kivonási, szorzási, osztási és egyéb műveletek végezhetőek el.

A műveletvégzéshez szükséges adatokat (általában bináris, vagy BCD kódú bináris szám) az áramkör bemeneteire vezetjük, és a kimeneten kapjuk meg az eredményt valamilyen kód formájában.



4.1.2.3.1. Digitális komparátor

Vegyük egy további példát, ahol megtanulhatjuk a kombinációs hálózatok felírását, természetesen a korábban megismert felírási lehetőségekkel.

❖ Szöveges megfogalmazás

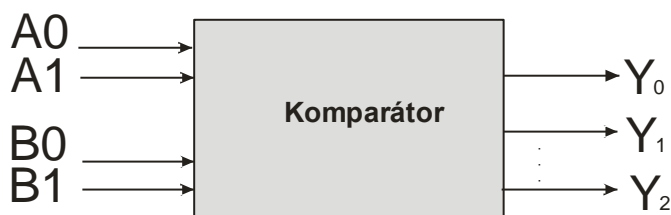
Legyen egy digitális komparátorunk, melynek az a feladata, hogy két binárisan felírt számot hasonlítsen össze. A két szám legyen eltárolva két biten.

❖ Blokk

Ebben az esetben a rendszer bemenete az a négy bit ahol a két számot eltároljuk, kimenete pedig az a három változó, melyek igaz értékeivel képesek eldönteni a két szám viszonyát. Erre azért van szükség, mert a három kimenet állapota csak igaz vagy hamis lehet, egyéb értéket nem vehet fel. Így az igaz állapot „helyzete” mondja meg, hogy mi a két szám viszonya.

Ez azt jelenti, hogy

- $Y_0 = 1$ ha $A > B$ egyébként 0
- $Y_1 = 1$ ha $A = B$ egyébként 0
- $Y_2 = 1$ ha $A < B$ egyébként 0



17. ábra A komparátor.

A rendszerben legyen (Azért, hogy az előbb tanult felírási módot tudjuk követni $A_0 = D$, $A_1 = C$, $B_0 = B$ és $B_1 = A$.)

❖ **Igazságtáblázat**

Ezután felrajzoljuk a rendszer igazságtáblázatát, amely 4 független változót és 3 függő változót tartalmaz, ezért 7 oszlopa lesz, viszont 4 független változójából következik, hogy 16 sorral fog rendelkezni. Felírjuk A, B, C, D lehetséges kombinációit, és kitöltjük a táblázatot.

i	A	B	C	D	Y₀	Y₁	Y₂
0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	0	1
3	0	0	1	1	0	0	1
4	0	1	0	0	1	0	0
5	0	1	0	1	0	1	0
6	0	1	1	0	0	0	1
7	0	1	1	1	0	0	1
8	1	0	0	0	1	0	0
9	1	0	0	1	1	0	0
10	1	0	1	0	0	1	0
11	1	0	1	1	0	0	1
12	1	1	0	0	1	0	0
13	1	1	0	1	1	0	0
14	1	1	1	0	1	0	0
15	1	1	1	1	0	1	0

18. ábraA komparátor igazságtáblázata

❖ **Logikai függvények**

A logikai függvény felírásához csak egy egyenletrendszert veszünk most figyelembe, hogy egyszerűbben fel tudjuk írni a függvényt.

Tekintsük most csak azt az esetet, amikor

– $Y_0 = 1$ ha $A > B$ egyébként 0

– Ekkor az igazságtáblában az alábbi sorok lesznek fontosak.

	i	A	B	C	D	Y_0	Y_1	Y_2
	0	0	0	0	0	0	1	0
	1	0	0	0	1	0	0	1
	2	0	0	1	0	0	0	1
	3	0	0	1	1	0	0	1
Y1	4	0	1	0	0	1	0	0
	5	0	1	0	1	0	1	0
	6	0	1	1	0	0	0	1
	7	0	1	1	1	0	0	1
Y2	8	1	0	0	0	1	0	0
Y3	9	1	0	0	1	1	0	0
	10	1	0	1	0	0	1	0
	11	1	0	1	1	0	0	1
Y4	12	1	1	0	0	1	0	0
Y5	13	1	1	0	1	1	0	0
Y6	14	1	1	1	0	1	0	0
	15	1	1	1	1	0	1	0

Az ebből felírható részegyenletek (soronként meghatározott mintermek):

$$Y_1 = \overline{A}BC\overline{D}$$

$$Y_2 = A\overline{B}C\overline{D}$$

$$Y_3 = \overline{A}B\overline{C}D$$

$$Y_4 = A\overline{B}C\overline{D}$$

$$Y_5 = A\overline{B}C\overline{D}$$

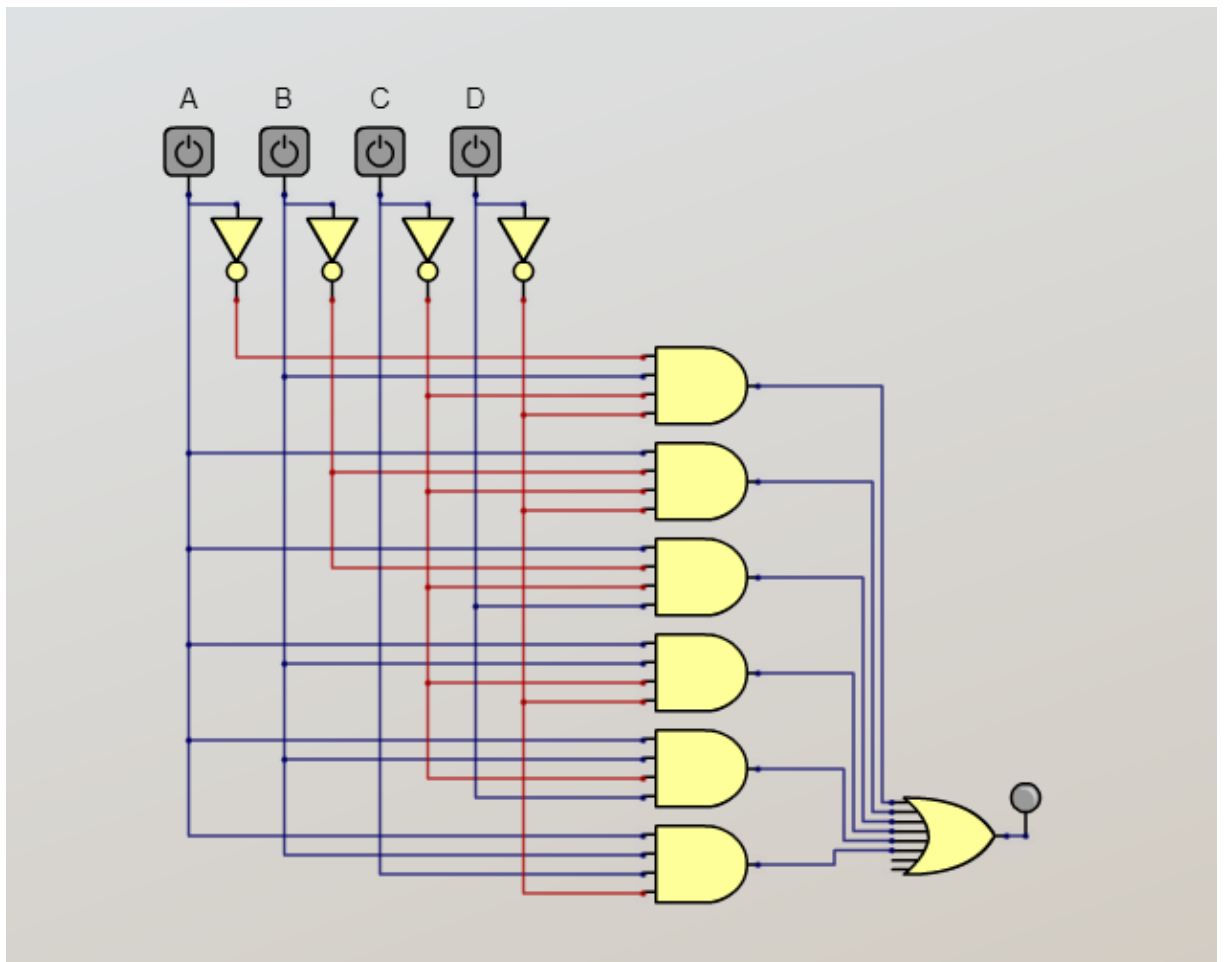
$$Y_6 = ABC\overline{D}$$

Tehát az egyenlet:

$$Y = \overline{A}BC\overline{D} + A\overline{B}C\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}C\overline{D} + ABC\overline{D}$$

❖ Kapcsolási rajz

Ebből az egyenletből felépíthetjük a hálózatot is. Ehhez 6 darab És és egy VAGY kapura van szükségünk. Ezen kívül természetesen minden bemenetnél képeznünk kell a negáltat is.



19. ábraA komparátor kapcsolási rajza

❖ Karnaugh - tábla

A Karnaugh tábla felírásához szükség van az előbbi egyenletre.

$$Y = \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D}$$

Ennek elemeit keressük meg a táblában, és írunk a helyére egyes értéket.

AB	CD	C				
		00	01	11	10	
A	00	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	B
	01	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	
	11	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	
	10	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	
		D				

AB	CD	C				
		00	01	11	10	
A	00					B
	01	1				
	11	1	1		1	
	10	1	1			
		D				

Ezután a tábla egyszerűsíthető: (Pirossal a kieső tagok):

AB	CD	C				
		00	01	11	10	
A	00	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	B
	01	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	
	11	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	
	10	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}D$	$\overline{A}B\overline{C}\overline{D}$	
		D				

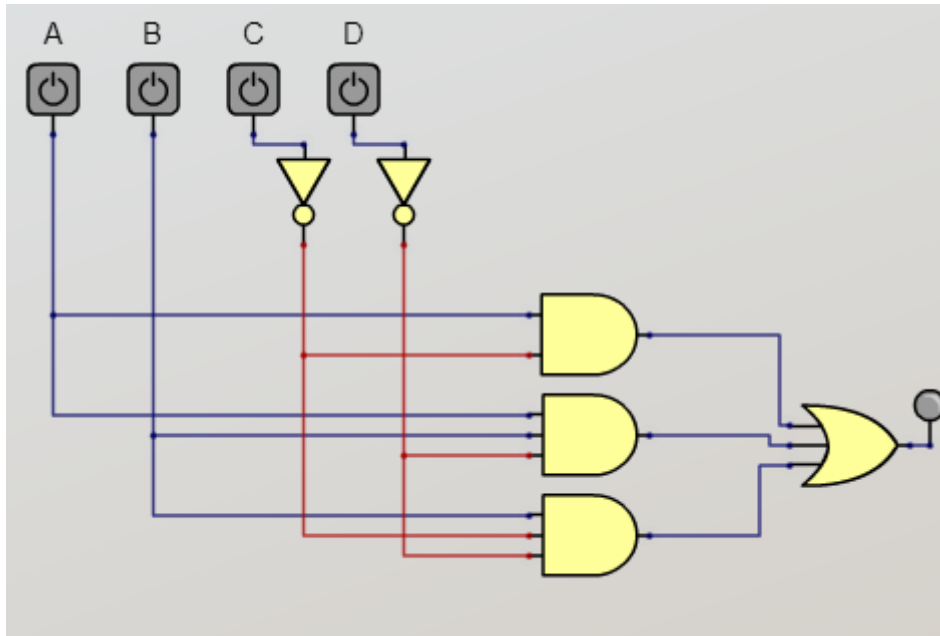
AB	CD	C				
		00	01	11	10	
A	00					B
	01	1				
	11	1	1		1	
	10	1	1			
		D				

Az egyszerűsítés után kapható egyenlet:

$$Y = \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D}$$

$$Y = \overline{A}C + \overline{A}B\overline{D} + \overline{B}C\overline{D}$$

Ezzel az egyszerűsítéssel tényleg egy könnyebben kezelhető rendszert kaphatunk: így már csak 3 ÉS kapu és egy VAGY Kapu szükséges a rendszer felépítéséhez. A négy negált helyett pedig csak kettőt használunk.

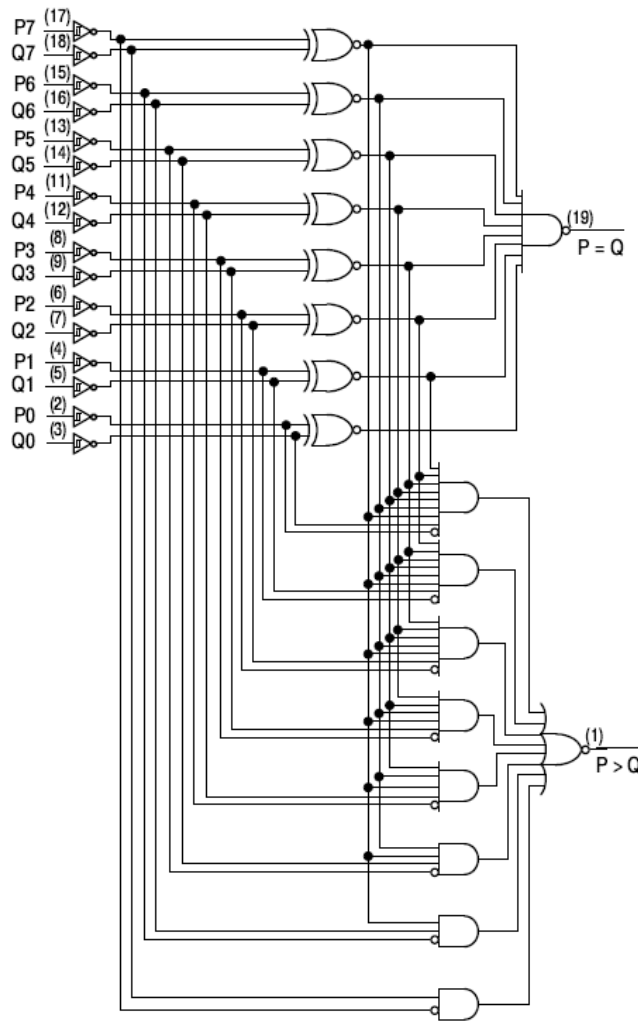


20. ábra Az egyszerűsített kapcsolási rajz

Konkréten pl. egy két binaris számot összehasonlítását elvégző komparátor a következőképpen valósítható meg.

INPUTS			OUTPUTS	
DATA	ENABLES			
P, Q	$\overline{G}, \overline{GT}$	$\overline{G2}$	$\overline{P = Q}$	$\overline{P > Q}$
P = Q	L	L	L	H
P > Q	L	L	H	L
P < Q	L	L	H	H
X	H	H	H	H

21. ábra A digitális komparátor igazságtáblázata (74LS682)



22. ábra A digitális komparátor megvalósítása: (két bináris szám) (74LS682)

4.1.2.3.2. Összeadó

❖ Az összeadás definíciója

Olyan áramkörök, amelyek a bemeneteikre kerülő két operandust bitenként összeadják, figyelembe véve az egyes helyi értékeken keletkező átvitelt is, és az eredményt a keletkezett átvittel együtt megjelenítik a kimenetükön. Ennek leggyakrabban alkalmazott formája a teljes összeadó.

A teljes összeadó megértéséhez fontos a számláló elvének megértése is.

A számlálás során, ha decimálisan írjuk a számokat, 10 darab számjeggyel írjuk le az összes lehetséges mennyiséget. A használt számjegyek 0,1,2,3,4, 5, 6, 7, 8, 9, .

9 után azonban elfogynak a számjegyek. Mi ilyenkor a teendő?

A rendszerben ilyenkor túlcsoordulás keletkezik: egyet lépünk a helyiértéken, és újra kezdjük a számlálást.

01
02
03
04
05
06
07
08
09
10 ←átvitel / túlcsoordulás
11
12
...

23. ábra A decimális számláló túlcsoordulása

Ugyanez történik bináris rendszerben is, azzal a különbséggel, hogy mivel itt csak két számjegyük van, ezért hamarabb fog jelentkezni a túlcsondulás.

```
0000
0001
0010
0011    ←átvitel / túlcsondulás
0100
0101
0110
0111    ←átvitel / túlcsondulás
1000
1001
1010
1011
1100
1101
1110    ←átvitel / túlcsondulás
1111
```

24. ábra a bináris számláló túlcsondulása

Az összeadás során is ezzel a túlcsondulással kell számolni. Ezért a decimális esetben a papíron végzett összeadásnál, jobbról balra haladva sorra összeadjuk az egyes számjegyeket. Ahol kilencnél nagyobb eredményt kapunk, ott hozzáadjuk a maradék egyest az eggyel nagyobb helyiértékű számjegyekhez.

$$\begin{array}{r} 2 5 6 \\ + 3 1 7 \\ \hline 5 7 (1) 3 \end{array}$$

25. ábra Összeadás decimális szám esetén

A bináris esetben pedig ugyanígy járunk el, csak a felhasználható számjegyek 0 és 1. Ahol 1-nél nagyobb eredményt kapunk, ott hozzáadjuk a maradék egyest az eggyel nagyobb helyiértékű számjegyekhez.

$$\begin{array}{rcccccccc}
 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & & (74) \\
 + & 1 & 0 & 0 & 1 & 1 & 1 & 1 & & + (79) \\
 \hline
 1 & (1) & 0 & 0 & 1 & (1) & 1 & (1) & 0 & (1) & 0 & 1 & & (153)
 \end{array}$$

26. ábra Összeadás kettes számrendszerben

A **teljes összeadónak** 3 bemenete van, amelyekre a két összeadandót és az előző helyiértéken keletkezett átvitelt vezetjük. Ilyen 1 bites modulok felhasználásával készíthetünk több bites összeadó áramköröket.

Mivel a kettes számrendszerben történő összeadás esetén az átviteleket is figyelembe véve:

Összeadás	Példa
$0 + 0 = 0$	$ \begin{array}{r} 1011_2 \\ + \quad 11_2 \\ \hline 1110_2 \end{array} $
$0 + 1 = 1$	
$1 + 0 = 1$	
$1 + 1 = 10$	

Kivonás	Példa
$0 - 0 = 0$	$ \begin{array}{r} 1011_2 \\ - \quad 11_2 \\ \hline 100_2 \end{array} $
$0 - 1 = -1$	
$1 - 0 = 1$	
$1 - 1 = 0$	

Szorzás	Példa
$0 \cdot 0 = 0$	$ \begin{array}{r} 1010_2 \\ \cdot \quad 11_2 \\ \hline 11110_2 \end{array} $
$0 \cdot 1 = 0$	
$1 \cdot 0 = 0$	
$1 \cdot 1 = 1$	

Osztás	Példa
$0 / 0 = \text{n.def.}$	$ \begin{array}{r} 1010_2 \\ / \quad 10_2 \\ \hline 101_2 \end{array} $
$0 / 1 = 0$	
$1 / 0 = \text{n.def.}$	
$1 / 1 = 1$	

27. ábra A műveletek kettes számrendszerben

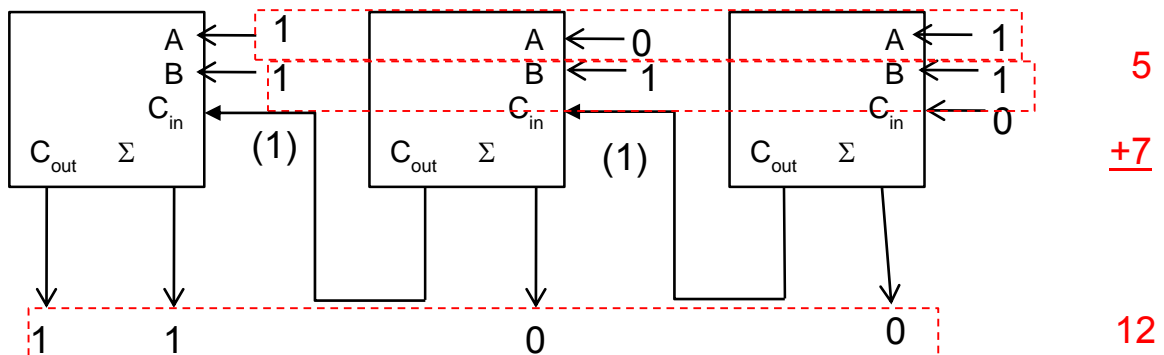
Ezért a rendszer igazságtáblázata a következőképpen írható fel:

C_{in}	A	B	Σ	C_{out}
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

28. ábra Az összeadó igazságtáblázata

Az összeadó (Áramköri felépítése az alábbi ábrán látható.) tehát

- ❖ Két bináris szám egy-egy bitjét adja össze
- ❖ Figyelembe veszi az egyel kisebb helyiértékről érkező átvitelt.
- ❖ Ha nála is keletkezik átvitel, akkor továbbítja azt.
- ❖ Több egybites összeadót összekapcsolva két bármilyen hosszú bináris számot összeadhatunk
- ❖ Kettes komplementes kóddal kivonóként is használható, az utolsó átvitel használata nélkül



29. ábra Az teljes összeadó

A teljes összeadó működését tekintve kétféle lehet: soros vagy párhuzamos.

❖ **Soros összeadó**

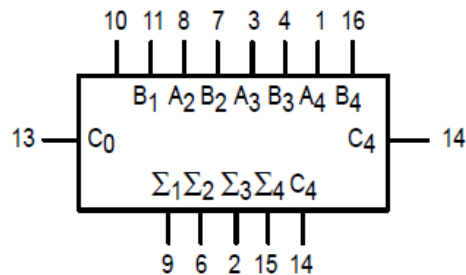
Az egyes helyiértékeken lévő bitek összeadása időben egymás után történik, kezdve a legkisebb helyiértékkal. Így csak egy 1 bites teljes összeadó áramkörre, regiszterekre és egy 1 bites késleltető tárolóra van szükség. Ezzel a módszerrel viszonylag hosszú idő szükséges egy nagyobb bitszámú művelet elvégzéséhez.

❖ **Párhuzamos összeadó**

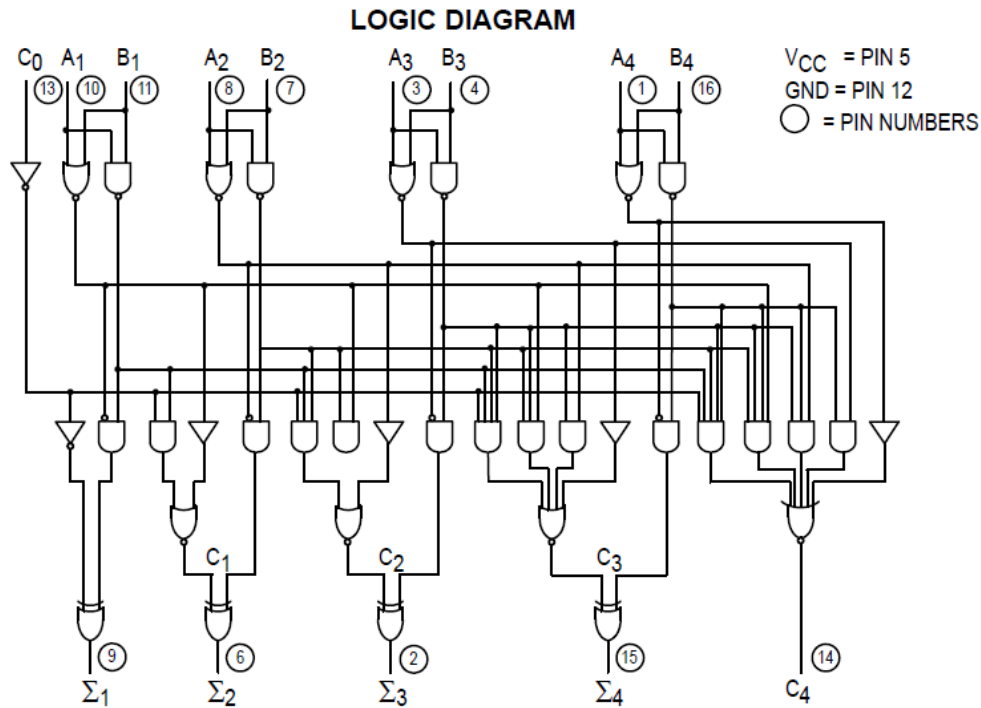
Az egyes helyiértékeken lévő bitek összeadása azonos időben történik, de természetesen az átvitelbitek megjelenéséig szükség van valamekkora továbbterjedési időre, vagy valamilyen áramkörre, amely az átvitel bitek létrehozását gyorsítja.

❖ **Példa**

Jó példa az összeadóra a **74LS8** amely egy **4 bites teljes összeadó**.



30. ábra A négy bites összeadó blokkja



31. ábra A négy bites számláló kapcsolási rajza

	C0	A1	A2	A3	A4	B1	B2	B3	B4	Σ1	Σ2	Σ3	Σ4	C4
Logic Levels	L	L	H	L	H	H	L	L	H	H	H	L	L	H
Active HIGH	0	0	1	0	1	1	0	0	1	1	1	0	0	1
Active LOW	1	1	0	1	0	0	1	1	0	0	0	1	1	0

(10+9 = 19)
(carry+5+6 = 12)

32. ábra A négy bites számláló igazságtáblázata

4.1.2.3.3. Kivonó

❖ A kivonás definíciója

A kivonás művelete matematikailag a **kivonandó szám mínusz egyszeresének hozzáadásával végezhető el**. Erre a 2-es komplement kódot használják. Az áramköri megoldás lényege az, hogy a kivonandó minden bitjét negáljuk és ehhez még 1-et hozzáadunk. Majd az így keletkezett operandust és a másik operandust vezetjük be egy összeadó áramkörbe. Az így kapott eredmény 2-es komplement kódban áll rendelkezésre. A kivonandó komplementjét előállíthatjuk párhuzamosan vagy sorosan.

A kivonó áramkörök szintén lehetnek soros vagy párhuzamos működésűek.

❖ **Példa:**

A számítógépek a kivonást a kettes komplement segítségével végzik. A kisebbítendőhöz hozzá kell adni a kivonandó kettes komplementerét. Ez két lépést jelent. Át kell váltani a kivonandót kettes számrendszerbe, majd a bitjeit át kell billenteni, majd 1-et hozzá kell adni. Az így kapott számot és a kisebbítendő bináris számát össze kell adni. például vonjuk ki a 8-ból az 5-öt.

$$8-5=3$$

8: 1000 5: 0101

5 I: 1010 (negálás)

5 II: 1011 (negálthoz adunk 1-et)

Ezeket most össze kell adni:

1000

+1011

————

10011

A kapott eredmény az 10011. Itt viszont keletkezik egy felesleges bit, túlcordulás, amit le kell választani, arra nincs szükség, ezt levágjuk.

1 | 0011

Az így kapott számot, ha vissza váltjuk 10-es számrendszerbe, akkor megkapjuk az eredményt.

$$0011 = 1+2 = 3$$

4.2. A nemidealitások

A legtöbb hálózat azonban nem fog ideálisan viselkedni. Ennek több oka lehet.

4.2.1. A nemidealitások okai

Alapvetően 3 olyan okról beszélhetünk, melyek a nem idealitásokat fogják eredményezni.

- 1) sebességjellemzők
- 2) jelterjedési ill. késési idő
- 3) teljesítményjellemzők

Ezek a paraméterek fontos tényezők a rendszer tervezése során. A rendszer egyenletének megfelelő egyszerűsítésével a jelenség könnyen kiküszöbölhető. Ezt az egyszerűsítést a legegyszerűbb a Karnaugh – táblával tudjuk létrehozni.

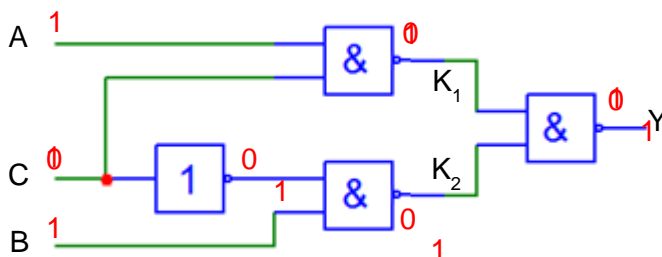
4.3. A kombinációs hálózatok hazard jelenségei

4.3.1. A hazard fogalma

A hazardok a digitális hálózatok tervezés folyamán fellépő jelenségek. Olyan működési hibák, amelyek kiszámíthatatlan módon jelentkeznek. Gyakran a megvalósításhoz felhasznált elemek, vagy bekötésük késleltetése okozza. Ezt a jelenséget hazardnak nevezzük. Kiküszöböléséhez módosítanunk kell a függvényt vagy másik hálózatot kell terveznünk, amely lefedi ezeket a hazardokat, ezzel megakadályozva, hogy a hazard kijusson a kimenetre. Ez utóbbi megoldás azonban növeli a hálózat szintjét, ami késleltetést jelent.

Vegyünk egy egyszerű példát a jelenségre:

Az alábbi kapcsolásban



$$Y = AC + B\bar{C}$$

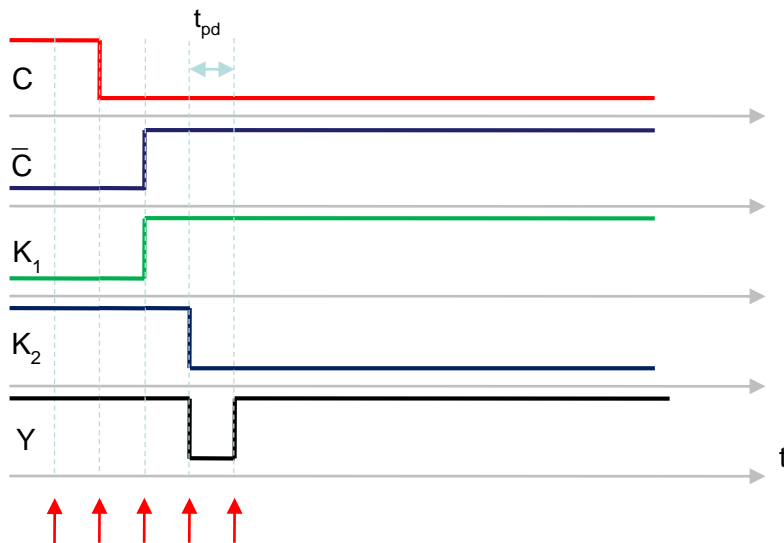
A = B = 1 állandó, C hirtelen 1-ről 0-ra változik.

Minden kapu késleltetése azonosan legyen t_{pd}

C változása az inverter kimenetén és a K1 kimeneten csak t_{pd} elteltével jelenik meg
Mivel K1 megváltozott az Y kimenet t_{pd} elteltével szintén megváltozik
Ugyan így K2 kimenet is megváltozik az inverter kimenetének változása miatt
Mivel K2 megváltozott az Y kimenet t_{pd} elteltével újból 1-esbe vált
A bemeneti C jel változása a jelkésleltetések miatt átmenetileg megváltoztatta a kimeneti jelet
A logikai függvényből ez nem kellene hogy bekövetkezzen.

$$Y = AC + B\bar{C}$$

Ezért a jelalakok az alábbiak lesznek:



Következzenek a hazard fajtái.

4.3.1.1. Statikus hazard

Statikus hazardnak nevezzük, amikor egy kombinációs hálózat bemenetének változására a kimenetnek változatlanoknak kellene maradnia, mégis megjelenik egy nem kívánt impulzus.

A legegyszerűbb kétszintű kombinációs hálózatokban is előfordulhatnak. Szomszédos bemeneti változás hatására következhet be (az előző és az új bemeneti kombinációk Hamming-távolsága 1), abban az esetben, ha ezt az átmenetet egyetlen implikáns sem fedi le. Ha kezdő és végállapotban is magas szint van, de az átmenet nincs lefedve statikus-1-hazardról beszélhetünk. Ilyenkor nullimpulzus jelenhet meg a kimeneten. Statikus-0-hazard esetén a kimeneten állandó 0-nak kellene lennie és egy impulzus (tüske) jelenik meg.

tehát

- ❖ A logikai működés alapján a kimeneti jelnek a bemenet változásakor nem szabadna változnia, átmenetileg, rövid időre mégis megváltozik
- ❖ a kimeneten „0” vagy „1” impulzus nem a logikai feltétel hatására keletkezik

4.3.1.2. Dinamikus hazárd

Csak kettőnél több szintű hálózatban fordulhat elő.

4.3.1.3. Funkcionális hazárd

Akkor jelentkezik, ha több bemeneti változó változik egyszerre (Hamming-távolság>1). Aszinkron sorrendi hálózatokban ez nem kívánt állapotátmeneteket okozhat.

4.3.1.4. Lényeges hazárd

Csak aszinkron sorrendi hálózatokban jelentkezik. Az első integrált áramkörök megjelenésekor találtak először a jelenséggel, mert itt a szekunder változó és a bemeneti változó sebessége összemérhető (versenyhelyzet). Csak akkor fordulhat elő, ha az állapottábla Unger-konfigurációt tartalmaz. Kiküszöböléséhez szándékos késleltetést kell alkalmaznunk vagy módosítanunk kell az állapottáblát. Ez általában az állapotszám növekedését jelenti.

4.3.1.5. Rendszerhazárd

Rendszerhazárról akkor beszélhetünk, ha több szekunder változónk van. A jelenség oka, hogy a visszacsatoló ágban lévő flip-flopok bemenetén más-más időkéleltetések vannak. Előfordulhat, hogy az egyik flip-flop jóval gyorsabb a többinél és már visszahat, így a többi flip-flop már nem az eredeti állapotból vesz mintát, hanem ebből a módosítottból.

4.3.2. A hazard kezelési technikái

A hazardmentesítés többlet kapuk beépítését jelenti, tehát többletköltséggel jár. Minden feladatnál elemezni kell, hogy lehet-e a legolcsóbb, legegyszerűbb – esetleg hazardos – megoldásokat alkalmazni vagy hazardmentes működésre van szükség. A statikus hazard akkor lép fel, ha szomszédos mintermek nincsenek összevonva egymással.

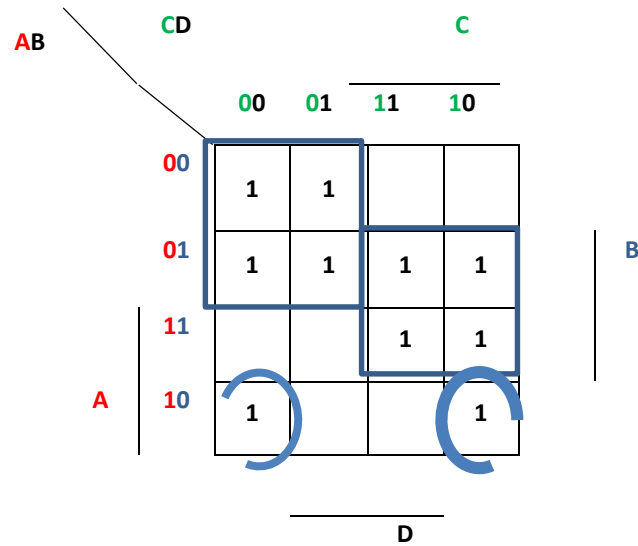
4.3.2.1. Első példa

Az alábbiakban megadtunk egy függvényt. Olvassuk ki először legegyszerűbb alakjában, vizsgáljuk meg a hazard veszélyt, majd olvassuk ki a legegyszerűbb hazardmentes függvény alakot is!

		CD		C		
		00	01	11	10	
AB	00	1	1			
	01	1	1	1	1	
	11			1	1	
	10	1			1	
A		D				B

Megoldás:

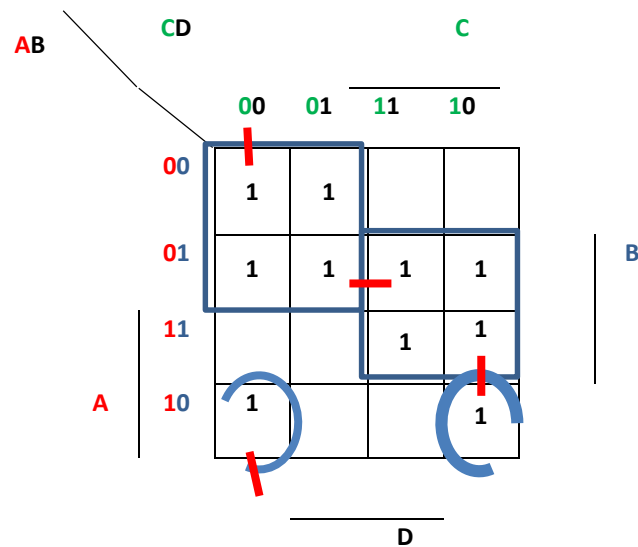
A legegyszerűbb alakhoz a tömböket úgy alakítjuk ki, hogy a lehető legkevesebb tömbbel minden 1-t letakarunk a minterm táblán:



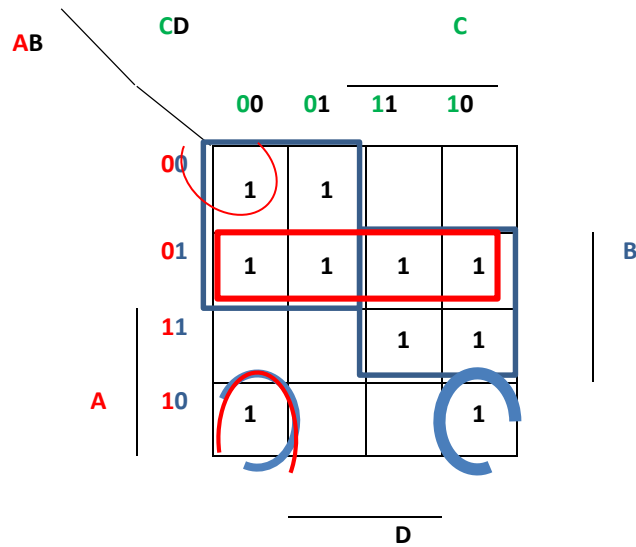
A legegyszerűbb minterm alak kiolvasva:

$$Q = \overline{A}\overline{C} + BC + A\overline{B}\overline{D}$$

Vizsgáljuk meg ezt a legegyszerűbb alakot, hogy tartalmaz-e hazard veszélyt:



Az előző ábrán szemléltettük, hogy többszörösen hazárdos ez a függvény alak. A hazárd mentesítéshez a szomszédos, **de össze nem vont mintermeket újabb tömbökkel le kell fedni:**

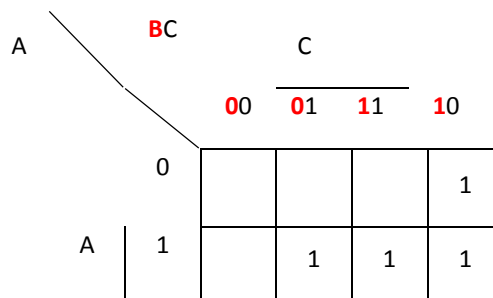


A hazárdmentes függvényalak pedig a következő:

$$Q = \overline{A}\overline{C} + BC + \overline{A}\overline{B}\overline{D} + A\overline{C}\overline{D} + \overline{A}\overline{B} + \overline{B}\overline{C}\overline{D}$$

4.3.2.2. Második példa

A legyen következő példa

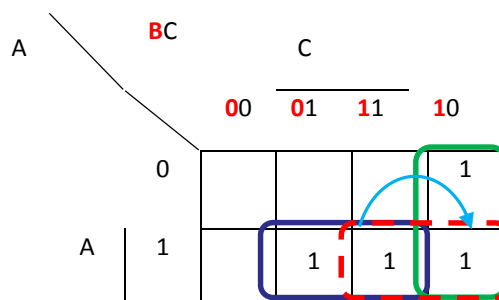


A kritikus átmenet akkor keletkezik, ha a bemeneteken $(A=1, B=1, C=1) \rightarrow (A=1, B=1, C=0)$ változás van

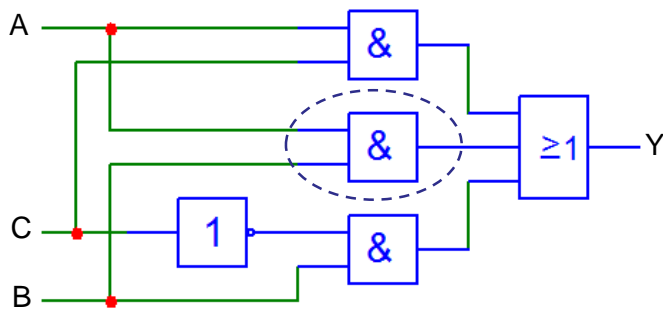
Ha ezt az átmenetet is lefedjük egy hurokkal a hazárd megszüntethető

$Y = AC + B\bar{C}$ helyett

$$Y = AC + B\bar{C} + AB$$



Ez, ha felrajzoljuk a hálózatot, az alábbi módon néz ki:



4.3.2.3. Harmadik példa

A harmadik példa pedig legyen a már tanult 2 kimenetű rendszerünk.

Kettő vagy több kimenettel rendelkező kombinációs hálózatok minimalizálásánál két külön táblába írjuk fel a rendszert. A kimeneti függvényeket külön-külön egyszerűsítjük, független de közös bemenetekkel rendelkező alhálózatokra bontva a rendszert. A függvényeket külön-külön írjuk fel, de az esetleges közös elemeket csak egyszer valósítjuk meg.

A rendszerünkről tudjuk, hogy a következő módon írható fel:

$$P = \sum^4 (13,15)$$

$$Q = \sum^4 (4,5,10,11,12,13,15)$$

Ezért két táblát tudunk felírni: két kimenetünk van.

P- re felírva:

$$P = \sum^4 (13,15)$$

		CD			
				C	
		00	01	11	10
AB	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
A	10	8	9	11	10
		D			
					B

AB		CD		C		
		00	01	11	10	
A	00	$\overline{A}BCD$	$\overline{A}\overline{B}CD$	$\overline{A}BC\overline{D}$	$\overline{A}\overline{B}C\overline{D}$	B
	01	$\overline{A}BC\overline{D}$	$\overline{A}\overline{B}C\overline{D}$	$\overline{A}BCD$	$\overline{A}\overline{B}CD$	
	11	$ABC\overline{D}$	$ABC\overline{D}$	$ABCD$	$ABC\overline{D}$	
	10	$\overline{A}BCD$	$\overline{A}\overline{B}CD$	$\overline{A}BC\overline{D}$	$\overline{A}\overline{B}C\overline{D}$	
				D		

AB		CD		C		
		00	01	11	10	
A	00					B
	01					
	11		1	1		
	10					
				D		

Ez természetesen egyszerűsíthető:

AB		CD		C		
		00	01	11	10	
A	00	$\overline{A}BCD$	$\overline{A}\overline{B}CD$	$\overline{A}BC\overline{D}$	$\overline{A}\overline{B}C\overline{D}$	B
	01	$\overline{A}BC\overline{D}$	$\overline{A}\overline{B}C\overline{D}$	$\overline{A}BCD$	$\overline{A}\overline{B}CD$	
	11	$ABC\overline{D}$	$ABC\overline{D}$	$ABCD$	$ABC\overline{D}$	
	10	$\overline{A}BCD$	$\overline{A}\overline{B}CD$	$\overline{A}BC\overline{D}$	$\overline{A}\overline{B}C\overline{D}$	
				D		

AB		CD		C		
		00	01	11	10	
A	00					B
	01					
	11		1 1			
	10					
				D		

Ezért

$$P = ABD$$

Q-ra felírva

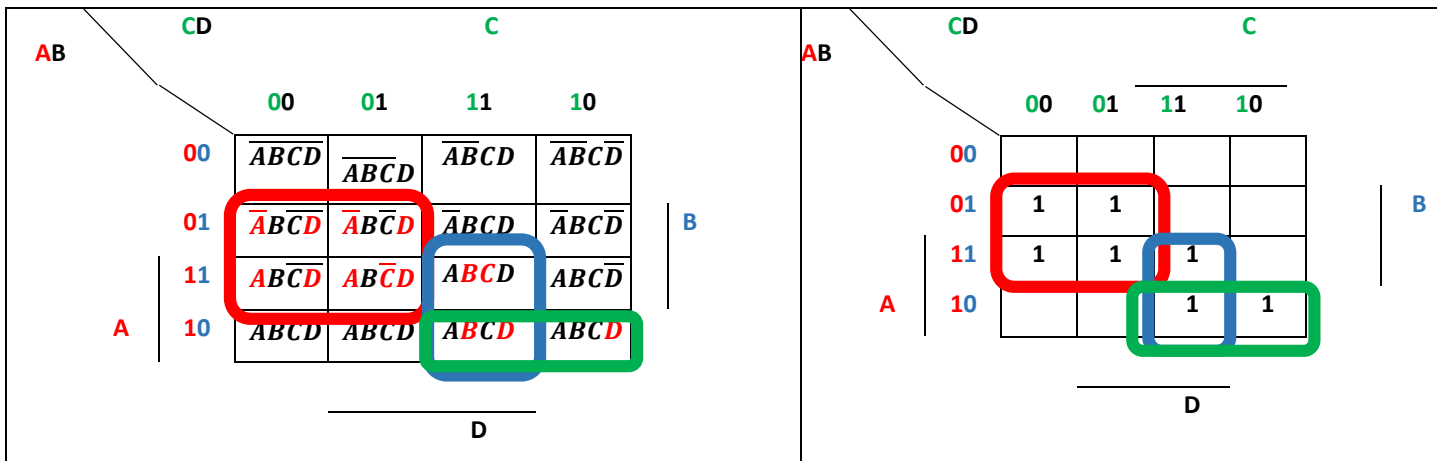
$$Q = \sum^4 (4,5,10,11,12,13,15)$$

		CD		C	
		00	01	11	10
AB	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10
		D			

AB		CD		C	
		00	01	11	10
A	00	$\overline{A}BC\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}BCD$	$\overline{A}\overline{B}C\overline{D}$
	01	$\overline{A}BCD$	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}BCD$	$\overline{A}\overline{B}C\overline{D}$
	11	$ABC\overline{D}$	$AB\overline{C}D$	$ABCD$	$ABC\overline{D}$
	10	$ABC\overline{D}$	$AB\overline{C}D$	$\overline{A}BCD$	$\overline{A}\overline{B}C\overline{D}$
		D			

AB		CD		C	
		00	01	11	10
A	00				
	01	1	1		
	11	1	1	1	
	10			1	1
		D			

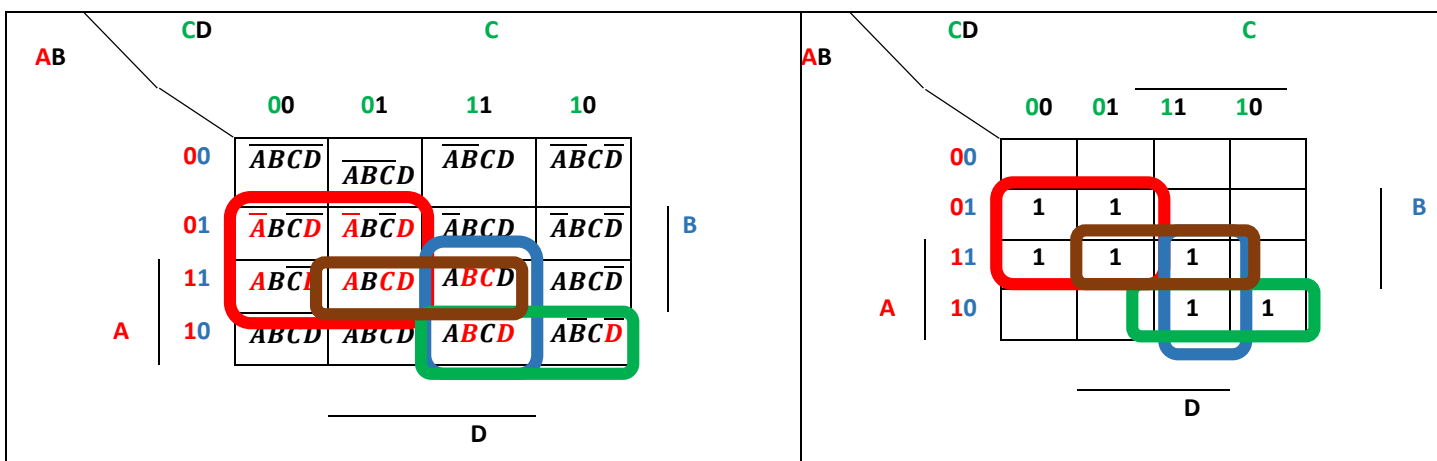
Természetesen ez is egyszerűsíthető:



Ezért az egyszerűsítés után az egyenlet a következő módon írható fel:

$$Q = \bar{B}\bar{C} + ACD + \bar{A}BC + ABD$$

A két rendszert egy táblába érdemes foglalni, hogy az azonos részeket csak egyszer valósítsuk meg.



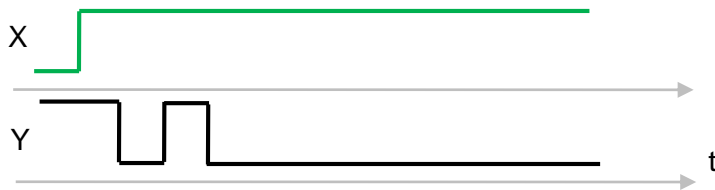
$$Q = \bar{B}\bar{C} + ABD + \bar{A}BC + ACD$$

Statikus hazard megszüntetése tehát úgy valósítható meg, hogy

- ❖ Meg kell akadályozni a kritikus átmenet hatását
- ❖ Bármely két szomszédos mintermhez találni kell legalább egy olyan hurkot, amely mindkét mintermet lefedí
- ❖ Példa: (az előző többkimenetű hálózategyszerűsítés)

Dinamikus hazardesetében pedig

- ❖ A bemeneti jel változásakor a kimeneti jelnek is változnia kell.
- ❖ De a vártól eltérően nem csak egyszer, hanem többször is megváltozik
- ❖ A jelenség három vagy többszintű hálózatokban léphet fel, és csak akkor, ha valamelyik szinten statikus hazard van
- ❖ Az egyes szinteken fellépő statikus hazardok kiküszöbölésével a dinamikus hazard is megszüntethető



33. ábra A dinamikus hazard

4.4. Ellenőrző kérdések

- 1) Definiálja a hálózatokra vonatkozó ideális helyzet fogalmát!
- 2) Definiálja a hálózatokra vonatkozó valódi helyzet fogalmát!
- 3) Sorolja fel a legfontosabb hálózati építőelemeket!
- 4) Jellemezze a kódolókat! Említsen rá példát!
- 5) Jellemezze a dekódolókat! Említsen rá példát!
- 6) Milyen típusait ismeri az adatút választóknak?
- 7) Jellemezze a multiplexereket! Említsen rá példát!
- 8) Milyen feladatokat tud ellátni a multiplexer? Említsen rá példát!
- 9) Jellemezze a demultiplexereket! Említsen rá példát!
- 10) Milyen feladatokat tud ellátni a demultiplexer? Említsen rá példát!
- 11) Sorolja fel az aritmetikai egységeket!
- 12) Jellemezze a digitális komparátort!
- 13) Mit tud az összeadóról? (Válaszát példával illusztrálja!)
- 14) Hogyan működik a kivonó? (Válaszát példával illusztrálja!)
- 15) Melyek a kombinációs hálózatok nem linearitásának okai?
- 16) Mi a hazard fogalma?
- 17) A hazard mely típusait ismeri?
- 18) Mit tud a statikus hazardról?
- 19) Mit tud a dinamikus hazardról?
- 20) Mit tud a funkcionális hazardról?
- 21) Mit tud a lényeges hazardról?
- 22) Mit tud a rendszerhazardról?
- 23) Hogyan lehet a hazard jelenségét megszüntetni? ? (Válaszát példával illusztrálja!)

4.5. Feladatok

- 1) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (4,5,13)$$

- 1) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (4,5,7,13,)$$

- 2) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (4,5,7,8,10,13)$$

- 3) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (4,5,7,8,10,11,14)$$

- 4) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (0,1,5,12,14)$$

- 5) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (0,1,5,10,12,14)$$

- 6) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (0,5,7,9,10,11,12,14)$$

- 7) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (0,9,10,11,12)$$

- 8) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (9,10,11)$$

- 9) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (0,2,3,6,7,8,10)$$

- 10) Írja fel a következő hálózatot blokkal, Karnaugh – táblával, igazságtáblázattal, logikai függvénnyel és a hozzá tartozó kapcsolási rajzzal! Ahol lehet, egyszerűsítsen!

$$F = \sum^4 (0,2,3,7,8)$$

4.6. Irodalom

Kóré László: Digitális elektronika I. (BMF 1121)

Zsom Gyula: Digitális technika I. (Műszaki Könyvkiadó, Budapest, 2000, KVK 49-273/I, ISBN 963 6 1786 6)

Zsom Gyula: Digitális technika II. (Műszaki Könyvkiadó, Budapest, 2000, KVK 49-273/II, ISBN 963 16 1787 4)

Arató Péter: Logikai rendszerek tervezése (Tankönyvkiadó, Budapest, 1990, Műegyetemi Kiadó 2004, 55013)

Zalotay Péter: Digitális technika (<http://www.kobakbt.hu/jegyzet/DigitHW.pdf>)

Rómer Mária: Digitális rendszerek áramkörei (Műszaki Könyvkiadó, Budapest, 1989, KVK 49-223)

Rómer Mária: Digitális technika példatár (KKMF 1105, Budapest 1999)

Matijevics István: Digitális Technika Interaktív példatár (ISBN 978-963-279-528-7 Szegedi Tudományegyetem)

http://www.inf.u-szeged.hu/projectdirs/digipeldatar/digitalis_peldatar.html

Pszota József: Digitális áramkörök

http://centroszet.hu/tananyag/digit_aramkorok/134_demultiplexerek_gyakorlati_alkalmazsa.html