

Zsom Gyula

Digitális technika I.

7. KIADÁS

**A BUDAPESTI MŰSZAKI FŐISKOLA
KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI FŐISKOLAI KAR
MEGBÍZÁSÁBÓL KIADJA A MŰSZAKI KÖNYVKIADÓ,
BUDAPEST**

Lektorok:
DR. KOHUT JÓZSEF
SAUFERT JÁNOS

Sorozatszerkesztő:
ZSOM GYULA

© ZSOM GYULA
© HUNGARIAN EDITION MŰSZAKI KÖNYVKIADÓ, 2000

ISBN 963 16 1786 6

Kiadja a Műszaki Könyvkiadó Kft.
Felelős kiadó: Bérczi Sándor ügyvezető igazgató

Nyomta és kötötte a Borsodi Nyomda
Felelős vezető: Ducsai György ügyvezető igazgató

Műszaki vezető: Trencsényi Ágnes
A könyv formátuma: B5. Terjedelme: 34 (A5) ív
Ábrák száma: 300. Azonossági száma: 49 273/I
Készült az MSZ 5601:1983 és 5602:1983 szerint

TARTALOMJEGYZÉK

	Oldal
1. ALAPFOGALMAK, KÓDOK	7
1.1. A digitális jellel kapcsolatos alapfogalmak ...	7
1.1.1. A digitális jel "természete"	7
1.1.2. Kód, kódolás	9
1.2. A kétértékűen kódolt jel megjelenítési, továbbítási formái villamos rendszerekben	12
1.3. Numerikus kódok (szám-kódok)	19
1.3.1. Kettes számrendszer	19
1.3.2. BCD (Binary-Coded-Decimal) kódok	33
1.3.3. Hexadecimális kód	36
1.3.4. "Egylépéses" kódok	37
1.4. Alfa-numerikus kódok (gyakori betű-szám kódok) ..	42
1.4.1. Nemzetközi táviró-kód (BAUDOT)	42
1.4.2. ASCII (American Standard Code for Information Interchange)	44
1.4.3. Egyéb bináris alfa-numerikus kódok	45
1.5. Hibajelzés és hibajavítás	46
2. KOMBINÁCIÓS LOGIKAI HÁLÓZATOK	50
2.1. A logikai hálózatok modellje	50
2.2. Logikai alapműveletek	56
2.2.1. Negáció, tagadás, invertálás	56
2.2.2. A logikai VAGY kapcsolat, diszjunkció ..	58
2.2.3. A logikai ÉS művelet, konjunkció	60
2.2.4. A BOOLE-algebra alaptételei, szabályai ..	64
2.2.5. Az összes lehetséges kétváltozós logikai kapcsolat	70
2.2.6. Többváltozós logikai kapcsolatok	81

	Oldal
2.2.7. Univerzális műveletek: logikai alap- műveletek megvalósítása univerzális építőelemekkel	81
2.3. Logikai függvények	90
2.3.1. Logikai függvények megadása	90
2.3.2. Logikai függvények felírása adott feladat alapján	95
2.3.3. Logikai függvények normál alakjai, mintermek, maxtermek	101
2.4. Minimálási módszerek	108
2.4.1. A minimális kombinációs hálózat	108
2.4.2. Az algebrai egyszerűsítés	112
2.4.3. A grafikus egyszerűsítés	115
2.4.4. Táblázatos egyszerűsítés: a QUINE- McCLUSKEY-módszer	134
2.5. Kombinációs hálózatok megvalósítása univerzális művelési elemekkel	137
2.5.1. NAND hálózat tervezése KARNAUGH-táblával.	137
2.5.2. NOR (NOT-OR vagy OR-INVERT) hálózat tervezése KARNAUGH-táblával	140
2.5.3. AND-OR-INVERT (ÉS-NEM-VAGY) hálózat tervezése	142
3. A LOGIKAI HÁLÓZATOK ÉPÍTŐELEMELI, LOGIKAI ÁRAMKÖR CSALÁDOK	146
3.1. Elektromechanikus logikák	146
3.2. Diódás logikai áramkörök, diódák kapcsoló üzeme.	152
3.2.1. A diódás VAGY kapu (DDL OR-GATE)	154
3.2.2. A diódás ÉS kapu (DDL AND-GATE)	156
3.3. A bipoláris tranzisztorok kapcsolóüzeme, inverter	160
3.4. A térvezérlésű eszközök kapcsolóüzeme, MOS inverter	172
3.5. Bipoláris áramkör családk	180
3.5.1. A digitális áramkörök legfontosabb jellemzői	181

	Oldal
3.5.2. Régebbi bipoláris áramkörök	186
3.5.3. TTL, T ² L (Transistor-Transistor-Logic= =tranzisztor-tranzisztor logika)	189
3.5.4. ECL, Emitter Coupled Logic, emitter- csatolt logika	234
3.5.5. Újabb bipoláris áramkörök: I ² L (Integrated Injection Logic: integrált injekciós logika)	238
3.6. MOS és CMOS áramkörök	242
3.6.1. MOS áramkörök	242
3.6.2. Komplementer MOS (CMOS, COS-MOS, McMOS....) áramkörök	245
3.7. Az áramkör családk összehasonlítása	254
3.8. Különböző áramkör családk illesztése	258
3.8.1. TTL-CMOS illesztés	258
3.8.2. CMOS illesztése TTL-hez	261
3.8.3. TTL-MOS illesztés	263
3.8.4. MOS-TTL illesztés	264
3.8.5. TTL-ECL illesztők	264
3.8.6. ECL-TTL illesztők	265
4. TIPIKUS KOMBINÁCIÓS HÁLÓZATOK ÉS MEGVALÓSÍTÁSI LEHETŐSÉGEK SSI-MSI-LSI ELEMekkel	267
4.1. SSI áramkörök	267
4.2. MSI kombinációs áramkörök	270
4.2.1. Dekódolók, kódolók (kódváltók)	271
4.2.2. Demultiplexerek, multiplexerek	281
4.2.3. Aritmetikai elemek	297
4.3. Kombinációs hálózatok megvalósítása LSI-vel ...	306
5. A SORRENDI HÁLÓZATOK ALAPELEMELI	318
5.1. A flip-flop-ok modellje	318
5.2. Bistabil flip-flopok	319
5.2.1. Az RS flip-flopok	320
5.2.2. A JK és a T flip-flop	331
5.2.3. A D flip-flop	342

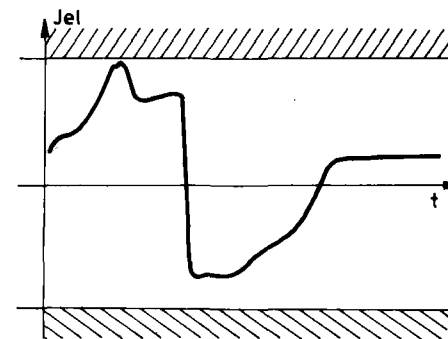
	Oldal
5.3. Monostabil flip-flopok	346
5.3.1. Működési elv	346
5.3.2. Monostabil áramkörök	349
5.4. Astabil kapcsolások, digitális oszcillátorok ...	363
5.4.1. Működési elv	363
5.4.2. Astabil kapcsolások felépítése mono- stabil elemek összekapcsolásával	364
5.4.3. Egyetlen időzítő tagot igénylő kapcsolások	366
5.4.4. Pontos oszcillátorok	374

1. ALAPFOGALMAK, KÓDOK

1.1. A digitális jellel kapcsolatos alapfogalmak

1.1.1. A digitális jel "természete"

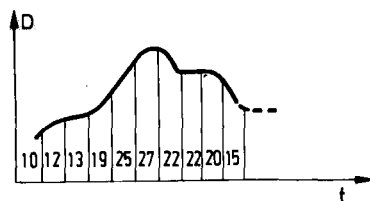
Az "analóg jel", amelyet többnyire lineáris egységek dolgoznak fel, állítanak elő, tudatunkban úgy él, mint egy időben folyamatosan változó jel, amely egy adott tartományt tet-szés szerinti "közbenő" értékekkel teljes mértékben kitölthet és általában folytonos időfüggvénnyel ábrázolható (1.1. ábra).



1.1. ábra

Az analóg rendszerekben legfontosabb jellemzők pl. az egyen-szint-átvitel, jelátviteli tényező, "frekvenciamenet" (sávszé-lesség), jel-zaj viszony, torzítás stb. Ez azt jelzi, hogy az analóg jel feldolgozása közben alapvetően fontos az eredeti információ torzítatlan megőrzése, alakhü rögzítése, továbbí-tása, stb., ill. előre elhatározott céllal valamely jellemző-jének módosítása (amplitudó, fázis, frekvencia sávhatárok..). Vegyünk példát a mindennapi életben megszokott analóg jelek közül: a "hang"-jeleket elektroakusztikai eszközökkel alakít-ják át villamos jellé, rögzítik, továbbítják, reprodukálják.

az akusztikai jelnek megfelelő információ villamos analóg megfelelője a "hangfrekvenciás" jel (a továbbiakban villamos rendszerekkel foglalkozunk csak, "jel" alatt villamos jelet értünk). A hangfrekvenciás jel főbb jellemzői: a 20 Hz...20 kHz frekvenciasáv, a jónak mondható 50...80 dB jel-zaj viszony és pl. a jó értéknek számító 0,1% alatti harmonikus torzítás mind olyan adatok, amelyek analóg jelre értelmezhetők, a jel "analóg természetére" utalnak.



1.2. ábra

A digitális jel (szó szerint: számjegyes jel) nevéből következően csak számokkal dolgozik, vagyis csak "diszkrét" "kvantált" értékei vannak (bármilyen "közel" is legyenek ezek az értékek egymáshoz) a digitális jel az információt elemi részekre osztva képes kifejezni, kódolni (például a hangfrekvenciás jel digitális megfelelőjének előállításához adott időnként a jelből "mintát kell venni" és ezen minták amplitudó értékéhez számokat kell rendelni az 1.2. ábrán szemléltetett módon. Már most meg kell jegyeznünk, hogy ez csak egyike a lehetséges digitalizálási lehetőségeknek, elképzelhető, hogy a jel más jellemzőihez rendelünk számokat. Minderről későbbi tárgyakban részletesen szó lesz.). A digitális jel tehát kódolt információt tartalmaz, a digitális berendezések ezt a kódolt jelet dolgozzák fel. Célszerű ezért a kódolással, kódokkal kapcsolatos kifejezéseket, definíciókat röviden összefoglalni, értelmezésükben megállapodni.

1.1.2. Kód, kódolás

A kód általános értelmezésben: valamely szimbólum, halmazhoz egyértelműen rendelt, azt egyértelműen leképező szimbólum-halmaz. Más szóval és "hétköznapi" értelmezésben a kód valamely információ kifejezésére, hordozására szolgáló rendszer, amely az információ elemeihez egyértelműen kód-jeleket rendel.

A kódolás maga az a művelet, amellyel az adott információ elemeihez a kód-jeleket, szimbólumokat egyértelműen hozzárendeljük. Az "adott információ" is lehet, hogy már kódolt (legtöbbször így van), ilyenkor a kódolás az egyik kódról a másikra való áttérést jelenti.

A dekódolás a kódolás inverz művelete, amelynek során a kódról visszatérünk az eredeti információra (ill. annak "eredeti" kódjára).

A jelkészlet, szimbólum készlet azoknak az elemi jeleknek az összessége, amelyeket kódolásra felhasználhatunk (pl. az emberi beszédben - amely gondolatok kódjának mondható - a hangok, ill. az ezeknek megfelelő írásjelek összessége). A digitális technikában kétértékű kódokkal dolgozunk, a szimbólum készlet a "0" és az "1", amelynek igen sokféle fizikai szimbólum felelhet meg: egy feszültség vagy áram "magasabb" High = H szintje (1), ill. "alacsonyabb" Low = L szintje (0), de lehet optikai "szimbólum" is: pl. egy lyukszalag vagy átengedi a fényt egy adott helyen (1), vagy nem (0), egy hordozó vagy visszaveri a fényt (0), vagy nem (1), mint ahogyan ez például a lézeres adatrögzítőkben van (digitális hang és kép-lemez, lézeres memória), a 0-nak és 1-nek megfelelően egy mágnesszalag vagy lemez egyik irányban, vagy másik irányban történő felmágnesezése, egy érintkező esetén szakadás vagy összekötés stb. Az is lehet, hogy a 0-nak és 1-nek az előbb említett (és még sok egyéb) fizikai mennyiségek statikus szintje helyett szint-változása ("ugrása") esetleg az ugrás fázisa, stb. felel meg (lásd később).

A kódszó (röviden szó = word) a kód szimbólumaiból (0-1-ekből) alkotott egybefüggő sorozat. A kódszavak egy-egy alkotó elemét, szimbólumát a kétértékű (bináris) rendszerben bit

nek nevezzük. A különböző rendszerekben, berendezésekben a kódszavak különböző bit-számúak lehetnek, egy-egy 8 bites egységet byte-nak szokás nevezni és a kódszavak hosszát nagyon gyakran byte-ban kifejezve adják meg (egy byte-os szó: 8 bites, 2 byte-os 16 bites, fél byte-os: 4 bites stb.). A munka megkönnyítése érdekében a kódszavak bizonyos adott bit-számú "darabjait" egy-egy szimbólummal, rövidítéssel helyettesítjük (pl. 4 bitet decimális számokkal, ill. az ábécé első betűivel), ami nem változtat a kódok kétértékűségén.

A kódszó készlet azoknak a kódszavaknak az összessége, amelyek az adott rendszerben kódolásra felhasználhatók. Lehetnek olyan szavak, kódkombinációk is, amelyek formailag beilleszkednek az adott kódrendszerbe, de nem tartoznak a kódszó készletbe, ezeket tiltott kódszavaknak nevezzük.

A kódolt információ jellege szerint a kódok fő csoportjai:

- a numerikus kódok, amelyek számok kifejezésére alkalmasak (legfontosabb a bináris szám-kód, ide tartozik a BCD, Gray stb. kód, lásd később);

- alfanumerikus kódok számok és betűk, valamint írásjelek kódolására alkalmasak (TELEX, ASCII stb.).

Az analóg és a digitális jel összehasonlítása érdekében térjünk vissza hangfrekvenciás példánkhoz. Végezzünk nagyságrendi becslést, amellyel megállapíthatjuk az 1.2. ábra szerinti elven előállított "digitális hangjel" alapjellemezőit. A hangfrekvenciás sáv felső határa 20 kHz, ezért minimálisan 40 kHz-es mintavételezési frekvencia szükséges (a mintavételezés alapelveiről később lesz szó), legyen a mintavételezési frekvencia a "biztonság kedvéért" 44 kHz (44,056 kHz a mai digitális hangrögzítők "szabványos" értéke). Egy-egy minta amplitudóját kifejező bináris 2-es számrendszerbeli számnak, minimálisan 14 "jegyűnek" kell lennie azért, hogy az amplitudó értékeket kellő pontossággal reprodukálni tudjuk. Vegyük figyelembe, hogy sztereo hang esetében két csatorna jelét kell digitalizálni. Így a minimálisan szükséges bit-frekvencia (másodpercenkénti bit szám, szokásos elnevezéssel BAUD-RATE):

$$44,056 \cdot 10^3 \cdot 2 \cdot 14 = 1\,233\,568 \text{ bit/s} = 1,233568 \text{ Mbit/s}(!).$$

Azért, hogy a rögzítéskor, átvitelkor és vételkor az óhatatlanul létrejövő hibákat a digitális rendszer automatikusan kiküszöbölhesse, "többlet" biteket ("ellenőrző-javító" biteket) kell a hasznos információt hordozó bitekhez "hozzákeverni", ami azt jelenti, hogy a szükséges bit sűrűség:

$$2 \dots 2,6 \text{ Mbit/s!}$$

Azonnal szembetűnik a lényeges különbség: a "felső határfrekvencia" több mint 2 MHz-re (100-szorosra) növekedett! Az is igaz, hogy ezzel a frekvenciával csak kétféle, 0 és 1 jelszintet kell továbbítani, rögzíteni, "lejátszani". A digitális jel vivő (rögzítő stb.) "csatorna" sokkal "zajosabb" lehet, hiszen csak a 0-nak és az 1-nek megfelelő szintet kell biztonsággal elkülöníteni (az esetlegesen létrejövő hibákat a "többlet" bitek segítségével "visszajátszáskor" korrigálni is lehet). A digitális rögzítésnek köszönhetően a visszaalakított hangfrekvenciás jel (pl. egy digitális, lézeres, ún. Compact Disc, CD lemezjátszó kimenetén) a következő jellemzőkkel áll elő:

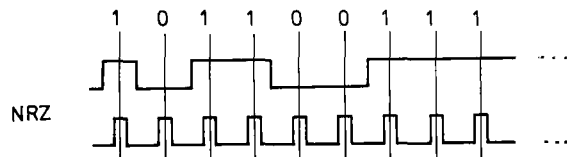
Frekvencia átviteli sáv:	20 Hz...20 kHz (-1 dB)
Jel-zaj viszony:	92 dB (!)
Harmonikus torzítás:	0,05% (!)
Sztereo csatorna-eltávolítás:	90 dB-nél jobb (!)
Hangmagasság-ingadozás:	(nem mérhetően kicsi!).

Összefoglalva: ahogy a példán is láttuk, az analóg jelek digitális feldolgozása igen nagy bit-sűrűséget igényel (az analóg jel felső határfrekvenciájához képest), de a feldolgozás nagyon biztonságos lehet: az egyszer már digitálisan előállított jel átmásolható, tárolható, továbbítható, gyakorlatilag mindenfajta minőség romlás nélkül, nem úgy, mint az analóg technikában! Ne felejtsük el azt a magától értetődő előnyt sem, hogy a digitális jel számítástechnikai eszközökkel kezelhető, feldolgozható.

1.2. A kétértékűen kódolt jel megjelenítési, továbbítási formái villamos rendszerekben

Említettük már, hogy a "0" és "1" jelet sokféle fizikai mennyiség kétféle állapota vagy állapotváltozása jelképezheti. A továbbiakban, mivel digitális áramkörökkel foglalkozunk, a "0" és az "1" villamos mennyiségekkel való megjelenítési formáit kell elsősorban megismernünk. Villamos rendszerben a digitális jelet feszültség vagy áram kétféle értéke jelképezheti, ha külön nem említjük, akkor feszültségről lesz szó a továbbiakban. A kétféle feszültség az L ("alacsony") és a H ("magas") szint. A következőkben nézzük meg, hogy egy digitális jelet vivő vezetékben az egymás után továbbított 0-1 biteknek milyen jelképeket feleltetnek meg leggyakrabban (a feszültség-idő függvényeket a szokásnak megfelelően leegyszerűsítve rajzoljuk meg: a tengelyvonalakat elhagyjuk és mivel a feszültségek kétféle lehetséges szintűek lehetnek, az amplitudónak, léptéknek nincs jelentősége, csak - több összetartozó jel esetén - az egymáshoz viszonyított helyzetnek, időzítésnek). A következőkben az 1.3...1.6. ábrákon néhány gyakori modulálatlan jelképet mutatunk be.

NRZ (Non Return to Zero: nullára nem visszatérő, 1.3. ábra).

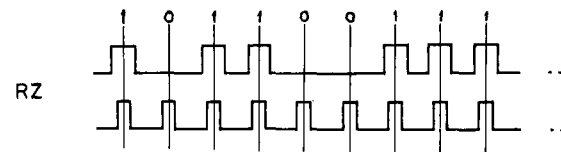


1.3. ábra

Ez a leginkább gyakori, "természetes" jelképe. Ha egy bit 1-es, akkor a feszültség teljes bit idő alatt H szintű, ha 0-ás, akkor L szintű. Két vagy több egymás utáni 1-es bit esetén a feszültség megszakítás nélkül H-ban marad a megfelelő ideig, az egyesek között nem tér vissza 0-ra. Azért, hogy tudjuk, mikor melyik bit jele van éppen a vezetéken, szinkronizáló órajelnek (CLOCK) is kell lennie, amely szintén logi-

kai szintű négyyszögjel, de általában "keskenyebb", kisebb kitöltési tényezőjű, és például a bit idő "közepén" érkezik. Általában úgy időzítik, hogy amikor a CLOCK jel 1-ben van, akkor a bit vezetéken levő jel "érvényes".

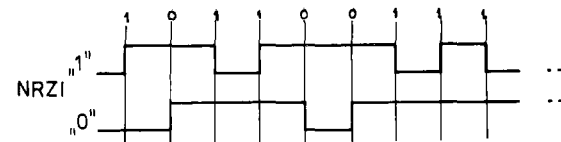
RZ (Return to Zero: nullára visszatérő, 1.4. ábra).



1.4. ábra

A nulla a "nyugalmi állapot", ha két 1-es bit következik egymás után, akkor a két bit között a jel visszatér a 0-ra.

NRZI (Non Return to Zero Interrupt: nullára nem visszatérő, "megszakadásos", 1.5. ábra).

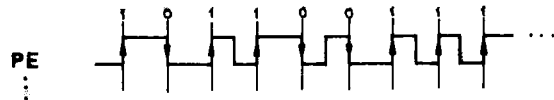


1.5. ábra

Egy példa arra, hogy a bit értékeket nem feltétlenül logikai szintek jelképezik. Ebben az esetben a jelátmenet határozza meg, hogy az illető helyen 1-es bit van vagy nem. Az átmenet iránya közömbös (mindegy, hogy 1-0 vagy 0-1 átmenet történik, lényeg, hogy "ugrás" legyen). A legtöbb esetben nem külön órajel "mellékelnek hozzá", hanem inkább egy másik, 0-ás csatornát. Így minden bit-időben történik ugrás: vagy az 1-es csatornán (amikor a bit érték 1), vagy a 0-ás csatornán (amikor a bit érték 0). Egyszerre mindkét csatornán nem lehet 1-0 vagy 0-1 átmenet (egyszerre nem lehet egy bit 1 és 0). Így módon ez a jelképe ön-órázó (self-clocking): az órajel "bújtatva" benne van a jelben. Ez a forma különösen mágnesszalagos jelrögzítésre alkalmas, mivel a "magnófejen" jel csak akkor indukálódik, amikor változás van a szalag-fluxusban, egyenszintet nem lehet rögzíteni. NRZI esetén minden bit-nél vagy

az 1-es, vagy a 0-ás sávon van "olvasott" jel (ezt a formát használják pl. a HP-67 programozható kalkulátorban a mágneskártyákon való rögzítésre-beolvasásra).

PE (Phase Encoded: fázis-kódolt jel, 1.6. ábra).

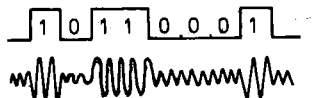


1.6. ábra

Ennél is jel-átmenet, ugrás jelképezi a biteket, de itt az ugrás irányának is jelentősége van: pl. 0-1 átmenet 1-es bitet, 1-0 átmenet 0-ás bitet jelöl. Akkor, amikor több azonos bit követi egymást, akkor a jelnek a két bit között "fél-időben" vissza kell térnie az eredeti szintre azért, hogy a következő bit idején ugyanolyan irányu átmenet következhesen be (lásd az ábrát). A jel detektálásakor, visszaállításakor, az alapfrekvenciás, bit értékeket hordozó átmeneteket el kell különíteni a kétszeres frekvenciájú "hamis" átmenetektől (a mai technikában ez nem okoz nehézséget). Mivel az információt ennél a formánál is jel-átmenetek hordozzák, kiválóan alkalmas mágneses adatrögzítéshez. További jelalakokkal a későbbiekben a II. kötetben a mágneses adatrögzítők tárgyalásakor foglalkozunk majd.

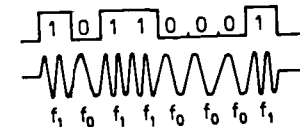
A modulált jeleket távolsági digitális jelátvitelre (rádió, hírközlésben, telefon-vonalon való továbbításakor), valamint pl. közönséges magnetofon készüléken történő digitális jelrögzítésre használjuk. A modulált jelformákban nem DC szintek vagy DC szint-megváltozások, hanem egy rendszerint szinuszos, a bit-frekvenciánál nagyobb frekvenciájú vivő valamely jellemzőjének megváltozásai jelentik a bit-értékeket. A modulációk tipikus formái:

amplitudó moduláció: amelynél a kétféle bit értékeket a vivő két, erősen különböző amplitudó értéke jelképezi (1.7. ábra);



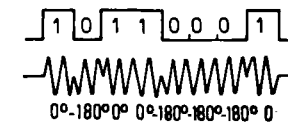
1.7. ábra

frekvencia moduláció, amelynél a két bit-értéket a vivő szinusz kétféle frekvencia értéke jelképezi (1.8. ábra). Szokásos elnevezés az FSK (Frequency Shift Key: "frekvencia billentyűzés") is. Igénytelen, nagyon elterjedt modulációs forma (pl. digitális jelrögzítésre);



1.8. ábra

fázis moduláció esetén a kétféle bit értéket a vivő kétféle fázishelyzete reprezentálja (1.9. ábra).



1.9. ábra

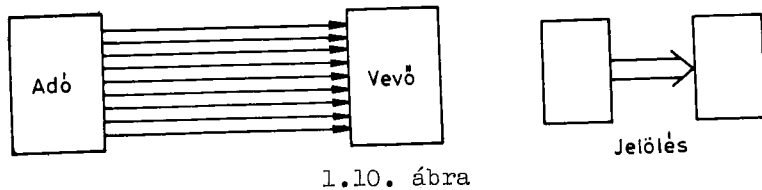
A digitális jel átvitele

Gyakori feladat az, hogy egy berendezés egyik részéből a másikba vagy az egyik berendezésből esetleg egy távol lévő másik berendezésbe kell digitális információt átvinnünk. Általánosan és leegyszerűsítve ez azt jelenti, hogy a megfelelően kódolt jeleket meghatározott sebességgel egy ADÓ-ból egy VEVŐ-be kell juttatni. Az ADÓ lehet pl. egy perifériás berendezés vagy egy digitális mérőberendezés, a vevő lehet pl. egy számítógép, vagy egy központi egység, vagy akár egy másik periféria, az is lehet, hogy egyazon berendezés két egységéről van szó - a lehetőségek száma szinte végtelen. Az átvitel a csatornán történik, amely lehet huzalpár vagy több huzalból álló kábel, rádiófrekvenciás összeköttetés, vagy akár fényvezető szál, stb. Az átvitel alapvetően kétféle lehet: párhuzamos és soros.

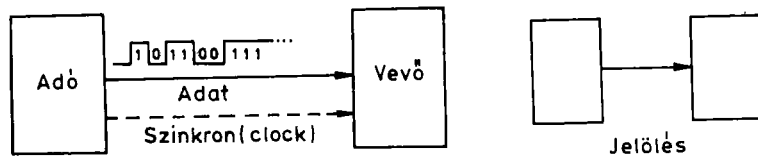
Párhuzamos átvitelkor egy-egy kódszó bitjeit egyidejűleg, külön-külön vezetéken továbbítjuk az adóból a vevőhöz, az összes bit gyakorlatilag egyidőben érkezik a vevő oldalra. A

kódszavak sorban egymás után kerülnek továbbításra (párhuzamos bit, soros szóátvitel, 1.10. ábra).

Soros (soros bit, soros szó) átvitelkor az információ biteit egyenként, egymás után továbbítjuk "egy" vezetéken (a valóságban vezeték-páron, az egyik vezeték sokszor a földvezeték, 1.11. ábra).



1.10. ábra



1.11. ábra

A soros és a párhuzamos átvitelt összehasonlítva nyilvánvaló, hogy a párhuzamos átvitel gyorsabb lehet, hiszen egyszerre több bitet közvetítünk, viszont az is látszik, hogy a párhuzamos átvitel megvalósítása sokkal drágább. Maga az "ADÓ" és a "VEVŐ" rendszertechnikailag általában ugyan egyszerűbb, de az összeköttetéshez sok huzal (sok csatorna) kell, amelyet nagy távolságra csak nagyon drágán lehet kiépíteni. Ezért ott gazdaságos, ahol az egységek közel vannak egymáshoz, akár egyetlen berendezés részei (egy berendezésen belül nyilván nem fizetődne ki a jel "sorosítása", majd visszaalakítása) vagy legalábbis "épületen belül" vannak. Gondolni kell azonban a csatlakoztatás mechanikai nehézségeire is. Egy kártyán lévő sok kivezetésű alkatrészeket viszont mindenképpen párhuzamosan, ha lehet busz-rendszerrel (lásd később) célszerű összekötni.

Szinkronizáció

Önmagától vetődik fel a kérdés, hogy a vevő honnan tudja meddig tart az egyik bit és mikor kezdődik a következő. A vevőnek a megfelelő időpillanatban kell érzékelnie a 0-át vagy 1-et, akkor is, amikor egymás után több 1-es vagy több 0 van, és a biteket nem választja el egymástól 0-1 vagy 1-0 átmenet. Szükség van tehát bit szinkronizációra ("egyidejűsítésre").

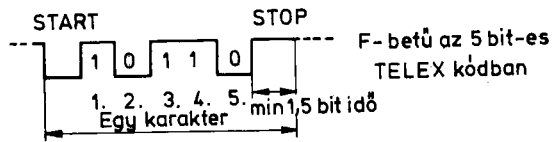
Soros átvitelkor a vevőnek azt is tudnia kell, hogy hol van a kódszavak, karakterek eleje és vége, ehhez kell a karakter-szinkronizáció. Ha több karakter blokkba rendezve kerül továbbításra, akkor el kell különíteni a blokkokat egymástól. Ez a blokk-szinkronizáció feladata. Természetesen érzékelni kell a vevő oldalon a teljes üzenet elejét és végét az üzenet-szinkronizáció segítségével. A szinkronizáció legegyszerűbb "legtisztább" módja külön szinkron vezeték létesítése az adó és vevő közt. Párhuzamos átvitelnél ez szokásos is, sorosnál akkor, ha az adó és vevő közel van, és nem jelent többletköltséget a járulékos, szinkront vivő vezeték. Legtöbbször a szinkronizálásról a vevőoldalon kell gondoskodni.

Az időzítés szempontjából az adatátvitel lehet aszinkron vagy szinkron.

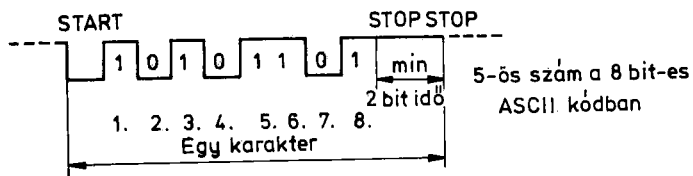
Az aszinkron átvitelt START-STOP üzeműnek is nevezik. Az információ szakaszosan, karakterenként, kódszavanként kerül továbbításra, a szavak között szünettel. Az "adás" mindig egy START bittel kezdődik és egy STOP (szünet) jellel fejeződik be, mely szünet feltétlenül hosszabb egy bit időnél, a maximuma nincs meghatározva. Ilyen aszinkron áramjeleket küld például a TELEX gép (kódja 5 bites TELEX kód, 1.12a ábra) vagy például a TELETYPE (kódja 8 bites ASCII - American Standard Code for Information Interchange - 1.12b ábra).

Amikor egy billentyűt leütünk, a gép először egy "START" impulzust ad ki, mely "0" szintű, majd a "hasznos" biteket és végül egy "1"-es STOP jelet, mely legalább 1,5, ill. 2 bit ideig tart, de tulajdonképpen addig tart, amíg a következő betű billentyűjét le nem ütjük. A START és a STOP jelek alapján így a kódszavak jól elkülöníthetők. A bit szinkron is könnyen megvalósítható, a bitek helye meghatározható a vevő oldalon,

ha a START jellel szinkronizált (indított) és az adó bitfrekvenciájával azonos frekvenciájú ÓRA-jélet állítunk elő a vevőben. A "regenerált" órajel frekvenciájának nagyon pontosnak kell lennie, mert eltérés esetén a hibák halmozódnak és az utolsó biteknél az "elcsuszás" elérhet egy teljes bit időt, akkor pedig a vett információ hamis lesz. Az elcsuszás veszélye annál nagyobb, minél hosszabb kódszavakat küldünk. Távgépirókban és más elektromechanikus elven működő perifériás készülékekben a szinkronizálás mechanikai uton - az adóban és vevőben karakterenként egyszerre induló és azonos fordulatszámmal elforduló - vezérlőművel történik. A START - STOP üzem nagy előnye, hogy a berendezések bizonyos mértékig egyszerűbbek és olcsóbbak lehetnek, mivel az információt nem kell tárolni, nem kell buffer memória. Hátrány viszont, hogy az átvívó csatorna legtöbbször nincs kihasználva a sokszor nagyon hosszú szünet idők miatt (amíg gondolkodunk vagy keressük a következő betű billentyűjét).



a)

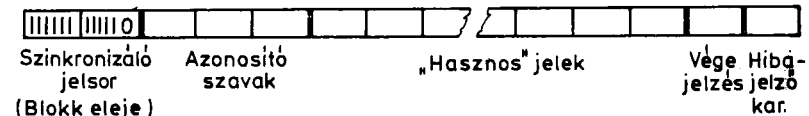


b)

1.12. ábra

A szinkron átvitel a lehető legjobb csatorna kihasználást teszi lehetővé. Itt általában nem tartanak szüneteket, a karakterek, szavak folyamatosan, egyik a másik után "áramlanak" START és STOP jel nélkül. A karakter-sorozat blokkokba van rendezve. A vevőnek az adóval pontos szinkronban kell lennie a blokk teljes hosszában, és nemcsak a bitek helyét kell "tudnia" elcsuszás mentesen, hanem azt is, hogy hol van a blokk eleje és a blokkban hol kezdődnek és végződnek az egyes kódszavak.

A blokk hossza, rendszertől függően, a legkülönbözőbb lehet, néhány karaktertől akár többszázig a továbbítandó jel fizikai természetéhez igazodva. A szinkronizáló oszcillátornak - különösen hosszú blokkok esetén -, igen pontosan fázisban kell lennie az adóval. Ezt általában a blokkok elején elhelyezett szinkronizáló jel "csomagokkal" biztosítják, amelyekkel az oszcillátort "behúzzák". Ezenkívül nagyon stabil frekvenciájú oszcillátort is kell alkalmazni (kvarc). Van olyan rendszer is, amelyben a blokkok között szünetet, "hézagot" tartanak az aszinkron átvitelhez hasonlóan (például a mágnesszalagos adatrögzítők). A szinkron átvitelhez tartozó berendezések általában drágábbak, mivel a blokkba rendezéshez járulékos elektronikára van szükség (buffer tároló). Főleg ott célszerű a szinkron adatátvitel, ahol egyetlen csatornára időosztással több berendezés kapcsolódik és ezért az időkihasználás nagyon lényeges. Egy blokk például az 1.13. ábra szerint nézhet ki.



1.13. ábra

1.3. Numerikus kódok (szám-kódok)

1.3.1. Kettes számrendszer

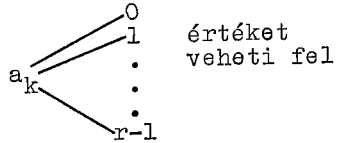
Ez a "legtriviálisabb", legtöbbször használt bináris (kétértékű) kód. Feltételezzük, hogy e jegyzet olvasója már a számrendszerekkel és ezen belül a kettes számrendszerrel tisztában van, így itt csak röviden átismételjük a legfontosabb tudnivalókat.

A kettes számrendszer is a súlyozott, helyiértékes számrendszerekhez tartozik (nem úgy, mint az additív számrendszerek, pl. a római számok, ahol nincsenek helyiértékek, a számokat az egyes jegyek megfelelő szabály szerinti összeadás-kivonásával képezzük). A kettes számrendszer alapja, radix-a a

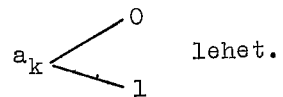
2, ennek egész számu hatványai adják a helyiértékeket. Valamely A számot tehát a következő formában szokás megadni:

$$A = \sum_k a_k r^k = \dots + a_3 r^3 + a_2 r^2 + a_1 r^1 + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \dots$$

RADIX
VESSZŐ
(TIZEDESVESSZŐ)



$$A(2) = \sum_k a_k 2^k = \dots + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} + \dots$$



Például:

$$A(2) = 101101,011(2) = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 45,375(10),$$

vagyis a szám leírásakor csak az a_k jegyeket írjuk le a megfelelő helyre, a súlyozás az illető jegy elhelyezkedéséből látszik (tizes rendszerben az egyesek, tízesek, százaskok, ... stb. helye a felirásból önkéntelenül adódik, ugyanez a helyzet a bináris rendszerben, csak itt egyesek, kettesek, négyesek, nyolcasok... stb. vannak az egyes helyiértékeken).

Egy számban, ill. a számot tartalmazó regiszterben a legnagyobb értékű (súlyu) bit helyét MSB-vel (Most Significant Bit), a legkisebb értékű helyét LSB-vel (Least Significant Bit) jelöljük:

$$\begin{array}{cccccccc} 2^7 & \dots & \dots & \dots & \dots & \dots & \dots & 2^0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \text{MSB} & & & & & & & \text{LSB} \end{array}$$

Konverziók

Gyakran előforduló feladat, hogy tizes számrendszerbeli számokat át kell írni kettesbe és viszont. Nem azért, mert a készülékek erre képtelenek, hanem azért, mert a tervezéskor, ellenőrzéskor, beméréskor, a fejlesztési munka közben előfordul, hogy pl. oszcilloszkópon vagy egyes párhuzamos vonalakon feszültségméréssel az eredmény, a vizsgált jel nem éppen a kívánt kódban jelenik meg, ezért "papír-ceruza" módszerrel kell elvégeznünk a konverziót. Tekintsük át ezt a módszert, a kódváltók elektronikus megoldásaival pedig a későbbiekben még foglalkozunk.

$10 \rightarrow 2$ tizes számrendszerből kettesbe:

Egész számok: a decimális számot 2-vel osztjuk, az osztás eredményét az eredeti szám alá írjuk, a maradékot (0 vagy 1) szintén leírjuk egy külön oszlopban. Az eredményt újra 2-vel osztjuk, az eredményt és a maradékot írjuk le, és újra osztunk 2-vel, amíg az osztandó 0-ra vagy 1-re "le nem fogy".

$$\begin{array}{r} \text{Pl.: } 238(10) = ?(2) \\ 238 \quad : 2 \\ 119 \quad 0 \\ \hline 238 \quad : 2 \\ 119 \quad 0 : 2 \\ 59 \quad 1 \\ \hline 238 \quad : 2 \\ 119 \quad 0 \\ 59 \quad 1 : 2 \\ 29 \quad 1 : 2 \\ 14 \quad 1 : 2 \\ 7 \quad 0 : 2 \\ 3 \quad 1 : 2 \\ 1 \quad 1 : 2 \\ 0 \quad 1 \uparrow \text{MSB} \end{array}$$

$$\underline{238(10) = 11101110(2)}$$

A külön leírt maradékokból álló oszlopot alulról felfelé összeolvassuk, ez adja a végeredményt.

Tört számok: az eredményt ismételt szorzással kapjuk meg: a számot 2-vel szorozzuk, az eredményt helyiérték helyesen aláírjuk, majd a tizedesvesszőtől jobbra eső részt újra 2-vel szorozzuk és így tovább. A tizedesvesszőtől balra keletkező 0-kat és 1-eket felülről lefelé összeolvassuk, ez a végeredmény:

pl.: $0,238_{(10)} = ?$	MSB ↓	0,	238 · 2	0,	21875 · 2
		0	476	0	43750
		0	952	0	87500
		1	904	1	75000
		1	808	1	50000
		1	616	1	00000
		1	232		
		.	.		
		.	.		
		.	.		

$$0,238_{(10)} = 0,001111_{(2)} \quad 0,21875_{(10)} = 0,00111_{(2)}$$

Ha a végeredmény véges tört, akkor előbb-utóbb (1)000-át kapunk a szorzás eredményeként, ha végtelen tört, akkor addig szorozgatunk 2-vel, amíg elegendő jegyet nem kapunk (véges tizedes tört 2-es megfelelője végtelen tört lehet!).

2 → 10 kettes számrendszerből tizesbe:

Egész számok: legcélszerűbb eljárás, ha a bináris számot a legnagyobb helyiértékű jeggyel felülről lefelé egy oszlopba írjuk, ettől balra egy vonalat húzunk. Az oszlop első 1-esét megszorozzuk 2-vel, az eredményt a vonaltól balra leírjuk, de ha az oszlop következő eleme 1-es, akkor az eredményhez 1-et hozzá kell adni, és azt leírni. Az így kapott és a vonal bal oldalára leírt számot újra 2-vel szorozzuk és ha mellette az eredeti számban 1-es van, akkor 1-et hozzáadunk, majd az eredményt újra szorozzuk 2-vel, stb.

$$\text{pl.: } 11101110_{(2)} = ?_{(10)}$$

	↓	
		1 · 2 MSB
2 + 1 = 3		1 · 2
6 + 1 = 7		1 · 2
14 + 0 = 14		0 · 2
28 + 1 = 29		1 · 2
.	59	1 · 2
.	119	1 · 2
.	<u>238</u>	0

Amikor a bináris szám minden jegyét "hozzáadásra" felhasználtuk, akkor az utolsó szorzat adja a végeredményt.

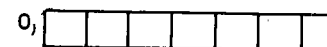
Tört számok: célszerű eljárás, ha a bináris törtszám jegyeit a tizedesvesszőtől kezdve alulról felfelé egy oszlopba leírjuk, ettől jobbra egy vonalat húzunk. Felülről kezdve az oszlop jegyeiből és a vonal jobb oldalán levő jegyekből álló számot 2-vel osztjuk, az eredményt mindig helyiérték-helyesen egy sorral lejjebb leírjuk stb. (az első sorban a vonal jobb oldalán csupa nullát képzelünk). A végeredmény az utolsó sorban levő osztáseredmény.

Pl.: $0,000111_{(2)} = ?_{(10)}$	1	0	:2	(10:2 = 5)
	1	5	:2	(15:2 = 7,5)
	1	75	:2	(175:2 = 87,5)
	0	875	:2	(875:2 = 437,5)
	0	4375	:2	(4375:2 = 2187,5)
	MSB ↑	0,	<u>21875</u>	

A szabályok természetesen érvényesek bármely más számrendszerre való átirásra is, csak éppen a megfelelő osztók és szorzók értéke nem 2, hanem a kérdéses számrendszer radix-a.

Bináris számábrázolás digitális berendezésekben Nagyságrend ábrázolása

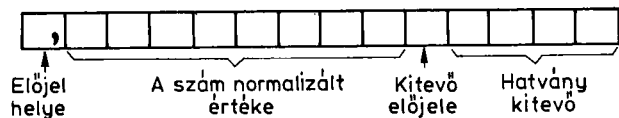
Fixpontos ábrázolás esetén a tizedesvessző (radix vessző) a számot tartalmazó regiszterben előzetes megállapodás szerinti rögzített helyen van, rendszerint az első értékes jegy előtt, vagyis minden szám csak 1-nél kisebb lehet (1.14. ábra).



1.14. ábra

Mérőműszerekben, adatfeldolgozó berendezésekben a mértékegységet úgy kell megválasztani, hogy a mérőszám kisebb legyen 1-nél, a számokon végzett műveletek eredménye sem lehet 1-nél nagyobb.

Lebegőpontos ábrázolásmód esetén a számok normalizált, féllogaritmikus alakban vannak. A regiszter tartalmazza a szám normalizált értékét, amely a fixpontoshoz hasonlóan valamilyen 1-nél kisebb szám (de a radix vessző után mindjárt értékes jegy van), ezt követően tartalmazza a szorzó 10 hatvány, ill. 2 hatvány kitevőjének előjel bit-jét és a hatványkitevő értékét (1.15. ábra).



pl.:

0	1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---	---

 . 2

1	1	0	1	1	0	1
---	---	---	---	---	---	---

1.15. ábra

Mivel a radix vessző előtt csak 0 állhat, ezt a helyet mindkét rendszerben az előjel megadásához használják.

Előjel ábrázolása

Az előjeles számok háromféle szokásos ábrázolási lehetősége a következő:

- a) előjel-nagyság (előjel-abszolút érték),
- b) 1-es komplement,
- c) 2-es komplement.

Először a komplement szám fogalmát ismételjük át:

2-es komplement (RADIX komplement).

A komplement képzés általános alapelve:

$$A + A^{\#} = r^N \quad \text{ahol} \quad \begin{array}{l} A \text{ az eredeti szám,} \\ A^{\#} \text{ az } A \text{ szám komplemente,} \\ r \text{ radix,} \\ N \text{ a számjegyek max. száma az} \\ \text{illető számkörben (ezt} \\ \text{előre rögzítenünk kell),} \end{array}$$

tehát a komplement:

$$\underline{A^{\#} = r^N - A}$$

például $N = 4$

$$\begin{array}{r} A^{\#} = 10^4 - 2245 \qquad 10000 \\ \qquad \qquad \qquad - 2245 \\ \hline A^{\#} = \qquad 7745 \end{array}$$

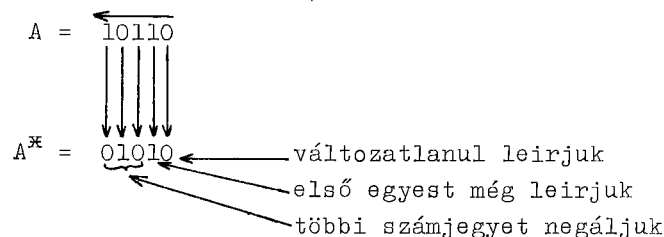
vagyis az eredeti számot a rendszerben lehetséges legnagyobb "kerék" számból ki kell vonni.

Kettes számrendszerben a kettes komplement:

keressük pl. az 10110 kettes komplementét, ha $N = 5$

$$\begin{array}{r} N = 5 \quad A^{\#} = 2^5 - 10110 \qquad 100000 \\ \qquad \qquad \qquad \qquad \qquad - 10110 \\ \hline 01010 = A^{\#} \end{array}$$

gyors módszer: jobbról-balra haladva az eredeti szám 0 jegyeit változatlanul leírjuk, amíg az első 1-hez nem érünk. Az első 1-est is még változatlanul leírjuk, de ettől kezdve a 0-kat 1-re, az 1-eket 0-ra cseréljük fel (negáljuk).



1-es komplement

Alapösszefüggése általában:

$$A + B^{\#} = r^N - 1 \quad \text{ahol} \quad \begin{array}{l} A \text{ az eredeti szám,} \\ B^{\#} \text{ a szám 1-es komplemente,} \\ r \text{ radix,} \\ N \text{ a számjegyek max. száma.} \end{array}$$

$$\underline{B^* = (r^N - 1) - A.}$$

A tízes számrendszerben ennek az un. 9-es komplementum felel meg.

Kettes számrendszerben: pl. keressük az 10110 1-es komplementumát, ha $N = 5$.

$$B^* = (r^N - 1) - A = 2^5 - 1 - A$$

11111	$2^5 - 1$
- 10110	- A
$B^* =$	01001

az 10110 1-es komplementum tehát:
01001,

vagyis az egy bináris szám 1-es komplementum a számjegyek invertálásával, 0-1 cseréjével kapható meg. (Az 1-es komplementum szám természetesen 1-gyel kisebb abszolút értékű, mint a 2-es komplementum.)

Előjeles bináris számok ábrázolása

a) Előjel-nagyság (-abszolút érték) ábrázolás

Olyan rendszerekkel foglalkozunk, amelyek 1-nél kisebb számokat ábrázolnak. A radix vessző előtti első hely, vagyis a számok első bitje az előjel ábrázolására szolgál. Az előjel-nagyság szerinti ábrázolásban az azonos abszolút értékű, de különböző előjelű számok a (képzelt) "tizedesvessző"-től jobbra eső részben teljesen megegyeznek, csak az előjel-bit különbözik.

$$\text{Pl.: } +26 (.2^{-5}) = 0,11010$$

$$-26 (.2^{-5}) = 1,11010.$$

b) Előjeles 1-es komplementum ábrázolás

A pozitív számokat hasonlóan az előzőhöz, 0 előjel-bit-tel a "tisztá bináris" formában, egyenes kódban ábrázoljuk. A negatív számokat komplementum (negált) kódban ábrázoljuk, az előjel-bit 1 lesz. Az azonos abszolút értékű negatív és pozitív számok tehát egymásnak komplementumai, valamennyi helyen a 0-ák és 1-ek fel vannak cserélve.

$$\text{Pl.: } +26 (.2^{-5}) = 0,11010$$

$$-26 (.2^{-5}) = 1,00101$$

c) Előjeles 2-es komplementum ábrázolás

A pozitív számok azonosak az előbbi két rendszerrel (0 előjel és egyenes kód). A negatív számok 2-es komplementum kódban jelennek meg: ezt úgy kapjuk, hogy a negatív szám pozitív megfelelőjét 2-ből kivonjuk.

$$\text{Pl.: } +26 (.2^{-5}) = 0,11010$$

$$-26 (.2^{-5}) = 1,00110 \quad \text{mert: } 10,00000 \quad (=2)$$

- 0,11010	-(szám)
1,00110	kompl.

Negatív számot ábrázoló 2-es komplementum a már ismert gyorsított módszerrel is megkapható.

$.2^{-3}$	Előjel-nagyság	1-es kompl.	2-es kompl.	Offset binary
(8)				
7	0,111	0,111	0,111	1,111
6	0,110	0,110	0,110	0,110
5	0,101	0,101	0,101	0,101
4	0,100	0,100	0,100	1,100
3	0,011	0,011	0,011	1,011
2	0,010	0,010	0,010	1,010
1	0,001	0,001	0,001	1,001
0	0,000	0,000	0,000	1,000
-1	1,001	1,110	1,111	0,111
-2	1,010	1,101	1,110	0,110
-3	1,011	1,100	1,101	0,101
-4	1,100	1,011	1,100	0,100
-5	1,101	1,010	1,011	0,011
-6	1,110	1,001	1,010	0,010
-7	1,111	1,000	1,001	0,001
-(8)			1,000	0,000

A 2-es komplementum úgy jön létre az 1-es komplementumból, hogy az 1-es komplementum utolsó jegyéhez (LSB) 1-et hozzáadunk. Az előző oldalon levő táblázatban felírtuk a +8 ($.2^{-3}$) és a -8 ($.2^{-3}$) közötti számokat a háromféle ábrázolásban, ehhez hozzá vettük az esetenként használatos "offset binary" kódot is (az "eltolt bináris" kód a legnegatívabb szám helyére rögzíti a zérót, csak előjel-bitben tér el a 2-es komplementustól).

Összeadás és kivonás a háromféle számábrázolással

Az összeadás pozitív számok esetén mindhárom számábrázolással ugyanolyan: a megfelelő helyiértékeket összeadjuk az előző helyiértékben keletkező átvitel figyelembevételével. Az összeg nem érheti el az 1-et, máskülönben a regiszter túlcserél.

A kivonás, ill. negatív számok összevonása szintén helyiértékenkénti összeadással történik, az illető számábrázolás sajátosságainak megfelelő szabályoknak megfelelően.

A továbbiakban mindhárom számábrázolással összefoglaljuk az összevonási módszereket. (A példákban az öt értékes jegyet tartalmazó bináris számok mellől elhagyjuk a 2^5 szorzót.)

a) Előjel-abszolút értékkel (előjel-nagysággal) adott számok összeadása, kivonása

- Két pozitív szám vagy két negatív szám:

+13:	0,01101	-13:	1,01101	összeadjuk az abszolút értékeket
+11:	+0,01011	-11:	1,01011	
+24:	0,11000	-24:	1,11000	leírjuk a közös előjelet.

- Ellentétes előjelű számok, az első (felső) abszolút értéke nagyobb, mint a másodiké:

+13:	0,01101	-13:	1,01101
-11:	1,01011	+11:	0,01011
+ 2	↓	+ 2	↓
	01101		01101
	+10100		+10100
	100001		100001
	+ → 1		+ → 1
	00010		00010

eredm.: 0,00010 eredm.: 1,00010

a felső szám absz.ért-hez a második szám komplementusának (invertáltjának) hozzáadása, átvitel keletkezik, ezt hozzáadjuk, így keletkezik az eredmény abszolút értéke

előjel helyére a felső szám előjelét írjuk.

- Ellentétes előjelű számok, a második abszolút értéke nagyobb, mint az elsőé vagy egyenlők:

+11:	0,01011	-11:	1,01011
-13:	1,01101	+13:	0,01101
- 2	↓	+ 2	↓
	01011		01011
	+10010		+10010
	11101		11101
	00010		00010

eredm.: 1,00010 eredm.: 0,00010

az első szám abszolút értékéhez a második absz.ért. komplementusának hozzáadása, nem keletkezik átvitel,

a végeredmény az összeg komplementusa,

előjele a második szám előjele.

b) 1-es komplementussal adott számok összevonása

Mindkét szám pozitív vagy mindkettő negatív:

+13:	0,01101	-13:	1,10010
+11:	+0,01011	-11:	+1,10100
+24:	0,11000	-24:	11,00110
			→ 1
eredm.:		eredm.:	1,00111

összeadjuk az 1-es kompl. számokat az előjel bittel együtt.

Ha az első helyen (MSB) átvitel keletkezik, akkor ezt az LSB-hez hozzáadjuk, és a végeredmény előjelhelyesen, 1-es kompl. kódban keletkezik.

Ellentétes előjelű számok, negatív eredmény:

-13:	1,10010	+11:	0,01011	összeadjuk a számokat az
+11:	<u>0,01011</u>	-13:	<u>+1,10010</u>	előjel bittel együtt,
- 2	1,11101	- 2	1,11101	nem keletkezik átvitel, az
				eredmény helyes.

Ellentétes előjelű számok, pozitív eredmény:

+13:	0,01101	-11:	1,10100	
-11:	<u>+1,10100</u>	+13:	<u>+0,01101</u>	
+ 2	10,00001	+ 2	10,00001	
	<u> </u>		<u> </u>	átvitel keletkezik az el-
	0,00010		0,00010	ső helyen, ezt hozzáadjuk.

Tehát ebben a rendszerben valamennyi esetben helyes eredményt akkor kapunk, ha a számokat az előjel bittel együtt összeadjuk (a számok 1-es komplementében vannak), ha keletkezik átvitel, akkor azt a legkisebb helyiértékre átvisszük és az összeghez hozzáadjuk, így az eredményt előjelhelyesen kapjuk (1-es komplementes ábrázolásban).

Ha a második számot ki akarjuk vonni, akkor az előjel bittel együtt komplementáljuk a kivonandót (akár pozitív, akár negatív), és hozzáadjuk (az előjel bittel együtt) a kisebbbitendőhöz.

c) 2-es komplementessel megadott számok összevonása

Összeadás:

-13:	1,10011	negatív szám: 2-es komplementesben,		
-11:	<u>+1,10101</u>	negatív szám: 2-es komplementesben,		
-24	11,01000	az első helyen keletkező átvitelt elhagyjuk, a végeredmény negatív, 2-es komplementesben.		
+13:	0,01101	-11:	1,10101	
-11:	<u>+1,10101</u>	+13:	<u>+1,01101</u>	
+ 2	10,00010	+ 2	10,00010	az első helyen keletkező átvitelt elhagyjuk, az eredmény előjele pozitív (egyenes kód)

-13:	1,10011	+11:	0,01011	
+11:	<u>0,01011</u>	-13:	<u>1,10011</u>	átvitel nem keletkezik, a végeredmény negatív előjelű (2-es komplementesben)
- 2	1,11110	- 2	1,11110	

Kivonás célszerű módja:

+13:	0,01101	+13:	0,01101
-(+11):	<u>-0,01011</u>	-(-11):	<u>-1,10101</u>
+ 2		+ 24	
	0,01101		0,01101
	1,10100		0,01010
	<u> </u>		<u> </u>
	10,00010		0,11000

képezzük a kivonandó 1-es komplementesét (áramkörileg könnyebb: még 1-et hozzáadunk (így lesz 2-es kompl.) eredmény (első helyen az átvitelt - ha van - elhagyjuk)

Tehát a 2-es komplementes ábrázolással adott számokat az előjel bittel együtt összeadjuk, ha az első helyen átvitel keletkezik, akkor azt elhagyjuk, az eredmény előjelhelyes lesz. Ha a második számot ki akarjuk vonni, akkor legegyszerűbb, ha ezt a kivonandót az előjel bittel együtt komplementáljuk (negáljuk) és hozzáadjuk a kisebbbitendőhöz, valamint még egyet hozzáadunk a legkisebb helyiértéken (LSB).

Összefoglalásul felrajzoljuk egy 1-es komplementesben, és egy 2-es komplementesben dolgozó 4 bites párhuzamos összeadó-kivonó tömbvázlatát. A blokkok készen kaphatók integrált áramköri formában, pl. a TTL áramkörök közül pl. a 7483-as egy 4 bites teljes összeadó, a 74H87-es egy 4 bites áteresztő-komplementáló. Így a blokkvázlat egyben áramköri kapcsolási rajznak is tekinthető (1.16. ábra).

(A felrajzolt tömbvázlatot, ill. kapcsolást természetesen 4-nél több bitre több blokk egymás után kapcsolásával bővíteni lehet. A szerkezet ugyanilyen marad, minden 4 bithez egy teljes összeadó és egy komplementáló áramkör kell. Az MSB felé haladva az előző összeadó C_4 kimenetét mindig a következő C_0 bemenetével kell összekötni és a több fokozat összekötése után a lánc elején, ill. végén maradó C_4 -et, ill. C_0 -át ugyanugy kell felhasználni, mint eredetileg.)

1.3.2. BCD (Binary-Coded-Decimal) kódok

Négy bites BCD kódok

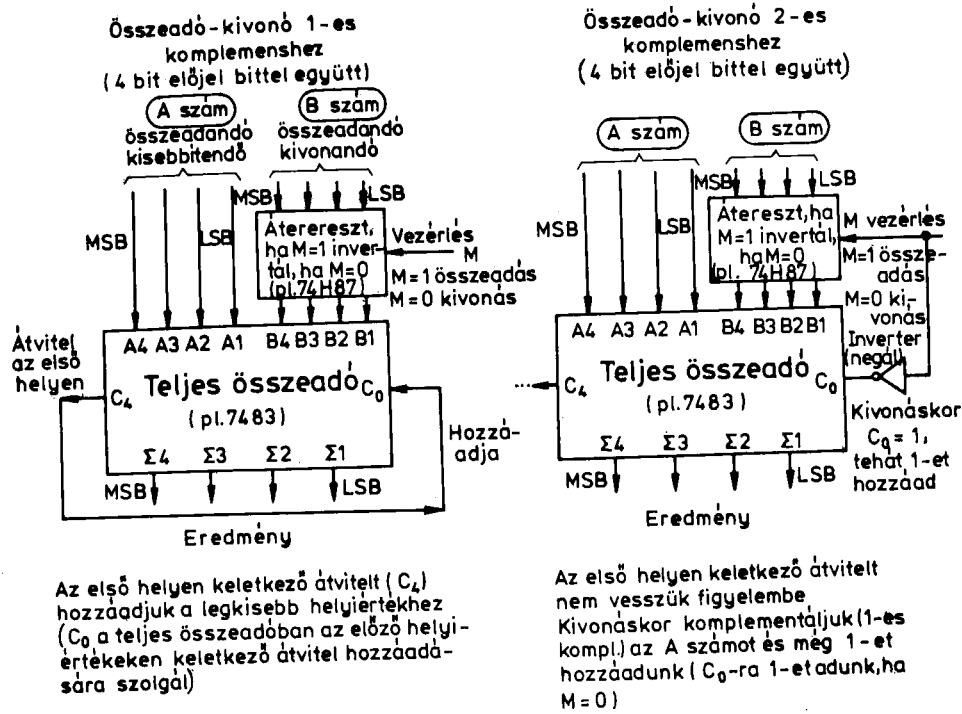
Négy bittel: $2^4 = 16$ különböző kódszót lehet előállítani. Ha a 16 kódszóból álló készletből 6-ot kirekesztünk, akkor 10 féle lehetőség marad, amelyet a decimális számjegyek kódolására használhatunk. Attól függően, hogy milyen súlyozásuk az egyes helyiértékek és, hogy milyen a felépítés szisztemája, többféle BCD kód terjedt el. Ezek közül a legfontosabbak:

	8-4-2-1 súlyozású	Excess-3 (Stibitz) (3-többletes)	2-4-2-1 (Aiken)	5-4-2-1
	2^3 2^2 2^1 2^0	2^3 2^2 2^1 2^0	2 4 2 1	5 4 2 1
	D C B A	D C B A	D C B A	D C B A
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1
2	0 0 1 0	0 1 0 1	0 0 1 0	0 0 1 0
3	0 0 1 1	0 1 1 0	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 1 1	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 0 0	1 0 1 1	1 0 0 0
6	0 1 1 0	1 0 0 1	1 1 0 0	1 0 0 1
7	0 1 1 1	1 0 1 0	1 1 0 1	1 0 1 0
8	1 0 0 0	1 0 1 1	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 0 0	1 1 1 1	1 1 0 0

A 8-4-2-1 súlyozású kódot szoktuk általában "BCD kód"-nak nevezni. Evvel a decimális számok egyes jegyeit kódoljuk binárisan, 4-es csoportokban.

Pl.: 3 8 5 7
 0011 1000 0101 0111

Nagyon sok készülék van a mérés technikában és az automatizálásban, amely ebben a BCD-kódban dolgozik. Minden olyan esetben, amikor az eredmények, adatok, stb. kijelzésre kerülnek,

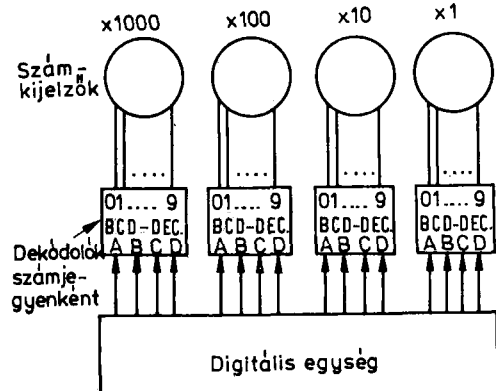


1.16. ábra

Az előjel-abszolút értékkel adott bináris számok összeadásának és kivonásának megoldása nem ilyen egyszerű, ezzel itt nem foglalkozunk. Ebben a formában adott számokkal szorzás-osztás viszont általában könnyebb, mint a többivel.

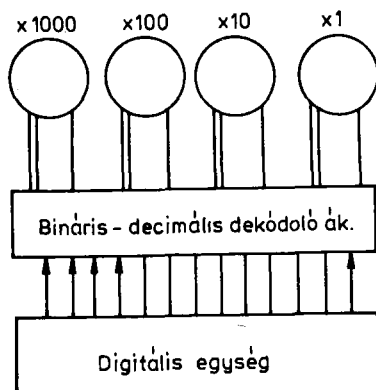
Az összeadás-kivonás 2-es komplementben a legegyszerűbb, és mai rendszerekben szinte kizárólagos a használata (kalkulátor, számítógép, mikroprocesszor) ezért, ha külön nem említjük, akkor a bináris aritmetikai műveleteket mindig 2-es komplementben megjelenítésben végezzük. Ezt a rendszert a későbbiek megértése érdekében feltétlenül jól kell ismernünk! (A műveletek 1-es komplementben sem sokkal bonyolultabbak - lásd a blokkvázlatokat -, csak az első helyen keletkező átvitel újbóli hozzáadása a művelet végzés sebességét csökkenti.) A szorzás-osztás 2-es komplementben és 1-es komplementben nehezebb, mint előjel-abszolút értékkel.

számjegyesen leolvashatók, célszerű, hogy maga a készülék is BCD rendszerben működjön, mert így a decimális jegyek dekódolásához (BCD-ből kijelezhető decimális rendszerbe való átalakításához) egy-egy azonos felépítésű dekódoló kell (1.17. ábra).



1.17. ábra

Amennyiben egy ugyanilyen készülék tiszta bináris számokkal dolgozna, egy nagyon bonyolult dekódolóra lenne szükség, amely a sokjegyű bináris számot többjegyű decimális számra alakítaná át. Ezt párhuzamos dekódolásnál pl. ROM-mal (Read Only Memory, állandó tartamu memória), soros dekódolásnál pl. számláncokkal lehet megoldani, - mindkettő költséges (1.18. ábra)



1.18. ábra

Tekintve, hogy a BCD kód 4 bitje a lehetséges 16 állapot közül csak 10-et használ, a BCD áramkörök (pl. számlálók) nincsenek "teljesen kihasználva", ami azt jelenti, hogy ugyanazon feladat ellátására 16/10-szer annyi áramkör kell, mint binárisban. A dekódolás viszont egyszerűbb. Ezen fő szempontok alapján kell eldöntenünk egy konkrét esetben, hogy melyik kód célszerűbb.

A továbbiakban a BCD számjegyekben az egyes helyiértékeket A, B, C, D-vel jelöljük. Mindig az "A" jelenti a legkisebb helyiértéket, az LSB-t.

MSB	2^3	2^2	2^1	2^0	LSB
	D	C	B	A	

Az Excess-3 (3-többletes) kód abban különbözik a 8-4-2-1 BCD-től, hogy a 0-nak a bináris 3-as, 0011 felel meg, az 1-nek a bináris 4-es és így tovább, vagyis mindig 3-mal több, mint a tényleges érték. A kód jellegzetességei:

- minden kódszó tartalmaz 1-est (nincs 0000),
- "önkomplementáló", azaz a kódtáblázatban a 4-5 között egy elképzelt szimmetria tengelytől egyenlő távolságra lévő számok egymásnak komplementensei (a bitek egymás negáltjai,

pl.: 4: 0111
 5: 1000).

A 2-4-2-1 Aiken kód szintén önkomplementáló. Az 5-nél kisebb számokhoz a "B", 2-es súlyozású helyiértéket, az 5 és annál nagyobb számokhoz a "D" jelű, 2-es súlyozású helyiértéket használjuk. Az önkomplementáló tulajdonság a digitális berendezésekben néhol előnyös lehet aritmetikai műveletek végzésekor. Az Excess-3 és az Aiken kód előnye az is, hogy a "D" helyiértéken lévő számjegyből lehet tudni, hogy a szám 5-nél kisebb-e (D=0) vagy 5, ill. annál nagyobb-e (D=1).

Az 5-4-2-1 súlyozású kód első, MSB helyiértékéből szintén el lehet dönteni, hogy a szám 5-nél kisebb-e, vagy sem. A fennmaradó három biten 0-tól 4-ig és 5-től 9-ig a számok bináris kód szerint növekednek, a két tartományban ismétlődően.

Egyéb BCD kódok

	5-ből 2	JOHNSON	BI - QUINER
	7 4 2 1 0	E D C B A	5 0 4 3 2 1 0
	sulyo- zás		
0	1 1 0 0 0	0 0 0 0 0	0 1 0 0 0 0 1
1	0 0 0 1 1	0 0 0 0 1	0 1 0 0 0 1 0
2	0 0 1 0 1	0 0 0 1 1	0 1 0 0 1 0 0
3	0 0 1 1 0	0 0 1 1 1	0 1 0 1 0 0 0
4	0 1 0 0 1	0 1 1 1 1	0 1 1 0 0 0 0
5	0 1 0 1 0	1 1 1 1 1	1 0 0 0 0 0 1
6	0 1 1 0 0	1 1 1 1 0	1 0 0 0 0 1 0
7	1 0 0 0 1	1 1 1 0 0	1 0 0 0 1 0 0
8	1 0 0 1 0	1 1 0 0 0	1 0 0 1 0 0 0
9	1 0 1 0 0	1 0 0 0 0	1 0 1 0 0 0 0

Az "5-ből 2" és a BI-QUINER kódok jellegzetessége, hogy mindegyik kódszóban két 1-es van. Ez a tulajdonság bizonyos hibavédettséget kölcsönöz ezeknek a kódoknak (ha egy kódszóban nem két 1-es van, akkor észrevehetően valami hiba van a rendszerben).

A JOHNSON kódot digitális érzékelőkhöz gyakran használják (pl. helyzet-kódoláshoz előnyös). Az ún. "egylépéses" kódok közé tartozik, mert a szomszédos kódszavak egyetlen helyen különböznek egymástól. Ezek a kódok nem túl gazdaságosak, hiszen 5, ill. 7 bitet használnak 10 állapot előállítására, de vannak esetek, amikor egy egész rendszer szempontjából mégis előnyösek.

1.3.3. Hexadecimális kód

A hexadecimális kód nevének megfelelően 16 kódszót tartalmaz. 4 bitjével bináris kódban mind a 16 lehetséges kombinációt tartalmazza (nem "áll meg" 9-nél, mint a BCD). A mi szokásos 10-es számrendszerünkben meglévő 10 darab szám-szimbólum nem elég a 16 féle szám megjelölésére, ezért a számsort 9 felett az ábécé betűivel folytatjuk:

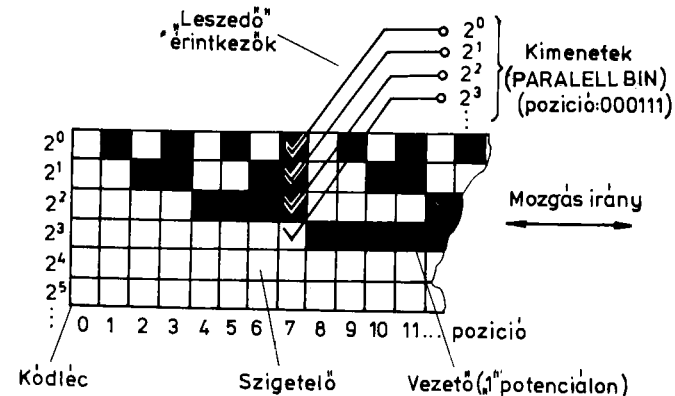
JEL	2 ³	2 ²	2 ¹	2 ⁰	JEL	2 ³	2 ²	2 ¹	2 ⁰
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	A	1	0	1	0
3	0	0	1	1	B	1	0	1	1
4	0	1	0	0	C	1	1	0	0
5	0	1	0	1	D	1	1	0	1
6	0	1	1	0	E	1	1	1	0
7	0	1	1	1	F	1	1	1	1

Igy a hexadecimális kód a bináris számok "rövidített" felírására alkalmas: a bináris 0-1 sorozatot az LSB felől 4-es csoportokra választjuk szét és, ezeket adjuk meg egy-egy számmal, ill. betűvel. Például az 1111 1001 1000 1100 bináris számot hexadecimális rövidítéssel a következőképpen ábrázolhatjuk:

1111	1001	1000	1100	1111100110001100	→ F98C H
F	9	8	C		

1.3.4. "Egylépéses" kódok

Az ún. egylépéses kódok meglehetősen fontosak és gyakran fordulnak elő a műszeriparban és az automatikában. Legtipikusabb példa az elmozdulás, elfordulás, ill. helyzet digitális érzékelése. Ezekhez ún. kódlécet vagy kódtárcsát használnak.

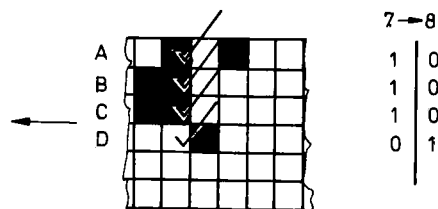


1.19. ábra

Egy bináris kódlec egyszerűsített vázlatát mutatja az 1.19. ábra.

Attól függően, hogy a mozgó gép-részre (melynek az elmozdulását mérni akarjuk) felerősített kódlec milyen relatív helyzetben van az álló "leszedő érintkezők"-höz képest, a leszedő kimeneteken 0 és 1-nek megfelelő feszültség szintek jelennek meg. Ha valamelyik helyiértékhez (valamelyik sorhoz) tartozó érintkező éppen szigetelőhöz ér, a kimenet "0" feszültségű, ha vezető bevonathoz ér, a kimenet "1" feszültségű. A bináris kódleccen a fémezett négyzetek az egymás utáni bináris számoknak megfelelően vannak elhelyezve, így a kimeneteken a pozíciónak megfelelő kombinációk jelennek meg (lásd az 1.19. ábrát).

A felrajzolt bináris kódlec nagy hátránya, hogy az egyik pozícióról a másikra való átmenetkor igen nagy leolvasási hiba keletkezhet, ha az érintkezősor elhelyezése nem precíz, az érintkezők nincsenek egész pontosan egyvonalban vagy az érintkezősor vonala nem pontosan merőleges a mozgás irányára. Gondoljuk meg, például, hogy az első 4 bitet figyelve milyen hiba keletkezhet a 7-esből a 8-as pozícióba való átmenetkor (1.20. ábra).



1.20. ábra

Egyáltalán nem várható mégoly precíz mechanikánál sem, hogy bal felé elmozduló kódlec esetén az A,B,C sávon a leszedők egyszerre váljanak le a fémezett részről és pontosan akkor, amikor a D sávon az érintkező a fémezett részhez ér. A leszedők tengelyének egészen kismértékű ferdülése esetén előfordulhat, hogy az A,B,C sáv érintkezői még a fémezett részhez érnek, amikor már a D sáv leszedője is a fémezett részhez ér. Ekkor egy pillanatra az 1111, azaz 15-ös szám jelenik meg

a kimeneten. Az is lehet, hogy az A,B,C sáv leszedői leváltak a fém részről még mielőtt a D érintkező hozzáért volna a fém részhez. Ekkor egy pillanatra a 0000 szám jelenik meg. Helyes működés esetén a 7-es után közvetlenül a 8-asnak kellett volna megjelennie, ehelyett a valóságban az átmenet pillanatában akár a 15-ös, akár a 0 is megjelenhet (vagy bármely más szám 0 és 15 között), ami megzavarhatja az információt fogadó berendezés működését (és ezáltal hibás vezérlést adhat).

Az egylépéses kódok felhasználása ezt a hibalehetőséget küszöböli ki. Ezekre a kódokra az jellemző, hogy az eggyel nagyobb számértéket jelentő kódszó az előzőtől csak egyetlen helyiértéken különbözik, vagyis az egyik kódszóról az utána következőre való átmenetnél csak egy bit "lép", vált át 0-ról 1-re vagy 1-ről 0-ra. A leggyakrabban használt egylépéses kód a GRAY kód:

	MSB		LSB		
D	C	B	A	
	$2^n-1..2^4-1$	2^3-1	2^2-1	2^1-1	
0.	0	0	0	0	0 0 0 0
1.	0	0	0	1	0 0 0 1
2.	0	0	1	1	0 0 1 1
3.	0	0	1	0	0 0 1 0
4.	0	1	1	0	0 1 1 0
5.	0	1	1	1	0 1 1 1
6.	0	1	0	1	0 1 0 1
7.	0	1	0	0	0 1 0 0
8.	1	1	0	0	1 1 0 0
9.	1	1	0	1	1 1 0 1
10.	1	1	1	1	1 1 1 1
11.	1	1	1	0	1 1 1 0
.
.

0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0

szimmetria

tengelyek

A TÜKRÖZÖTT BINÁRIS elnevezést is használják, mert a kódtáblázatban egyes zónák egymásnak tükörképei. A GRAY kódnál nem fordul tehát elő olyan eset, mint pl. a bináris 7 és 8-nál, hiszen ott a szomszédos kódszavak valamennyi helyiértékén eltérés van (1-ek 0-ra váltanak, a 0-ák 1-re):

7.	0 1 1 1	0 1 1 1
		↓ ↓ ↓ ↓
8.	1 0 0 0	1 0 0 0

A GRAY kód minden szomszédos kódszava között csak egyetlen helyen van eltérés, pl.:

7.	0 1 0 0
	↓
8.	1 1 0 0

Ezért van az, hogy a GRAY kódléc alkalmazásakor nem keletkezhet érzékelési hiba, nem "ékelődhetnek közbe" hibás kombinációk (nem kell a leszedő érintkezőknek több helyen "ugrani"). Ugy is kifejezhetjük ezt, hogy a szomszédos kódszavak a lehető legkisebb "távolságra" vannak egymástól, hiszen majdnem egyformák, egy bitet kivéve. Ez az említett "távolság" annál nagyobb, minél több helyiértéken van változás. Ahány bit-ben eltér az egyik kódszó a másiktól, annyi a két kódszó HAMMING TÁVOISÁGA (H).

A GRAY kódtáblázatban az egyes bitek súlyozása is fel van tüntetve. Az eddigiekhez képest lényeges eltérés, hogy a számértéket nem azoknak a helyiértékeknek az egyszerű súlyozott összege adja, amelyeken 1-es van, hanem az 1-esek súlyozott értékét változtatott előjellel kell képezni: a legértékesebb jegy felől (MSB) haladva az első 1-es helyiértékét pozitív előjellel kell venni, a következő 1-es helyiértékét negatívvá, majd megint pozitívvá, stb.

Pl.:

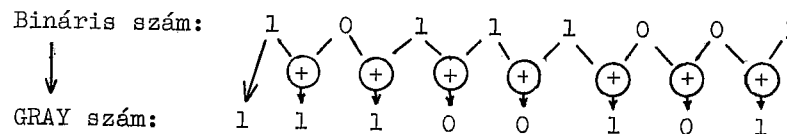
	15 7 3 1	
4. →	0 1 1 0	→ +7-3 = 4
10. →	1 1 1 1	→ +15-7+3-1 = 10
	GRAY	

Sokszor előfordul, hogy szükség van valamely bináris szám GRAY számmá való átalakítására és viszont.

Bináris-Gray konverzió

A bináris szám szomszédos jegyeit a modulo-2, átvitel nélküli összeadás szabályai szerint ($0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 1 = 0$) összeadjuk és megkapjuk a szám GRAY megfelelőjének jegyét. A bináris szám legelső 1-esét (MSB) változatlanul leírjuk!

Pl.:

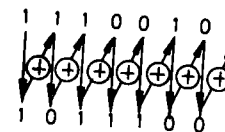


GRAY-bináris konverzió

Az MSB-től kezdve az eredményül kapott bináris szám jegyét a következő GRAY jegyhez adjuk modulo-2 összeadással, így megkapjuk az eredmény következő jegyét. A kiindulási GRAY szám első jegyét változatlanul leírjuk.

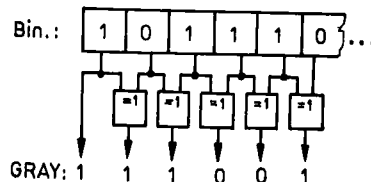
Pl.:

GRAY szám:

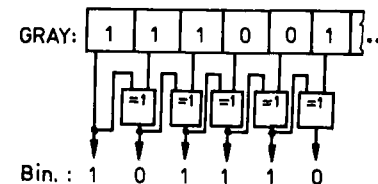


Bináris szám:

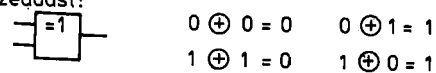
Bináris - GRAY kód konverter:



GRAY - bináris kód konverter:



A kizáró-VAGY áramkör valósítja meg az átvitel nélküli összeadást:



1.21. ábra

Tekintve, hogy modulo-2 összeadó (kizáró-VAGY) kapu áramkör készen kapható (lásd később), a párhuzamos kód-konverzió áramköri megoldása meglehetősen egyszerű és teljes egészében megfelel a papír-ceruza módszernek (1.21. ábra).

Érdemes megjegyezni, hogy a helyzetkódolásban a hiba csökkentésére nemcsak a GRAY kód felhasználása az egyedüli módszer. Igen sok "trükk" és kód változat létezik még ezen kívül. Ennek illusztrálására néhány további, decimális egylépéses kódot mutatunk be a következő táblázatban:

	GLIXON	O'BRIEN I	O'BRIEN II	TOMPKINS I	TOMPKINS II	TÜKRÖZÖTT 3 TÖBB- LETES
	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA
0.	0000	0000	0001	0000	0010	0010
1.	0001	0001	0011	0001	0011	0110
2.	0011	0011	0010	0011	0111	0111
3.	0010	0010	0110	0010	0101	0101
4.	0110	0110	0100	0110	0100	0100
5.	0111	1110	1100	1110	1100	1100
6.	0101	1010	1110	1111	1101	1101
7.	0100	1011	1010	1101	1001	1111
8.	1100	1001	1011	1100	1011	1110
9.	1000	1000	1001	1000	1010	1010

1.4. Alfánumerikus kódok (gyakori betű-szám kódok)

1.4.1. Nemzetközi táviró-kód (BAUDOT)

Ez az 5 bites alfánumerikus kód a hírközlésben szinte a legelterjedtebb nemzetközi kód. A TELEX gépek ezzel működnek, ezért TELEX-kódnak is nevezik.

Az 5 bit $2^5 = 32$ kódszót állíthat elő maximálisan, ebbe "nem fér be" az ABC összes betűje, valamint az összes szám és írásjel. Ezért minden kódszót kétszeresen használnak ki:ugyanaz a kombináció jelenthet valamely betűt vagy egy számot (ill.

írásjel) attól függően, hogy megelőzően az 11111 betűváltó vagy az 11011 számváltó kódszó érkezett-e. A betűváltó megérkezése után minden kódszó betűt jelent, egészen addig, amíg a számváltó meg nem érkezik, ezután minden kódszámot és írásjel jelent stb. 6 jelkombináció nem betűk és számok kódolására szolgál, hanem a telex gépet vezérli.

A CCITT szabványos nemzetközi táviró-kód táblázata:

KÓDJEL	BETŰ	SZÁM	KÓDJEL	BETŰ	SZÁM
11000	A	-	11100	Q	1
10011	B	?	01010	P	4
01110	C	:	10100	S	,
10010	D	Ki az?	00001	T	5
10000	E	3	11100	U	7
10110	F	"megj."	01111	V	=
01011	G	"megj."	11001	W	2
00101	H	"megj."	10111	X	/
01100	I	8	10101	Y	6
11010	J	csengő	10001	Z	+
11110	K	(00000	üres	} vezérlő kódok
01001	L)	11111	betűváltó	
00111	M	.	11011	számváltó	
00110	N	,	00100	szóköz	
00011	O	9	00010	kocsi-vissza	
01101	P	0	01000	soremelő	

"megj.": a jelentése nincs rögzítve, minden ország tetszőleges jelentést adhat neki.

A továbbítás általában sorosan történik, start-stop üzemben (lásd az 1.2. fejezetet). A soros biteket a TELEX gép a billentyük lenyomásakor mechanikus uton állítja elő és a vevő oldalon a bit szinkronizáció szintén mechanikus - a START, ill STOP jelek felhasználásával (1. az 1.12a ábrát).

1.4.2. ASCII (American Standard Code for Information Interchange = Amerikai szabvány kód információ cseréhez)

Igen sok periférikus berendezés működik ezzel a 8 bites kóddal. A nyolcadik bit igazság szerint nem hordozza az információt, hanem hibajelzésre szolgáló paritás bit, amely a karakterekben az 1-esek számát párosra egészíti ki. A maradék 2^7 kombináció bőségesen elegendő az összes kis- és nagybetű, szám írásjel és a vezérlőjelek kódolására. Azt, hogy kisbetűről, nagybetűről, számokról, ill. írásjelekről vagy vezérlőjelekről van e szó, a 6. és 7. bit határozza meg. A többi 5 biten kódolják az egyes betűket, írásjeleket, oly módon, hogy - ha például betűkről van szó - ahányas bináris számot alkotnak az első 5 bit jegyei, az ABC annyiadik betűjét jelentik (bináris szám - folytonos kódolás).

A kódtáblázat a következő:

7.6. bit				5.4.3.2.1.	7.6. bit				5.4.3.2.1.
00	01	10	11		00	01	10	11	
NULL	<	@	\	0 0 0 0 0	DLE	0	P	P	1 0 0 0 0
SOH	!	A	a	0 0 0 0 1	DC1	1	Q	q	1 0 0 0 1
STX	"	B	b	0 0 0 1 0	DC2	2	R	r	1 0 0 1 0
ETX	#	C	c	0 0 0 1 1	DC3	3	S	s	1 0 0 1 1
EOT	§	D	d	0 0 1 0 0	DC4	4	T	t	1 0 1 0 0
ENQ	%	E	e	0 0 1 0 1	NAK	5	U	u	1 0 1 0 1
ACK	&	F	f	0 0 1 1 0	SYN	6	V	v	1 0 1 1 0
BELL	'	G	g	0 0 1 1 1	ETB	7	W	w	1 0 1 1 1
BS	(H	h	0 1 0 0 0	CAN	8	X	x	1 1 0 0 0
HT)	I	i	0 1 0 0 1	EM	9	Y	y	1 1 0 0 1
LF	⌘	J	j	0 1 0 1 0	SUB	:	Z	z	1 1 0 1 0
VT	+	K	k	0 1 0 1 1	ESC	;	[{	1 1 0 1 1
FF	,	L	l	0 1 1 0 0	FS	<	\		1 1 1 0 0
CR	-	M	m	0 1 1 0 1	GS	=]	}	1 1 1 0 1
SO	.	N	n	0 1 1 1 0	RS	>	^	~	1 1 1 1 0
SI	/	O	o	0 1 1 1 1	US	?	-	DEL	1 1 1 1 1

Egy adott szimbólum (betű stb.) kódját úgy kapjuk meg a táblázatból, hogy a mellette álló 5 bit elé odairjuk 6. és 7. bitként annak az oszlopnak a kódját, amely oszlopban a kérdéses betű van.

A soros átvitel áram-idő diagramja pl. TELETYPE esetén az 1.12b ábrán látható.

1.4.3. Egyéb bináris alfanumerikus kódok

EBCDIC (Extended Binary Coded Decimal Interchange Code = Kibővített BCD információ cseréhez)

8 bites, és mindegyik helyiérték információt hordoz. A betűk, számok, írásjelek, vezérlőjelek nem használják ki mind a 256 kombinációt, sok hely van speciális jelek kódolására. Nagy számítógépekhez és a hozzájuk tartozó berendezésekhez használják előszeretettel.

SELECTRIC

8 bites kód, ebből 7 az információ hordozó, a nyolcadik a paritásbit, hibajelzés céljára.

"N az M-közül" kódok

Ahogy az előző kódoknál már láttuk (ASCII, SELECTRIC) hibajelzés céljára gyakran fenntartanak egy paritásbitet, amely a kódszóban levő 1-esek számát pl. párosra egészíti ki. Ha a vevő oldalon az 1-esek száma nem páros, akkor a készülék jelzi a paritáshibát. Az igazság azonban az, hogy az információ átvitele közben általában nem csak 1 bit "romlik el", hanem több is; a kódszóhoz esetleg hozzáadódik néhány bit zavarjel vagy éppen elmarad néhány bit. Ilyenkor a paritás jelzés nem túl hatásos, hiszen kb. 50% a valószínűsége annak, hogy a hibásan vett kódszó is páros 1-est tartalmaz és a hibát nem vesszük észre.

Az "N az M-közül" kódok jobb hibavédelmet nyújtanak. Az ilyenek kódkészlete olyan, hogy a karakterek M számú bitje között mindig N számú 1-es van. Pl. 4 a 8-közül kód olyan rend-

szerű, hogy bármelyik 8 bites kódszóban 4 egyes van. Vételkor azt kell figyelni, hogy pont 4 db 1-es (és 4 darab 0-ás) van e karakterben, ha nem, akkor jelzést kell adni. Így a fellépő zavar biteket, vagy kieséseket jobban észre lehet venni. Igaz, hogy ennek ára van: egy 8 bites kódrendszer (melyben 1 paritásbit van) összesen 128 kombinációt állíthat elő, a "4 a 8 közül" kód csak 70-et (6-tal többet mint egy tiszta bináris 6 bites kód).

1.5. Hibajelzés és hibajavítás

A digitális információ átvitele és feldolgozása közben sokszor fellép valamilyen hiba, főleg az átvivő csatorna által "felszedett" zajok és zavarok miatt és a két végén lévő készülékek fogyatékoságai miatt (torzítás, instabilitás, mechanikus érintkezők "pergése" stb.). Nem sorolhatjuk ide a kezelők által elkövetett hibákat, amelyek védhetetlenek.

Sok esetben bizonyos mértékű hiba-arány megengedett, nem zavaró; pl. telex szövegben egy-egy betű "elromlása" - az írott szöveg meglehetősen nagy bőbeszédűsége, többlet információ tartalma, rudundanciája miatt - nem változtatja meg az üzenet értelmét. Az 1 a 10^5 -ben arány például azt jelenti, hogy ha ennek a jegyzetnek a teljes szövegét továbbítanánk, akkor összesen néhány betű lenne hibás, amit általában az olvasók többsége észre sem vesz. Más rendszerekben viszont 1 a 10^7 , 1 a 10^8 -ban arány sem számít jónak.

Amennyiben kiderül, hogy az illető rendszerben az elérhető hibaarány rosszabb, mint ami megengedhető, akkor valamilyen módon biztosítani kell a hiba elleni védelmet. A keletkező hibákat felfedezni, észlelni vagy esetleg javítani csak akkor lehet, ha a továbbításra olyan kódot használnak, amely a feltétlenül szükséges információ tartalom mellett bizonyos mértékű többlettel, redundanciával rendelkezik. Ekkor viszont a kódszavak hosszabbak lesznek és emiatt a berendezések nyilvánvalóan bonyolultabbak, drágábbak kell hogy legyenek, vagyis a hiba elleni védelemnek komoly ára van. (Ha nem a kódot bővítjük többlet bitekkel, akkor a kódátvivő rendszernek kell a

védelmet megoldania pl. a vett üzenet visszaküldésével és az adóban az egyezés ellenőrzésével stb.)

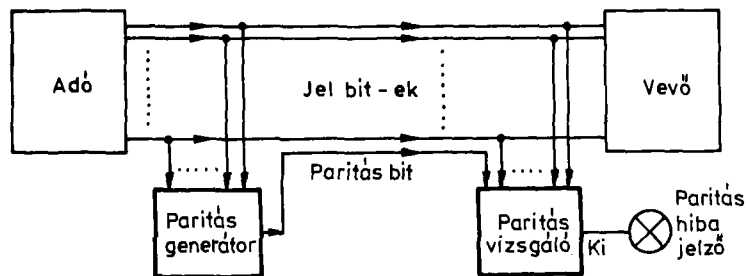
A hiba elleni védelem, hibakorlátozás egyszerűbbik módja az átvitelkor, feldolgozáskor keletkezett bit-hiba jelzése. Ehhez olyan kód kell, amelynek egy-egy kódszaváról valamilyen véges valószínűséggel megállapítható, hogy van-e benne hiba vagy nincs. Egy bináris számot például hiába vizsgálgatunk akármeddig, nem jövünk rá, hogy mialatt hozzánk megérkezett, keletkezett-e benne hiba vagy sem, hiszen bármilyen kombináció is érkezik, az megengedett kódszó. A hibát csak akkor vesszük észre, ha a kód rendszere olyan, hogy a "hasznos", megengedett kódszavak nem merítik ki az összes lehetőséget, megengedett kombinációk is lehetségesek, melyek nyilván csak akkor jelennek meg, ha a rendszerben hiba van.

A hiba akkor deríthető fel egyáltalán, ha a kódszó készletben bármely párosításban a HAMMING távolság 1-nél nagyobb (vagyis a kód Hamming távolsága 1-nél nagyobb). Ha maga a kód nem ilyen (mint pl. a bináris, BCD, Gray, stb.), akkor a távolságot "mesterségesen" növelni kell, többlet bit(ek) hozzáadásával vagyis a megengedett kódszavak közé szisztematikusan meg nem engedett kódszavakat kell "ékelni". Ezt tesszük, amikor valamely kód szavaihoz paritás bitet adunk: minden olyan kódszót tiltunk, amelynek a paritása páratlan, minden olyan szót megengedünk, amelynek paritása páros, ezzel bármely kódszó bármelyikhez hasonlított távolságát 2-re növeltük, aminek segítségével 1 bit hibát észlelhetünk (lásd ASCII, SELECTRIC). Ha pl. a BCD kódot egy paritás bittel kiegészítjük, akkor hibajelzésre alkalmassá tesszük:

P	D	C	B	A	P	D	C	B	A
0	0	0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	1	1	0
1	0	0	1	0	1	0	1	1	1
0	0	0	1	1	1	1	0	0	0
1	0	1	0	0	0	1	0	0	1

A paritás hibajelzéssel megvalósított párhuzamos kódátvitel vázlatát mutatja az 1.22. ábra. A PARITÁS GENERÁTOR "figyeli" az adó kimeneteit és előállítja a paritás bitet, amivel kie-

gészíti az átviendő jelet. A vevő oldalon a PARITÁS VIZSGÁLÓ (PARITY CHECKER) megvizsgálja a paritás bittel kiegészített információt és ha nem páros (vagy páratlan) számú 1-es van a jelben, akkor a kimenetén jelzést ad.



1.22. ábra

A PARITÁS GENERÁTOR és a PARITÁS VIZSGÁLÓ gyakorlatilag egyforma áramkörök, hiszen a vizsgáló is képez egy paritás bitet, amely csak akkor lesz 1, ha a kijövő jel paritása nem megfelelő, tehát, ha hiba van. A paritás bit képzése az átvitel nélküli összeadás (moduló-2 összeadás) elvén megy végbe: a bit-jegyeket egyszerűen (átvitel nélkül) összeadjuk az eredmény a páros paritás bit. Pl.:

0 0 0 1 -hez a paritás bit: $0 \oplus 0 \oplus 0 \oplus 1 = 1$
 0 1 1 0 $0 \oplus 1 \oplus 1 \oplus 0 = 0$
 1 1 1 0 $1 \oplus 1 \oplus 1 \oplus 0 = 1$ stb.

Ha egynél több bit hibás, akkor a paritáselemes kódok "nem kötelesek" a hibát jelezni. 2 bit hibánál a paritás nem hibás, 3 bit hibánál megint hibás, stb., vagyis a paritás bittel kiegészített kódok páratlan számú hibát jeleznek (csak sajnos arra nincs semmi biztosíték, hogy páratlan számú hiba következik be). Látható, hogy a hibajelzés csak bizonyos véges valószínűséggel végezhető. Az alfanumerikus kódok közül is sok olyan van, amely tartalmaz paritás bitet - ahogyan ezt az előzőkben már láttuk.

A hibajavításhoz - érthető módon - még több "beépített redundanciára", a megengedett kódszavak közé "ékelt" meg nem engedett kódszóra van szükség. A kód megfelelő felépítésével

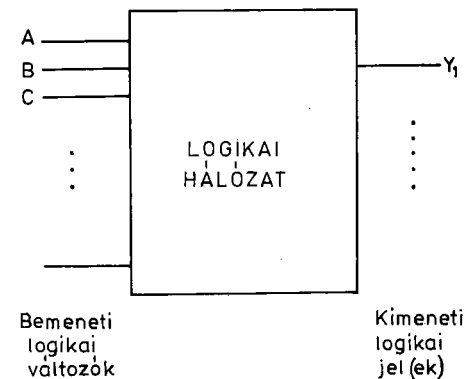
elérhető az is, hogy adott szabályok szerint elvégzett vizsgálatok után kiderüljön, hogy a kérdéses kódszóban melyik bit a hibás és akkor ez a hiba automatikusan ki is javítható. A többlet redundancia igény természetesen azt jelenti, hogy az elengedhetetlenül szükségesnél jóval több bitet kell előállítanunk, továbbítanunk, feldolgoznunk, ami a költségek rohamos növekedését vonja maga után. A jelfeldolgozásban alkalmazott hibakorlátozási (hibajelzési és hibajavítási) módszerekkel részletesen majd egy későbbi tantárgyban foglalkozunk.

2. KOMBINÁCIÓS LOGIKAI HÁLÓZATOK

2.1. A logikai hálózatok modellje

A digitális (vagyis számjegyekkel dolgozó) berendezések alapvető alkotó elemei a logikai hálózatok, amelyek közül mi a villamos változatokkal, a logikai áramkörökkel foglalkozunk. A "logikai" elnevezés azért szokásos, mert a logikai áramkörök működésének törvényszerűségei, szabályai és a formális logika között igen nagy hasonlóság van. (A formális logika az emberi gondolkodás formai törvényeivel foglalkozó tudomány.)

A logikus gondolkodás során az alaptételekből, állításokból, itéletekből, általánosságban premisszákból logikai műveletek végzésével vagyis a premisszák logikai kapcsolatainak feltárásával valamilyen következtetésre (konklúzióra) jutunk. A következtetés alapjául szolgáló, általában kijelentő módban megfogalmazott állítások, premisszák egy adott időben vagy IGAZAK, vagy HAMISAK (harmadik lehetőség nincs). Ugyanigy a következtetések is vagy IGAZ, vagy HAMIS értékűek lehetnek. Ez a kétféle igazságtartalom adja meg a lehetőségét annak, hogy a formális logika törvényeit követő "gondolkodó" logikai hálózatokat, akár "intelligens (értelmes)" szerkezeteket hozunk létre, hiszen tudjuk, hogy ezek is kétféle jellel dolgoznak. Az IGAZ (TRUE) tartalomnak a logikai 1-et, a HAMIS (FALSE)-nak a logikai 0-át feleltetjük meg és olyan áramköröket építünk, amelyek a 0 vagy 1 értékű premisszák között a logikus gondolkodásban is előforduló logikai kapcsolatokat tudják létrehozni és ennek alapján jutnak valamilyen 0 vagy 1 értékű következtetésre. A logikai hálózatok a bonyolultabb logikai kapcsolatokat mindig egyszerű, elemi logikai alapműveletekből állítják elő. Ezeket az alapműveleteket a következőkben részletesen megismerjük.



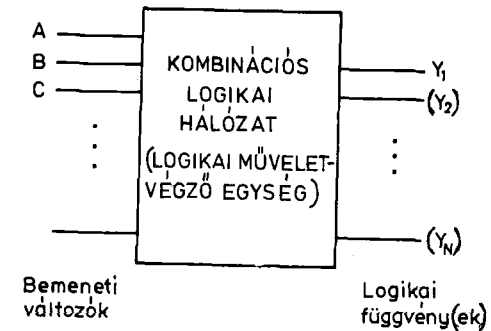
2.1. ábra

Az előbbieken alapján a logikai hálózatok egészen leegyszerűsített modelljét a 2.1. ábrán látható módon vázolhatjuk fel. A hálózat bemenetére érkező jelek a bemeneti logikai változók, ezeket szokásosan A, B, C,....-vel, az ábécé nagybetűivel vagy $X_1 X_2 X_3 \dots$ -el jelöljük. E bemeneti változók jelképezik azokat a premisszákat, amelyeket egy-egy kijelentő mondat fogalmazhatunk meg, más szóval valamely esemény bekövetkezését jelentik és amelyek igazságtartalma, "értéke" 0 vagy 1 lehet. Például: "Záródott az A kapcsoló", "A mozgó gépkatrész helyzetérzékelője szélső helyzetet jelez", "A meghajtó motor forog" stb. Ezeket az "értesüléseket", amelyek már villamos, logikai jelek formájában jutnak a bemenetekre, a logikai áramkör "feldolgozza", műveleteket végez azokkal, és végül a változók értékétől függően a kimenetén (kimenetein) valamilyen "döntést hoz", konklúzióra jut vagyis az Y-nal jelölt kimeneti vezetéken (vezetékeken) 0 vagy 1 jelszintet (jelszinteket) állít elő (és például leállítja a meghajtó motort, jelzéseket ad, stb.). Amikor az áramkört analizáljuk vagy tervezünk, már általában nem törődünk a bemeneti változók fizikai jelentésével, csak igazságtartalmával és a közöttük levő logikai kapcsolatokkal, a következtetéssel és természetesen az eszközként felhasznált matematikai-logikai módszerrel, szoftverrel stb., és csak e munka végeztével térünk vissza a "fizikai valósághoz".

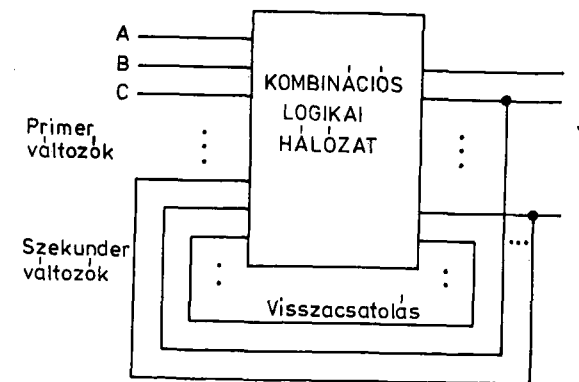
A logikai hálózatokról alkotott képünket azonban még finomítani kell. Figyelembe kell vennünk például azt a tényt, hogy a digitális áramkörök nagy része a kimeneti Y értékeket nemcsak az éppen abban a pillanatban a bemenetre érkező jelekből állítja elő, hanem figyelembe veszi az előző vezérlési állapotokat is, más szóval "emlékezik az előéletére" is. Ehhez a hálózatban "emlékező": tároló, memória elemeknek kell lenniük. Minőségi ugrást jelent, amikor a hálózat a pillanatnyi vezérlésektől és a tárolt "megelőző" információktól függően "önálló döntéseket" is hoz, több lehetőség között választ, sőt, vagyis "intelligens" (természetesen ember által készített program segítségével). Ezek szerint a logikai hálózatokat többféle szempontból többféle csoportba oszthatjuk, de egy biztos: a bonyolultabb áramkörök (rendszerek) mindig egyszerűbb, alapvető funkcionális egységekből épülnek fel, nemegyszer ezekből több ezret tartalmazva (ami természetesen a működésben minőségi különbséget is jelent). Legegyszerűbb áramkörtípus a KOMBINÁCIÓS LOGIKAI ÁRAMKÖR, amely a "nem intelligens" kategóriába sorolható és amely a bemenetekre érkező jelek között azonnal elvégzi a "logikai műveleteket", ezek eredményét az áramkörre jellemző késleltetési idő elteltével a kimenet(ek)re adja. Vázlata lényegében megegyezik a 2.1. ábrán látható általános modellel, a lényeg a működésben van: ha a bemeneti változók értékét megváltoztatjuk, akkor ezt az előállított kimeneti jelek értékváltozása azonnal (a késleltetési idő elteltével) követi, függetlenül attól, hogy a hálózat előzőleg milyen állapotokat vett fel. A kombinációs hálózat tehát egy "logikai műveletvégző egység"-nek tekinthető, amely a bemeneti változók 0-1 érték kombinációihoz Y kimeneti 0-1 értékeket rendel aszerint, hogy a változók között milyen kapcsolatot, továbbiakban logikai függvényt kell létrehozni (2.2. ábra). A kombinációs hálózatok többnyire kapu-áramköröket tartalmaznak (l. később), ezért szokásos a "kapu-hálózat" elnevezés is.

Az egyszerű, szintén alap építőelemként használt "nem intelligens" SORRENDI LOGIKAI ÁRAMKÖRÖK a bemenetekre érkező jeleken kívül figyelembe veszik az előzőleg felvett állapotokat is, ami úgy lehetséges, hogy a kimeneteken megjelenő je-

leket vagy azok közül valamennyit visszacsatolunk a bemenetre, ezek "értékesítik" a hálózatot az előző állapotról (2.3. ábra).



2.2. ábra

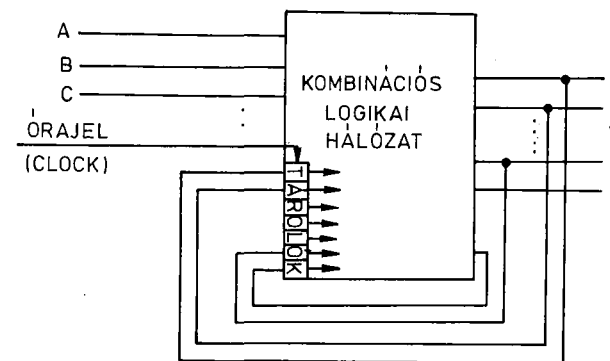


2.3. ábra

A "dobozban" most is kombinációs logikai (kapu) hálózat van, általában azzal a kikötéssel, hogy az áramköröknek a bemenettől a kimenetig 1-nél nagyobb erősítéssel kell rendelkezniük vagyis a visszacsatolásnak nemcsak logikainak, hanem áramköri visszacsatolásnak is kell lennie. Működés közben, ha valamilyik bemenet jelét (A, B, C, "primer változó" értékét) megváltoztatjuk, akkor ezek hatására a visszacsatolt jelek ("szekunder változók") függvényében megváltoznak a kimeneti Y jelek. Ez utóbbiak értékváltozása visszakerülve a bemenetre újabb kimeneti jelváltozást idézhet elő - és így tovább - mindaddig, amíg a hálózat stabil állapotba nem jut (hacsak nem astabil az áramkör vagyis oszcillátor). Belátható, hogy az ilyenfajta hálózatok működése "rendezetlen", különféle logikai állapotváltozások egymás után, nem egyidőben zajlanak le, ezért nevezzük ezeket aszinkron sorrendi logikai áramköröknek. Az aszinkron jelleg természetesen nem jelenti azt, hogy az ilyen hálózat használhatatlan, az alapáramkörök (például bistabil billenőkörök, flip-flop-ok) és a bonyolultabb áramkörök (pl. számlálók) között sokféle aszinkron változat van, amelyet igen gyakran használunk, főleg egyszerű felépítésük miatt. Az aszinkron logikai hálózatok "emlékező", az előzőleg felvett állapotot figyelembe vevő tulajdonságát vagyis a tárolási funkciót a kimenetről a bemenetre menő visszacsatolás hozza létre.

A szinkron sorrendi áramkörök is visszacsatolással működnek, rájuk is illik a 2.3. ábra vázlata. A lényeges különbség az, hogy a kimenetről a bemenetre visszacsatolt jelek nem azonnal fejtik ki hatásukat, hanem egy speciális bemeneti jelre, az ütemjelre vagy órajelre (CLOCK, Cp: Clock pulse) "várnak", majd ennek megérkezése után a bemeneten erre a célra felépített tároló-sorba íródnak. Ezek a tárolt jelek "emlékeztetik" a hálózatot az előző állapotára és teszik lehetővé az újabb kimeneti jelek létrehozását, de a megváltozott kimeneti jelek visszacsatolás utáni hatása nem érvényesül azonnal, hanem csak a következő órajel megérkezésekor. A szinkron sorrendi hálózatban így minden változás az órajel által meghatározott pillanatban az órajellel időzítve, az órajellel szinkronizálva jön létre (2.4. ábra). Ez az egyidejűség nagyon sok-

szor előnyös; minden "esemény" előre pontosan definiált időpillanatban történik, mégpedig a legtöbb digitális rendszerben végigmenő és jelenlevő órajel fel- vagy lefutó élének megérkezését követően igen kis "idő-tűrésmezőben"! Ennek azonban ára van: a szinkron hálózatok az esetek többségében bonyolultabb felépítésűek és ezért drágábbak, mint az aszinkron változatok - a mérlegelés és a megfelelő megoldás, típus kiválasztása az áramkör-tervező dolga.



2.4. ábra

Emlitettük már, hogy a bonyolultabb (összetett kombinációs, sorrendi, sőt az "intelligens") áramköri egységek mindig egyszerűbb, alapvető funkcionális egységekből épülnek fel, ezért elsőknek ezekkel az alapvető kombinációs áramkörökkel és az ezek analíziséhez, tervezéséhez szükséges alapfogalmakkal foglalkozunk.

A logikai feladatok megoldásához George BOOLE (1815-1864) hozott létre olyan kétértékű algebrát, amely a bonyolultabb logikai kapcsolatok egyszerű kapcsolatok segítségével való leírását, egyszerűsítését teszi lehetővé. A BOOLE-algebrát kb. az 1930-as évek végén kezdték alkalmazni kapcsolóáramkörök tervezéséhez, és mint tudjuk, ma is ez a legalapvetőbb, legfontosabb eszköz a logikai áramkörök analíziséhez és szintéziséhez.

BOOLE alaptétele az, hogy bármilyen bonyolult logikai kapcsolat megfelelően választott alapműveletek segítségével kifejezhető, azok segítségével összeállítható, éppen ez teszi

alkalmassá a BOOLE-algebrát bonyolult áramkörök matematikai tárgyalására, tervezési segédeszközként való felhasználására.

A továbbiakban áttekintjük a logikai "alapműveleteket", azokat, amelyeket a BOOLE-algebra használ a tetszőlegesen bonyolult logikai kapcsolatok leírására. Először a negáció, majd a VAGY kapcsolat, valamint az ÉS kapcsolat értelmezését, jelöléseit, megvalósításait nézzük meg. Az alapműveletek áttekintése után összefoglaljuk a BOOLE-algebra alaptételeit és szabályait. Ezután felírjuk az összes lehetséges 2 változós logikai kapcsolatot (logikai függvényt), mindegyiket kifejezzük az alapműveletek segítségével, végül a logikai kapcsolatokat kettőnél több változóra is kiterjesztjük, értelmezzük.

2.2. Logikai alapműveletek

2.2.1. Negáció, tagadás, invertálás

A negáció, tagadás vagy invertálás azt jelenti, hogy valamely esemény, logikai változó, vagy akár valamely következtetés, logikai függvény igazságtartalmát ellenkezőjére változtatjuk. Az igazságtartalomhoz rendelt 0-t 1-re vagy 1-et 0-ra cseréljük, vagyis általában egy logikai jel 0 és 1 értékét felcseréljük.

Az "értéktáblázat", igazságtáblázat nagyon egyszerű: ha az A változó logikai 0 értékű, akkor a negáció eredménye 1-es értékű, ha az A értéke 1, akkor az eredmény 0 (2.5a ábra).

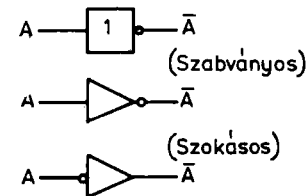
A tagadott változót vagy kifejezést szokás szerint "negált"-nak nevezzük (pl.: \bar{A} : "á-negált"). Előfordulhat, hogy valamely változó nem negált, hanem "valóságos" értékével kívánunk számolni, akkor erre a "ponált" kifejezéssel hívhatjuk fel a figyelmet (A: "á-ponált"). A negáció szokásos műveleti jele a felülhúzás: $Y = \bar{A}$ vagy a külön "negált jel": $Y = \bar{A}$.

Logikai áramkörökben a negálást INVERTER végzi. A kapcsolási rajzokon használatos jelöléseket a 2.5b ábrán láthatjuk. Magát az invertálást a kis kör jelképezi. Ez azért lényeges, mert lehetnek olyan logikai hálózatok, "dobozok", amelyek inverter kimenetűek, azaz a kimeneti függvény negáltját, inver-

zét állítják elő, ilyenkor azt is kis körrel jelölik (pl. egy "aktív 0" kimenetű dekódoló, amellyel később találkozni fogunk). Ugyanigy jelölik, ha valamely doboz bemenetét kell inverz jellel vezérelni; ilyenkor a bemeneti vezeték csatlakozik null-körrel a dobozhoz.

Az IGAZSÁGTÁBLÁZAT: Az INVERTER rajzjele:

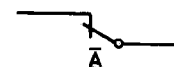
A	Y = \bar{A}
0	1
1	0



a)

b)

Érintkezős megvalósítás: KARNAUGH-tábla:



c)

A	
0	(\bar{A}) (A=0)
1	(A) (A=1)

d)

2.5. ábra

A logikai jelek invertálásához mindig vezérelhető, aktív (erősítő) elemre van szükség. Gondoljunk pl. a földelt emitteres erősítő fokozatra. Ennek 180°-os fázisfordítása alkalmas arra, hogy a logikai 0 feszültséget 1-re, a logikai 1 feszültséget 0-ra cserélje (részletesen 1. később). Aktív elem a jelfogó is: ha az A jelre a jelfogó meghuz, akkor az \bar{A} -val jelölt nyugalmi érintkezőt (nyugalomban záró, meghuzva bontó érintkezőt) megszakítja, a rajta átfolyó áram zérus lesz, előállt az A jel negáltja (2.5c ábra).

Felrajzoltuk a negáció KARNAUGH TÁBLÁ-ját is. Egy független A változó - mint a logikában egy ítélet általában - azonos valószínűséggel lehet "IGAZ" vagy "HAMIS", ill. 1 vagy 0 értékű, ezért az egységnyi területet 2 egyenlő részre osztottuk fel a diagramban. Az alsó fél az A változó területe, itt

A = 1, amelyet a területrészt melletti jelölés mutat. A felső területrészt az A negáltjának területe, tehát itt az \bar{A} igaz, az A = 0. A Karnaugh-táblát - ami lényegében nem más, mint egy "átrendezett" igazságtábla - nagyon jól fel lehet használni a logikai függvények szemléletes, grafikus minimalizálásához, ahogyan ezt a későbbiekben látni fogjuk. Több változó esetén a Karnaugh-tábla is összetettebb és sokkal több információt tartalmaz, mint ebben a legegyszerűbb esetben.

2.2.2. A logikai VAGY kapcsolat, diszjunkció

A VAGY kapcsolat jelentése: ha a logikai változók közül akár egyetlen egy is 1-es értékű, az eredmény, a függvényérték "már" 1-es lesz. Ha egyszerre több változó 1-es, a függvényérték akkor is 1-es. A függvényérték csak akkor zérus, ha valamennyi változó zérus értékű. Az így definiált alapművelet eltér a "hétköznapi" értelemben használt "VAGY"-tól, hiszen a logikai VAGY esetében az eredmény akkor is 1, ha több változó 1-es ("megengedő-VAGY"). Két változóra az igazságtáblázatot a 2.6a ábrán láthatjuk. Itt most 2 változó összes lehetséges érték-kombinációjára kell a függvényt megadnunk vagyis $2^n = 2^2 = 4$ esetet kell felsorolnunk. A felsorolást valamilyen rendszer szerint célszerű végrehajtani; az A és B oszlopba a 0-kat és 1-eket úgy írjuk be, mintha bináris számok lennének, amelyeket a bináris 0-tól növekvő sorrendben írjuk fel, amíg a legnagyobb számig, a "csupa 1-es-ig" el nem érkezünk. Így biztos, hogy valamennyi eset előfordul. A továbbiakban az igazságtáblázatokot mindig ezzel a módszerrel készítjük. A VAGY logikai függvény három esetben ad 1-es, egy esetben zérus értéket.

A VAGY kapcsolat műveleti jele v jel vagy az algebrában megszokott + jel. A VAGY műveletet szokás logikai összeadás-nak is nevezni (diszjunkció).

Azt az egységet, amely a változók között a VAGY kapcsolatot létrehozza VAGY-KAPU-nak (OR-GATE-nek) nevezzük. Szabványos, valamint az elfogadott, szokásos kapcsolási rajzjeleit a 2.6c ábrán láthatjuk. A kapuk áramköri megvalósításáról

a későbbiekben részletesen szó lesz. Az érintkezőkkel megvalósított VAGY kapcsolást felrajzoltuk a 2.6d ábrán. Mivel az A és B munkaérintkezők (meghúzásra bekapcsoló érintkezők) párhuzamos kapcsolásban vannak, akár az A, akár a B, vagy akár mindkét kapcsoló egyszerre zár, az áramkör vezet az áramot, az eredmény logikai 1 ($Y = 1$). Szakadás, logikai 0 ($Y = 0$) csak akkor lesz, ha mindkét érintkező nyitott vagyis $A = 0$ és $B = 0$.

Az igazságtáblázat:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

a)

Műveleti jel:

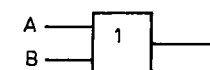
$$Y = A \vee B$$

vagy

$$Y = A + B$$

b)

VAGY KAPU (OR GATE):



(Szabványos rajzjel)

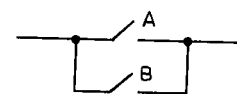


(Szokásos rajzjelek)



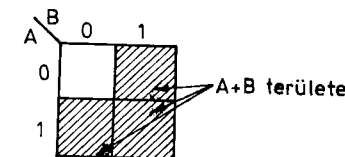
c)

Erintkezős megvalósítás:



d)

KARNAUGH-tábla:



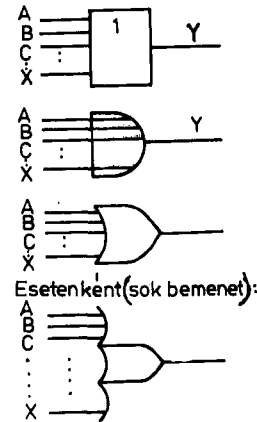
e)

2.6. ábra

A Karnaugh-diagramban - amelyet most két változóra rajzoltunk meg, a táblát vízszintesen és függőlegesen is két részre osztva, összesen tehát négy rekeszt kijelölve - az A + B-nak megfelelő terület magába foglalja a teljes A területet (ahol A = 1, az egység négyzet alsó 1/2 részén) és a teljes B területet (a négyzet jobb oldali 1/2 részén), vagyis azt a teljes részt, amelyben vagy A, vagy B jelen van (vagy A = 1, vagy B = 1).

A VAGY kapcsolat egyszerűen értelmezhető több változóra is; tetszőleges számú bemeneti változó esetén a kimenet "már" akkor 1 állapotú lesz, ha bármelyik bemeneten 1-es van. A kimenet csak akkor 0 állapotú, ha valamennyi bemeneten 0 van (2.7. ábra).

A	B	C	...	X	Y = A+B+C+...+X
0	0	0	...	0	0
0	0	0	...	1	1
.
.
1	1	1	...	1	1



2.7. ábra

2.2.3. A logikai ÉS művelet, konjunkció

Az ÉS kapcsolat eredménye akkor 1, ha az összes változó értéke egyidejűleg logikai 1 - egyezően a "hétköznapi" értelemben használt ÉS-sel. A kétváltozós igazságtáblázatból (2.8a ábra) is látszik: az ÉS állítás akkor igaz (1), ha az A esemény ÉS a B esemény is igaz, minden más esetben hamis (0). Az ÉS kapcsolat, konjunkció jelölése a változók közé irt \wedge jel vagy a szorzó pont, és megengedett az algebraiban szokásos rövidítés is, amellyel a változókat egyszerűen egymás mellé írjuk. Az ÉS műveletet szokás logikai szorzás-nak is nevezni (konjunkció).

A műveletet megvalósító egység az ÉS-KAPU (AND-GATE), amelynek szabványos, valamint szokásos, általánosan használt egyéb rajzjeleit a 2.8c ábrán láthatjuk.

Érintkezős hálózattal az ÉS kapcsolatot sorba kötött munkaérintkezők segítségével valósíthatjuk meg. Áram akkor folyhat ($Y = 1$), ha az A ÉS a B érintkező is zárt, ha bármelyik nyitott, akkor a lánc megszakad (2.8d ábra).

Az igazságtáblázat:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

a)

Műveleti jel:

$$Y = A \wedge B$$

vagy

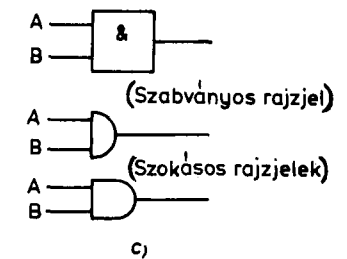
$$Y = A \cdot B$$

vagy

$$Y = AB$$

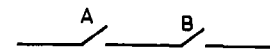
b)

ÉS KAPU (AND-GATE):



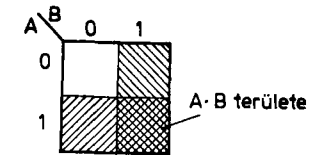
c)

Érintkezős megvalósítás:



d)

KARNAUGH-tábla:



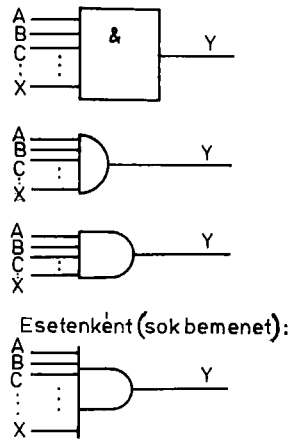
e)

2.8. ábra

A KARNAUGH-diagramon az $A \cdot B$ területet az A ÉS B terület közös része adja, ahol A is és B is jelen van (a 2.8e ábrán kétszeresen vonalkázott terület). Érdemes megfigyelni, hogy az $A \cdot B$ terület a KARNAUGH-diagramon minimális helyet foglal el (hiszen ennél az 1/4 résznél kisebb egység két változó esetén nincs).

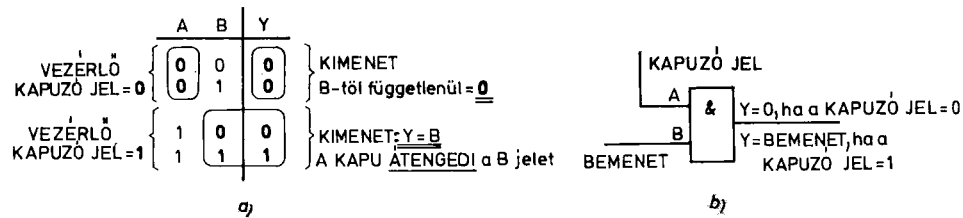
A több változóra való kibővítés ebben az esetben is egyszerű: tetszőleges számú változó esetén a függvényérték csak akkor 1-es, ha valamennyi bemeneti változó értéke 1-es; ha a bemeneti változók közül akár csak egyetlenegy 0 értékű, akkor a kimenet is 0 értékű lesz (l. a 2.9. ábra igazságtáblázatát és több bemenetű ÉS kapuját).

A	B	C	...	X	Y = ABC...X
0	0	0	...	0	0
0	0	0	...	1	0
.	0
.	0
.	0
.	0
1	1	1	.	1	1



2.9. ábra

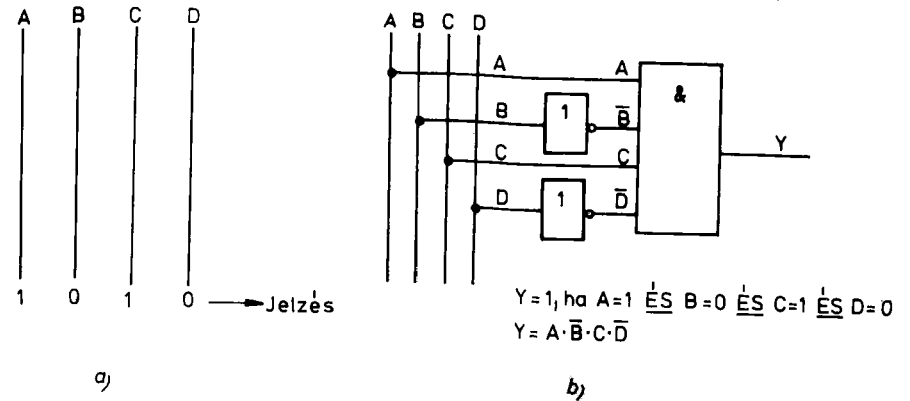
Érdeemes már most, digitális áramkörökkel kapcsolatos tanulmányunk elején megismerkedni a logikai kapuk alapvető áramköri funkciójával, leggyakoribb alkalmazási példájával: magával a "kapuzással". A kétváltozós ÉS kapcsolat igazságtábláját úgy is értelmezhetjük, hogy az ÉS-kapu akkor "engedi át" az egyik bemenet jelét, ha a másik (mondjuk "vezérlő" vagy "kapuzó") bemenete 1-en van, máskülönben letiltja ezt a bemeneti jelet.



2.10. ábra

A 2.10a ábrán újra rajzoltuk az igazságtáblát, most már "ebben a felfogásban": ha az A vezérlő, kapuzó jel 0 értékű, akkor a kimeneten is 0 jelenik meg a másik B bemenet jelétől függetlenül, azaz a kapu nem engedi át a B jelet (l. az igazságtábla első két sorát!). A vezérlő, kapuzó jel 1-es értéké-

nél viszont a kimenet jele azonos a B bemenet jelével, a kapu átengedi a B jelet (második két sor). Az ÉS kapu ily módon 1 bit átengedésére, ill. letiltására alkalmas, "elektronikus vezérléssel" (2.10b ábra). Természetesen az A és B jel szerepet cserélhet, a működés azonos. A VAGY kapu is alkalmas jel kapuzásra abban az esetben, ha a logikai 0 jelet tekintjük "értékesnek" (un. "aktív 0" rendszerben - a bizonyítást az olvasóra bizzuk).



2.11. ábra

Hasonlóan fontos, tipikus kapuzási feladat valamely előre meghatározott jelkombináció "figyelése", kiválasztása a többi közül. Tegyük fel például, hogy 4 vezeték (A, B, C, D) jelét kell figyelni és akkor kell jelzést adnunk (logikai 1-gyel), amikor a vezetéseken éppen rendre 1 0 1 0 értékkombináció áll elő (azaz A=1, B=0, C=1, D=0 - 2.11a ábra). A feladatot ÉS kapuval oldhatjuk meg, felhasználva azt a tulajdonságát, hogy kimenetén akkor és csak akkor jelenik meg 1-es, amikor minden egyes bemenete 1-es szinten van. Ezt az "esetet" kell megvalósítanunk az adott értékkombinációra. Tekintve, hogy a kiválasztani kívánt kombinációban a B és a D jel 0 értékű, viszont az ÉS kapu bemeneteire éppen ebben az esetben csupa 1-esnek kell jutnia, a B és D jelet invertálnunk kell, vagyis az A jel ponáltját, a B jel negáltját, a C jel ponáltját, valamint a D jel negáltját kell a 4 bemenetű ÉS kapura vezetni (a negáláshoz invertert használunk, 2.11b ábra). Az

ÉS kapu Y kimenetén így éppen az előírt esetben áll elő 1-es, minden más esetben valamelyik (vagy több) bemeneten 0 lesz, ezért a kimeneten is 0 lesz, "kikapuztuk" az 1010 kombinációt. Magától értetődő, hogy az előállított ÉS függvény:

$$Y = A \cdot \bar{B} \cdot C \cdot \bar{D}$$

formában írható le. Az ÉS kapu tehát alkalmas arra, hogy egy adott logikai feltétel teljesülése esetén jelzést adjon. Ilyen kapuzást alkalmazunk pl. számlálókhöz csatlakozó áramkörökben egy adott szám "felismerésére", de ugyanilyen elven választhatunk ki egy memória, periféria stb. címet egy mikroszámítógép rendszerben is. (Érdemes megjegyezni, hogy "aktiv 0" rendszerben a VAGY-kapu ugyanilyen elven használható egy adott bit-kombináció "kikapuzására".)

2.2.4. A BOOLE-algebra alaptételei, szabályai

A BOOLE-algebra alapműveletei - a NEGÁCIÓ, az ÉS, valamint a VAGY kapcsolat - áttekintése után, röviden összefoglaljuk ezen alapműveletekre vonatkozó szabályokat, tételeket, amelyek általában ismertek, "maguktól értetődőek" és a BOOLE-algebra axiómáit fogalmazzák meg vagy belőlük következnek és amelyeket tudatosan, vagy ösztönösen a digitális technikában nap mint nap felhasználunk. Nem tekintjük célunknak a teljes axióma-rendszer precíz tárgyalását csak a gyakorlatban is felhasznált azonosságokat, szabályokat, "következményeket" soroljuk fel. A csoportosítás is elsősorban a gyakorlati felhasználásnak megfelelő és nem követ feltétlenül logikai sorrendet. A könnyebb megértés érdekében a legtöbb azonossághoz ábrát is rajzolunk: érintkezős megvalósítást vagy KARNAUGH-táblát, ezzel, ha nem is igazoljuk, de "kisérletileg" meggyőződünk az egyenlőségről.

- A tagadás törvénye:

$$\bar{\bar{0}} = 1 \quad \bar{\bar{1}} = 0$$

- A kettős tagadás törvénye:

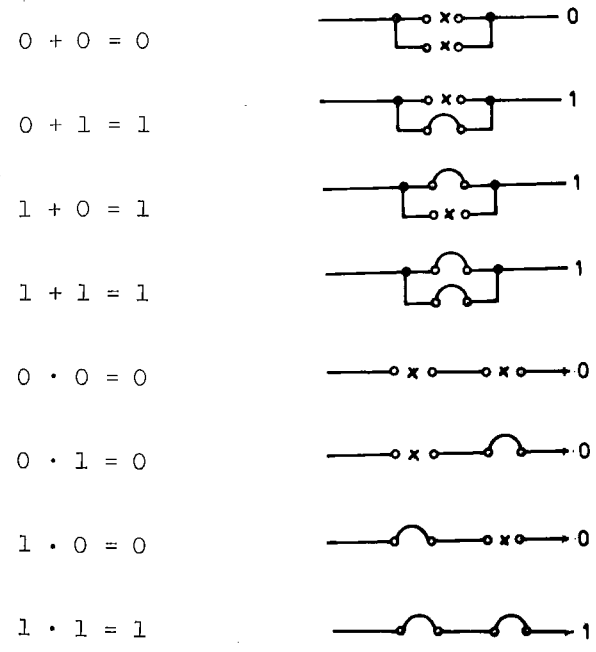
$$\bar{\bar{0}} = 0$$

$$\bar{\bar{1}} = 1$$

$$\bar{\bar{A}} = A$$

Egy logikai jelet vagy változót (eseményt) kétszer tagadva az eredeti érték (esemény) igaz. Általában páros számú tagadás ponált, páratlan számú tagadás negált értéket eredményez. (A magyar nyelv ezen a téren nem követi a formális logikát, pl. "A nincs semmi olvasni valóm" állítás "logikus" értelme ez lenne: "Van olvasni valóm".)

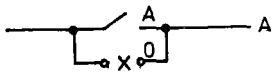
- A logikai 1 és logikai 0 VAGY, ill. ÉS kapcsolata. (Érintkezős hálózatban a 0-nak állandó szakadás, az 1-nek rövidzár felel meg, 2.12. ábra.)



2.12. ábra

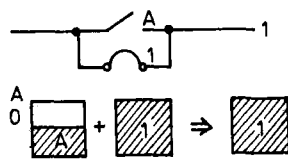
- Változó és állandó érték logikai kapcsolata.

$$A + 0 = A$$



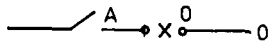
Ha egy változóhoz vagy kifejezéshez zérust "hozzáadunk", akkor az eredeti érték nem változik (párhuzamos szakadás nem változtat a működésen).

$$A + 1 = 1$$



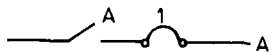
Bármely változóhoz kifejezéshez 1-et "hozzáadva" (logikai összeadással) az eredmény azonosan 1 lesz (párhuzamos rövidzár rövidzárt eredményez).

$$A \cdot 0 = 0$$



Ha a logikai szorzat egyik tényezője zérus, akkor az eredmény is zérus (soros szakadás szakadást eredményez).

$$A \cdot 1 = A$$

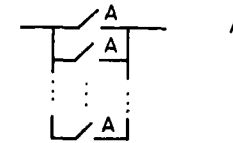


1-gyel való logikai szorzás nem változtat az eredményen (soros rövidzár nem befolyásolja az érintkező hálózat működését).

2.13. ábra

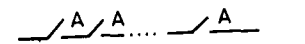
- Műveletek egyazon változóval.

$$A + A + \dots + A = A$$



(Egyszerre működő párhuzamos érintkezők ugyanazt a feladatot látják el, mint egyetlen egy).

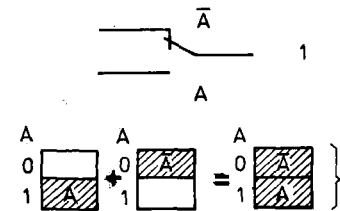
$$A \cdot A \dots A = A$$



(Minden érintkező egyszerre A-ra záródik, akkor a soros áramkör is A-ra záródik.)

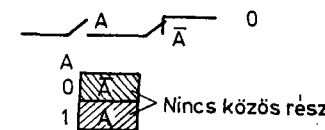
E két eset azt a szabályt írja le, hogy az ÉS-VAGY függvény ismétléssel tetszőlegesen bővíthető (ill. ismétlés esetén egyszerűsíthető) - a függvény nem változik.

$$A + \bar{A} = 1$$



Vagy a változó, vagy a negáltja mindenképpen 1-et ad (egy állítás vagy a tagadása biztosan igaz).

$$A \cdot \bar{A} = 0$$



Nem lehet egyszerre igaz egy állítás és a tagadása.

2.14. ábra

- Kommutatív tulajdonság ("felcserélhetőség").

Az ÉS és VAGY művelet elemei (tényezői, ill. tagjai) felcserélhetők (sorrendjük nem befolyásolja a függvényértéket):

- a) $A + B = B + A,$
- b) $AB = BA.$

- Asszociatív tulajdonság ("társíthatóság").

- a) $A + (B+C) = (A+B) + C = (A+C) + B = A+B+C,$
- b) $A(BC) = (AB)C = (AC)B = ABC,$

vagyis azonos művelet végzésekor a sorrendiség nem játszik szerepet, a zárójel *feleslegesek*.

- Disztributív tulajdonságok.

a) $A(B+C) = AB + AC$

- a logikai szorzás tagonként végrehajtható (a zárójel beszorozható),

b) $A+(BC) = A+BC = (A+B)(A+C)$

- a logikai összeadás tényezőnként hajtható végre (a zárójel beszorzás szimmetrikus művelete).

Az a) és a b) azonosság közötti hasonlóság egyik példája a logikai szorzás és a logikai összeg teljes "szimetriájának" dualitásának; a szabályok mindkét műveletre "egyformák". A szimmetria ellenére a BOOLE-kifejezésekben a zárójelet a "normál" algebránál megszokott módon használjuk; "elsődlegesnek", "magasabbrendűnek" a logikai szorzást tekintjük, először mindig a szorzást végezzük el, azután a logikai összeadást. Pl.: az $AB + AC$ kifejezést "automatikusan" úgy értelmezzük, hogy az AB és az AC szorzatot kell összeadnunk, nem pedig úgy, hogy például az A -t szorozzuk a $B+A$ -val és ezt végül a C -vel. Ha egyéb sorrendet írunk elő, akkor kell záróje-

let kitenünk, mindig úgy, hogy a műveletek kijelölése egyértelmű legyen. Például az $A(B+C)$ kifejezésben a zárójel azt jelenti, hogy a $B+C$ összeget kell A -val szorozni, nem pedig az AB szorzattal C -t összegezni, ami zárójel nélkül magától értetődő lenne.

A b) azonosságot a következőképpen bizonyíthatjuk:

$$(A+B) \cdot (A+C) = A+AB+AC+BC = A(1+B+C) + BC = \underline{A + BC}$$

(mert $1+B+C = 1$).

- Abszorpciós (elnyelési) tételek

a) $A+AB+ABC+\dots = A$

mert:

$$A+AB+ABC+\dots = A(\underbrace{1+B+BC+\dots}_1) = A,$$

b) $A(A+B) \cdot (A+B+C) \dots = A$

mert:

$$A(A+B) \cdot (A+B+C) \dots = A$$

$$A(\underbrace{AA+AB}) (\underbrace{AA+AB+AC})$$

$$\begin{array}{cc} \downarrow & \downarrow \\ \underbrace{A+AB} & \underbrace{A+AB+AC} \end{array}$$

$$\begin{array}{cc} \downarrow & \downarrow \\ \underbrace{A(1+B)} & \underbrace{A(1+B+C)} \end{array}$$

$$\begin{array}{cc} \downarrow & \downarrow \\ A \cdot A \cdot 1 & \cdot \quad A \cdot 1 = A \end{array}$$

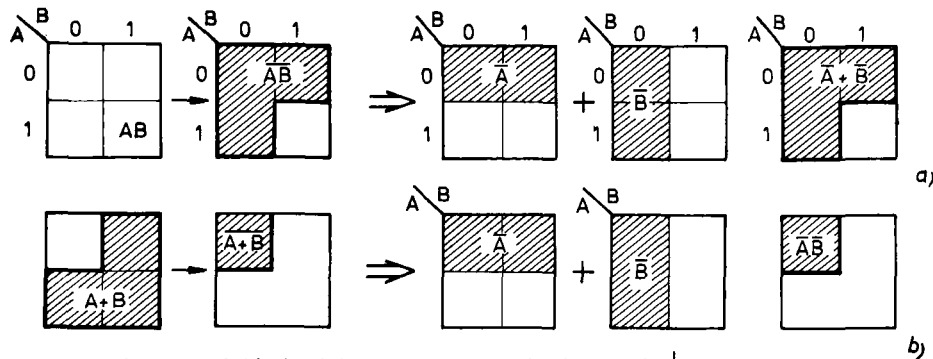
- DE-MORGAN szabályok

a) $\overline{A \cdot B} = \overline{A} + \overline{B},$

b) $\overline{A + B} = \overline{A} \cdot \overline{B}.$

Vagyis, a logikai szorzat negáltja összeg, amelyben a változók is negált értékekkel szerepelnek, a logikai összeg negáltja szorzat, amelyben a változók is negált értékkel szerepelnek. Negációnál tehát a változókat és a "műveleteket" is negálnunk kell. A két alap-egyenlet KARNAUGH táblákkal iga-

zolható (l. a 2.15a és b ábrát). Az abszorpciós tételkből és a DE-MORGAN szabályokból is látszik a logikai szorzás és összeadás szimmetriája, dualitása, vagyis hogy a logikai szorzás és összeadás jeleinek megcserélésével is igaz ugyanaz a szabály, azonosság. (Vigyázat! a duál kifejezés nem negáltat jelent, hanem csak a VAGY és az ÉS művelet cseréjét!)



Az egyenlet bal oldala = az egyenlet jobb oldalával

2.15. ábra

2.2.5. Az összes lehetséges kétváltozós logikai kapcsolat

Tudjuk, hogy a BOOLE-algebra a negáció, az ÉS, valamint a VAGY műveletekkel operáló rendszer. Az is világos, hogy ezeken az alapműveleteken kívül még sokféle logikai kapcsolat képzelhető el (melyekre szintén algebrát lehetne építeni). Ezek a logikai kapcsolatok is felírhatók a BOOLE-algebra segítségével Nem - És - Vagy, röviden NÉV alapműveletekkel.

Azért, hogy egyetlen logikai függvény se maradjon ki, egy olyan táblázatot célszerű készíteni, amely biztosan tartalmazza az összes lehetőséget. A függvénykapcsolat-lehetőségek száma:

$$2^{2^n}, \text{ ahol "n" a változók száma,}$$

hiszen az igazságtáblázat bal oldalán (ahol a változók 0-1 értékek-kombinációt soroljuk fel) 2^n "eset fordulhat elő" (ahány

darab n jegyű bináris szám van). Ekkor a 2^n "esethez" annyi Y logikai függvény tartozhat, ahányféleképpen az igazságtábla jobb oldalát ki lehet tölteni; ez pedig annyi, amennyi darab 2^n jegyű bináris szám írható függőlegesen a jobb oldalra (2^{2^n}). Két változó esetén eszerint 16 féle függvény lehetséges.

Két változóra tehát úgy rajzoljuk meg az összefoglaló táblázatot, hogy a bal oldalra az A és B változók érték-kombinációt írjuk 4 sorba úgy, ahogy eddig is tettük, a jobb oldalra pedig 16 oszlopba sorban az összes 16 féle lehetőséget felírjuk, mintha 4 jegyű bináris számok volnának 0000-tól 1111-ig (az index a decimális megfelelő érték):

A B	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉	Y ₁₀	Y ₁₁	Y ₁₂	Y ₁₃	Y ₁₄	Y ₁₅
0 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Vizsgáljuk meg a 16 féle lehetséges függvénykapcsolatot (nem követjük a számsorrendet)! Érdeemes megfigyelni, hogy az egymást 15-re kiegészítő indexű Y függvények egymásnak negáltjai

$$- Y_0 \equiv 0$$

"Null függvény", a változó értékétől függetlenül mindig 0 értékű.

$$- Y_{15} \equiv 1$$

"Egység függvény", a változók értékétől függetlenül mindig 1 értékű. E két függvény ("nulla tautológia", ill. "egy tautológia") nem jelent "hasznos" logikai kapcsolatot, egy logikai 0-ra, ill. logikai 1-re kötött vezetékkel valószínűleg meg. Inkább akkor érdekesek, amikor egy hosszabb kifejezés egyszerűsítésekor eredményül 0 vagy 1 eredményt kapunk, ilyenkor kiderül, hogy a függvénykapcsolatnak látszó kifejezés valójában konstans, áramköri megvalósításához tehát hozzá sem kell látnunk

- $Y_{12} \equiv A$
- $Y_3 \equiv \bar{A}$
- $Y_{10} \equiv B$
- $Y_5 \equiv \bar{B}$

Ezek a függvények nem fejeznek ki 2 változó közötti logikai kapcsolatot, hanem csak valamelyik változó ponált vagy negált értékét állítják elő,

- Y_8 :

A	B	Y_8
0	0	0
0	1	0
1	0	0
1	1	1

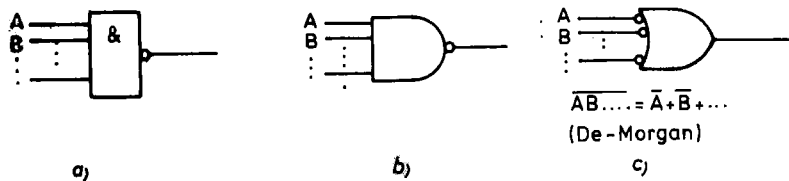
 $Y_8 = AB$ a kétváltozós ÉS kapcsolat, az előző-
ekből ismerjük: alapfüggvény,

- Y_7 :

A	B	Y_7
0	0	1
0	1	1
1	0	1
1	1	0

 $Y_7 = \overline{AB}$ az előző ÉS kapcsolat táblájával összehasonlítva látszik, hogy az Y_7 az előző ÉS függvény negáltja.

Az $Y_7 = \overline{AB}$ neve NEM-ÉS, NAND (Not-And) művelet.



2.16. ábra

A NAND UNIVERZÁLIS művelet, ami azt jelenti, hogy segítségével bármelyik logikai kapcsolat megvalósítható. Elvileg elegendő tehát csak NAND kapu áramköröket "tartanunk", belőlük bármely logikai áramkör felépíthető (l. a következő fejezetet!). Ennek nagyon nagy a gyakorlati jelentősége: a mai integrált áramkörös elektronikában is legtöbbször univerzális kapukat használunk, olyankor, amikor külön-külön elemekből ál-

litunk össze valamely feladat megoldására kapu-hálózatot (pl. TTL-ben a NAND az univerzális építőelem). A NAND kapu rajzjele: egy ÉS kapu rajza, kimenetén kiegészítve egy invertálást jelentő null-körrel (2.16a és b ábra). Némely kapcsolási rajzon a 2.16c ábrán látható jelöléssel találkozunk: VAGY kapu, invertált bemenetekkel. A De-Morgan-szabály alkalmazásával igazolható, hogy ez a jelölés is következetes.

A NAND bemenetek száma kettőnél több is lehet, a művelet az előzőekhez hasonlóan kiterjeszhető tetszőleges változóra

$$Y = \overline{ABC\dots X} \text{ formában.}$$

- Y_{14} :

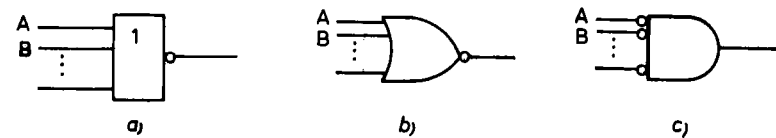
A	B	Y_{14}
0	0	0
0	1	1
1	0	1
1	1	1

 $Y_{14} = A + B$ a kétváltozós VAGY
kapcsolat,

- Y_1 :

A	B	Y_1
0	0	1
0	1	0
1	0	0
1	1	0

 $Y_1 = \overline{A + B}$ a VAGY kapcsolat
negáltja a NEM-
-VAGY, NOR (Not-
-Or) művelet.



2.17. ábra

A NOR is UNIVERZÁLIS művelet. Kizárólag NOR áramkörökből is felépíthető bármely logikai rendszer (pl.: ECL-ben és a CMOS-ban használják elterjedten, l. később). A NOR kétváltozós esethez hasonlóan több változóra is értelmezhető

$$Y = \overline{A + B + C + \dots} \text{ formában.}$$

A NOR kapu rajzjele olyan, mint a VAGY-kapué, csak a kimeneten a negációt kell null-körrel jelölni. (2.17a ábra a szab-

ványos, a b ábra a másik leggyakrabban szokásos, a c ábra pedig az esetenként előforduló "negált bemenetes" rajzjelet mutatja.)

- Y_6 :	A	B	Y_6	$Y_6 = A \oplus B$	ANTIVALENCIA kizáró-VAGY (exclusive-OR, EOR) kapcsolat
	0	0	0		
	0	1	1		
	1	0	1		
	1	1	0		

Az igazságtáblázatból láthatóan Y_6 akkor lesz 1, ha A nem egyenlő (antivalens) B-vel. Ez két esetben áll fenn, ha

$$A = 0 \quad \text{ÉS} \quad B = 1, \quad \text{VAGY} \\ A = 1 \quad \text{ÉS} \quad B = 0$$

ez BOOLE-egyenlettel, NÉV rendszerben felírva:

$$Y_6 = \bar{A}B + A\bar{B}$$

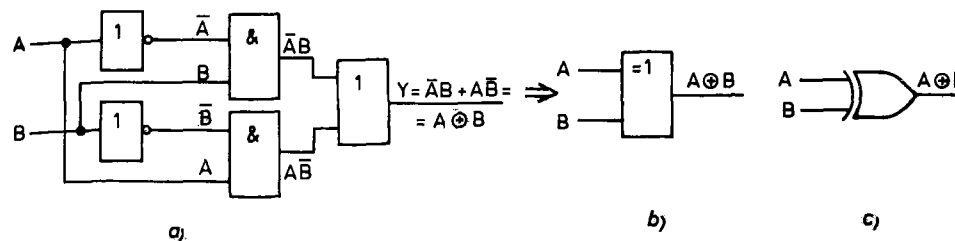
hiszen az egyenlet akkor ad 1-et, ha $A=0$ ÉS $B=1$ (ekkor $\bar{A}=1$ ÉS $B=1$, tehát az ÉS kapcsolat eredménye 1) VAGY, ha $A=1$ ÉS $B=0$ (ekkor tehát $A=1$ ÉS $\bar{B}=1$, az eredmény 1). Az Y_6 függvény két esetben ad 1-et, ezt az esetet két ÉS kapcsolattal jelöltük ki ($\bar{A}B$, $A\bar{B}$), közéjük VAGY jelet tettünk, mert akár az egyik, akár a másik ÉS kapcsolat teljesül, a függvényérték 1 lesz.

Az ANTIVALENCIA függvény másik neve KIZÁRÓ VAGY (EXCLUSIVE-OR), mert ez is VAGY kapcsolat, de kizáró értelemben; az eredmény 1-es, ha vagy az egyik, vagy a másik változó 1-es (ha mindkettő 1-es, akkor $Y = 0$). A hétköznapi életben is ilyen értelemben szoktuk használni a "vagy" szót.

A szokásos műveleti jel: =1, vagy \oplus a változók közé tett, bekarikázott \oplus jel (ugyanaz, mint a modulo-2 összeadás jele; az igazságtáblázat alapján beláthatjuk, hogy az átvitel nélküli összeadás értéktáblázata is ugyanilyen: $0 \oplus 0 = 0$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$ és átvitel nincs). A kizáró-VAGY tehát két bit maradék nélküli összeadásának aritmetikai műveletét is elvégzi ("fél összeadó").

A megvalósítás NÉV rendszerben (Nem-ÉS-Vagy elemekkel) a BOOLE-egyenlet alapján történhet (2.18a ábra), de ma már a

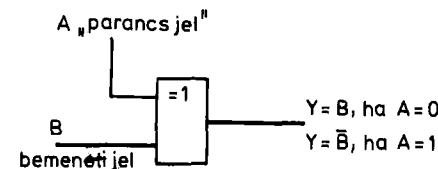
legtöbb integrált áramkör készletben kapható exclusive-OR kapu (pl. TTL-ben a 7486-os), ezekre külön kapcsolási rajzjel létezik (a 2.18b ábrán a szabványos, a 2.18c ábrán az ezenkívül szokásos jel látható).



2.18. ábra

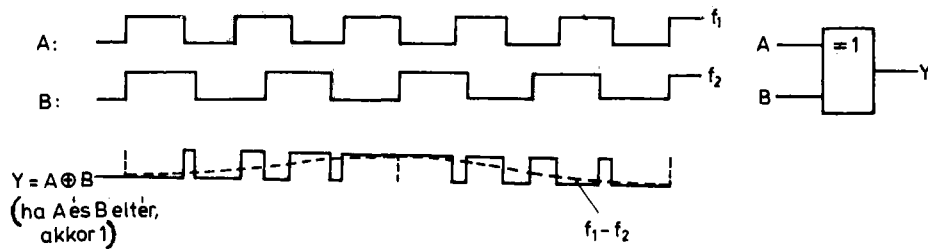
Az antivalencia kapu egy bites "digitális komparátor"-nak, összehasonlítóknak is felfogható: ha a bemenetre érkező két bit azonos értékű, akkor a kimenetén 0 jelzést ad, ha eltérő, akkor 1-est (ℓ. az igazságtáblázatot!).

A kizáró-VAGY kapu "vezérelhető inverter"-ként is felhasználható. Az igazságtáblázatból belátható, hogy amikor az egyik bemenet, pl. az A "parancsjel" 0, akkor a másik, B, bemenetű jel változtatás nélkül a kimenetre jut (első két sor), ha a parancsjel 1-es, akkor a B jel negáltja jut a kimenetre (2.19. ábra). Így olyan áramköri egység birtokába juthatunk, amely "elektronikusan vezérelhetően" egy digitális jelet változtatás nélkül átenged vagy negál.



2.19. ábra

Ugyancsak ennek a kapunak tulajdoníthatjuk a "páratlan-ság vizsgáló" funkciót is: ha páratlan számú bemenet van 1-es szinten, akkor a kimenet 1-es, ellenkező esetben a kimenet 0.



2.20. ábra

Érdekes és legjobban idődiagramon bemutatható funkciója e kapunak a "digitális modulátor". Ha a kizáró-VAGY kapu bemenetére eltérő frekvenciájú, 50%-os kitöltési tényezőjű négyzet (logikai szintű) jelet adunk, akkor a kimeneten olyan periodikusan "szélesedő", majd "keskenyedő" impulzus-sorozat jelenik meg, amelynek középértéke (aluláteresztő szűrővel nyert komponense) szinuszosan változik és frekvenciája egyenlő a két bemeneti jel frekvenciájának különbségével (a 2.20. ábrán szaggatott vonal). Így lehet például "digitális szinuszgenerátort" készíteni. Az EOR kapu a híradástechnikában, műszerelektronikában gyakran alkalmazott kétszeresen kiegyenlített balanszmodulátor "digitális megfelelője".

A több változóra való kiterjesztés ebben az esetben már nem annyira magától értetődő, mint az ÉS, és a VAGY kapcsolatnál. A szabvány (MSZ 9200/33-73 "VILLAMOS RAJZJELEK Kétállapotú (bináris) logikai elemek" - erre a szabványra számának feltüntetése nélkül többször hivatkozni fogunk) ezt az értelmezést írja elő egy mellékelt háromváltozós igazságtáblázatban:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Más értelmezés is létezik azonban különböző hardver leírásokban, a szakirodalomban. Eszerint a kiterjesztést úgy kell végezni, hogy a kétváltozós eset igaz maradjon ismételt alkalmazással is, azaz, ha két változót EOR kapcsolatba hozunk, és ennek 0 vagy 1 eredményéhez kapcsoljuk a következő változót és így tovább, akkor az így adódó eredmény adja a függvényértéket egy adott kombinációban. Ebben az értelmezésben pl. a háromváltozós esetben az igazságtábla utolsó sorában az A és B EOR kapcsolatából 0 adódik ($1 \oplus 1 = 0$); ehhez hozzávéve $C = 1$ -et, az eredmény 1, nem pedig 0. A függvényérték képzése ebben a rendszerben is egyszerű, logikus: felhasználva a \oplus jelből következő rokonságot, modulo-2 összegezéssel össze kell adnunk a változók aktuális értékét, más szóval páratlan számú 1 esetén $Y=1$ -et kell írunk (az említett esetben: $1 \oplus 1 \oplus 1 = 1$). Így a háromváltozós igazságtáblázat a következő:

A	B	C	Y' = A \oplus B \oplus C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Amikor kettőnél többváltozós EOR-ral találkozunk, minden esetben meg kell győződnünk arról, hogy melyik értelmezésről van szó.

A szabvány egyébként az előbbi igazságtáblázattal a Modulo-2 ÖSSZEADÓ alapfunkciót definiálja (M2).

- Y₉:

A	B	Y ₉
0	0	1
0	1	0
1	0	0
1	1	1

Y₉ = A \odot B

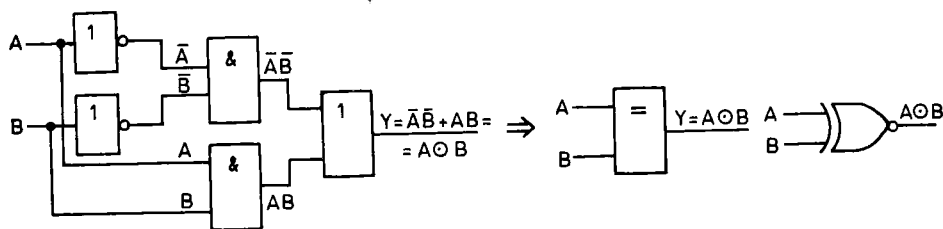
EKVIVALENCIA
(exclusive-NOR,
ENOR) kapcsolat

Ez az antivalencia negáltja; a kimenet akkor 1-es, ha A és B értéke egyenlő, ekvivalens. Ez két esetben lehetséges; ha mindkettő 0 (\overline{AB} akkor 1), vagy ha mindkettő 1-es (AB akkor 1). Az ennek megfelelő BOOLE-egyenlet:

$$Y = \overline{AB} + AB = A \odot B.$$

A szokásos műveleti jel a bekarikázott szorzójel: \odot és az egyenlőség jel: =.

A NÉV megvalósítás az előzőkhöz hasonlóan a BOOLE-egyenlet alapján történhet, de az ekvivalencia áramkört egyetlen blokk-ként is kezelhetjük külön jelöléssel, vagy egy negált kimenetű kizáró-VAGY kaput rajzolunk (exclusive-NOR, 2.21. ábra).



2.21. ábra

A kétváltozós ekvivalencia kapcsolatra minden, az EOR kapcsolatra vonatkozó megállapítás érvényes - figyelembe véve természetesen, hogy az ENOR az EOR negáltja.

A szabványban közölt háromváltozós igazságtáblázat (amelyből többváltozós esetre is következtethetünk) a következő:

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Egyéb forrásokban előforduló más értelmezésben az ENOR akkor 1-es, ha a változók között páros számú 1-es értékű van (EOR-negált).

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

A szabványban erre külön a "párosság" alapműveletet definiálják.

- Y_2 :

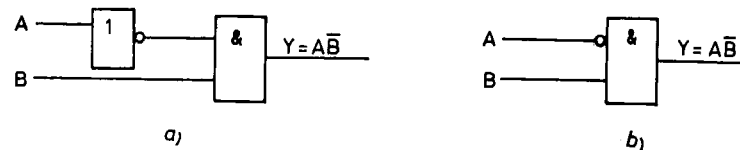
A	B	Y_2
0	0	0
0	1	1
1	0	0
1	1	0

$$Y_2 = A \supset B$$

TILTÁS, INHIBICÍÓ művelete: "A tiltja B-t"

Az A esemény megjelenése tiltja B-t, azaz a kimenet értéke azonos B-vel, ha az A nem következik be; ha A következik ($A=1$ lesz), akkor B nem jut a kimenetre, Y értéke 0 lesz, ahogyan ez az igazságtáblázatból látható. A BOOLE kifejezés felírása egyszerű: Y_2 akkor 1, ha $A=0$ és $B=1$:

$$Y = \overline{A}B$$



2.22. ábra

Áramköri megvalósítása a 2.22a, rövidített jelölése a 2.22b ábrán látható. Funkciója hasonló az ÉS kapuzási funkcióhoz, csak a kapuzó jel ebben az esetben negált értékű (\bar{A}). A "tiltást", mint alapműveletet éppen ezért ritkán használjuk és a műveleti jel helyett is inkább az algebrai alakot ($\bar{A}B$) írjuk le. A tiltás - szemben az összes eddig tárgyalt alapművelettel - nem kommutatív: nem mindegy, hogy A tiltja-e B-t vagy fordítva.

- Y_4 :	<table style="border-collapse: collapse; margin: 0 auto;"> <tr><td style="padding: 2px 5px;">A</td><td style="padding: 2px 5px;">B</td><td style="padding: 2px 5px;">Y_4</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td></tr> </table>	A	B	Y_4	0	0	0	0	1	0	1	0	1	1	1	0	$Y_4 = B \supset A = \bar{A}B$	TILTÁS, INHIBÍCIÓ művelete: "B tiltja A-t"
A	B	Y_4																
0	0	0																
0	1	0																
1	0	1																
1	1	0																

"B tiltja A-t": azt jelenti, hogy ha B megjelenik (1 lesz), akkor A nem juthat a kimenetre. Az előző tiltási művelethez képest csupán annyi az eltérés, hogy a két bemeneti változó szerepet cserélt.

- Y_{11} :	<table style="border-collapse: collapse; margin: 0 auto;"> <tr><td style="padding: 2px 5px;">A</td><td style="padding: 2px 5px;">B</td><td style="padding: 2px 5px;">Y_{11}</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td></tr> </table>	A	B	Y_{11}	0	0	1	0	1	1	1	0	0	1	1	1	$Y_{11} = A \rightarrow B = \bar{A} + B$	IMPLIKÁCIÓ, következtetés művelete: "Ha A, akkor B is"
A	B	Y_{11}																
0	0	1																
0	1	1																
1	0	0																
1	1	1																

Ez a művelet a gyakorlatban ritkán használatos, inkább arra alkalmas, hogy belássuk: "bonyolultabb" logikai kapcsolatok is léteznek, amelyeket a BOOLE-algebra alapműveleteivel minden nehézség nélkül le tudunk írni. Az implikáció: $A \rightarrow B$ igazságtáblázatához (amely az Y_4 inhibíció negáltja) a következő leírást adhatjuk: ha az A esemény bekövetkezik ($A=1$ lesz), akkor B-nek is be kell következnie, különben az állításunk nem igaz ($Y = 0$). Ha $A = 0$ (táblázat első két sora), akkor B értéke közömbös, nem sértjük meg a szabályt, $Y = 1$. Az A esemény megtörténte tehát maga után vonja a B esemény megtörténtét. Az algebrai alak felírásában felhasználjuk, hogy ez a függvény az Y_4 tiltás-függvény negáltja, így a De-Morgan szabályt alkalmazhatjuk:

$$Y_{11} = \overline{Y_4} = \overline{\bar{A}B} = A + B$$

- Y_{13}

A	B	Y_{13}
0	0	1
0	1	0
1	0	1
1	1	1

$$Y_{13} = B \rightarrow A = A + \bar{B}$$

IMPLIKÁCIÓ,
következtetés
művelete:
"Ha B, akkor
A is"

Egyezik az előző művelettel, csak a két változó szerepet cserélt - az implikáció sem kommutatív kapcsolat.

2.2.6. Többváltozós logikai kapcsolatok

Az előző pontban áttekintettük, definiáltuk az összes kétváltozós logikai kapcsolatot és amelyiket lehetett, értelmeztük több változóra is, ezért ezekkel újra nem foglalkozunk. Léteznek azonban olyan függvények, amelyek csak kettőnél több változóra értelmezhetők. A következőkben betű szerint idézzük a már említett MSZ.9200/33-73 szabvány erre vonatkozó részét, amelyben közli ezen függvények elnevezését, háromváltozós igazságtáblázatát, műveleti jelét. (1. táblázat)

2.2.7. Univerzális műveletek: logikai alapműveletek megvalósítása univerzális építőelemekkel

Az előzőkben azt mondtuk az univerzális műveletekről, hogy segítségükkel bármely logikai függvény előállítható. A gyakorlatban ez azt jelenti, hogy NAND vagy NOR kapukkal minden logikai feladatot meg lehet oldani. A mai, aktiv elemes integrált áramköri családok szempontjából ez nagyon jelentős, ui. áramkörileg a NAND és a NOR valósítható meg legelőnyösebben. Vannak áramkör családok, amelyek főként NAND-ra építenek, NAND kapukból rendelkeznek a leggazdagabb választékkal (pl. DTL, TTL), vannak, amelyek főleg NOR-ral dolgoznak (pl. CMOS, ECL - az említett áramkörökről később részletesebben szó lesz).

(Az 5. táblázat folytatása)

Sor-szám	A funkció megnevezése	Igazság táblázat*	Rajzjel	Megjegyzés
13.	n és csakis n	$X_1 X_2 X_3 Y$ 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 0		n csak 1-nél nagyobb és az összes bemenetek számánál kisebb lehet.
14.	Logikai küszöb általános jelölés	$X_1 X_2 X_3 X_4 Y$ 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 1 1 0 1 1 0 1 0 1 1 1 1 1 0 0 0 0 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		n csak 1-nél nagyobb és az összes bemenetek számánál kisebb lehet.
15.	Majoritás	$X_1 X_2 X_3 Y$ 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 1		Az elemen tet-szöleges páratlan számú egynél több bemenet lehet

* Műveleti táblázat kifejezés is használatos.

Ahhoz, hogy a NAND és a NOR univerzális tulajdonságát be-lássuk, lényegében elegendő, ha igazoljuk, hogy a BOOLE-alap-műveletek mindegyike (Nem-És-Vagy) megvalósítható NAND-del és NOR-ral (hiszen a BOOLE-alapműveletekkel minden más logikai kapcsolat megvalósítható).

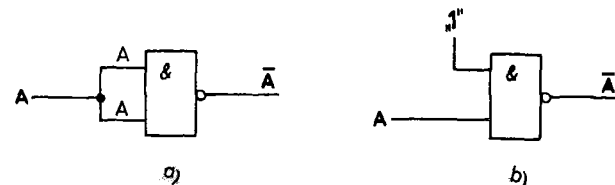
NAND-del való megvalósítás

A megvalósítandó függvényeket minden esetben \overline{AB} alakra kell hoznunk (amelyben A és B helyén bármely változó ponált vagy negált értékkel is szerepelhet, vagy egy egész kifejezés is lehet).

a) Negáció

Egy NAND kapunak legalább két bemenete van, a negáció egyváltozós művelet. Felmerül a kérdés, hogy a "felesleges" be-menettel mit tegyünk. A negáció két bemenetre bővítésének két lehetősége van:

- $Y = \overline{A} = \overline{A \cdot A}$ tekintve, hogy egy változót akárhányszor meg-szorozhatunk önmagával - lásd a 2.2.4. pont alaptételeit.



2.23. ábra

Ennek a 2.23a ábra megvalósítása felel meg: a két bemenetet ugyanarra az A jelre kötjük.

- $Y = \overline{A} = \overline{A \cdot 1}$ mivel az 1-gyel való szorzás nem befolyásol-ja egy változó értékét.

Eszerint a nem használt NAND kapu bemenetet állandó lo-gikai 1 szintre is köthetjük (2.23b ábra). A kétféle megoldá-si lehetőség ismerete azért is fontos, mert logikai áramkörök tervezésekor gyakran előfordul (nem csak invertálás esetén),

hogy egy-egy rendelkezésre álló NAND kapunak a szükségesnél több bemenete van, ilyenkor a felesleges bemeneteket az előző két módszer szerint kezelhetjük, vagyis:

- a nem használt kapubemeneteket egy-egy változó vezetékével összeköthetjük vagy

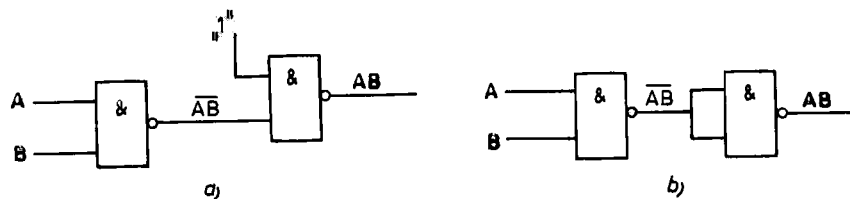
- a nem használt bemeneteket logikai 1-re köthetjük.

A két módszer logikai szempontból ekvivalens, de áramköri szempontból különböző. Ha két vagy több nem használt bemenetet egy változó-bemenettel összekötünk, akkor az illető változó vezetékét többszörös bemenő áram terhelheti (több bemenet - nagyobb áramfogyasztás), ami legtöbbször nem kezdvező. A második megoldás ettől a hátránytól mentes, ezért általában ezt részesítjük előnyben, tehát a nem használt NAND kapu bemeneteket logikai 1-re kötjük.

Érdemes megjegyezni, hogy az invertálás NAND kapuval való végrehajtása inkább elvi jelentőségű, hiszen minden áramkör családban vannak külön inverter típusok.

b) Az ÉS művelet megvalósítása NAND kapukkal

Tekintve, hogy a NAND kapu az ÉS művelet negáltját állítja elő (Nem-És), a kapu kimeneti jelének negálásával az ÉS műveletet valósíthatjuk meg (negálni az előző pont értelmében a lehetséges NAND kapuval: 2.24a és b ábra).



2.24. ábra

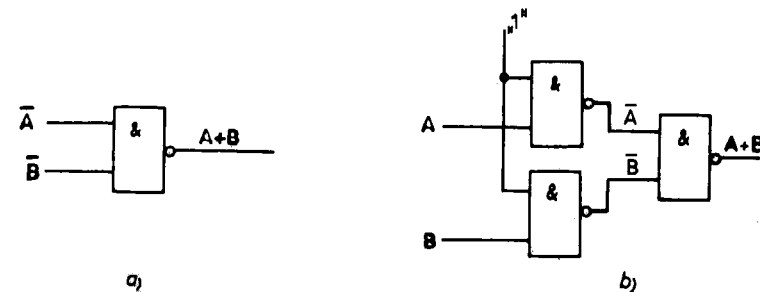
c) A VAGY művelet megvalósítása NAND kapukkal

Negáljuk kétszer a VAGY függvényt (De-Morgan szabály szerint), akkor az értéke nem változik:

$$Y = \overline{\overline{A + B}} \quad \text{az "alsó" negálást elvégezve:}$$

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

az eredmény NAND függvény, csak a bemenetekre a változók negáltját kell adni (negálni viszont lehet NAND kapuval, 2.25. ábra).



2.25. ábra

Logikai rajzokon legtöbbször (és a továbbiakban), ha az A, B, C stb. változók negáltját kell valamelyik bemenetre vezetnünk, akkor a negáló invertert (NAND kaput) nem tüntetjük fel, hanem csak jelöljük \overline{A} , \overline{B} , \overline{C} stb. odairásával, ahogyan azt a 2.25a ábrán tettük. Egy nagyobb rendszerben, készülékben ui. valószínű, hogy valahol közvetlenül rendelkezésre áll a jel negáltja is, nem érdemes külön invertereket ezért lefoglalni (pl. flip-flopok kettős kimenetén a jel-negált is levehető, a mechanikus kapcsolók is legtöbbször negált jel kiadására is "készíthetők"). Ha a végső összeállítás során kiderül, hogy nem áll rendelkezésre valamelyik változó negáltja és szükség lenne rá, akkor berajzoljuk az invertert.

d) Egyéb fontos kétváltozós alapműveletek megvalósítása

Az előző pontokban beláttuk, hogy a NÉV (Nem-És-VAGY) alapműveletek megvalósíthatók NAND kapuval, ha egy kicsit bonyolultabban is, mintha közvetlenül ÉS kapuval, VAGY kapuval dolgoznánk. Ezzel tulajdonképpen azt is beláttuk, hogy valamennyi logikai művelet megvalósítható NAND kapuval. Gyakorlásképpen a fontosabb kétváltozós logikai függvényeket is valósítsuk meg!:

- Kizáró VAGY, antivalencia $Y = A \oplus B$

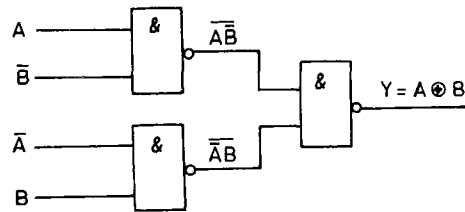
$$Y = \overline{A}B + A\overline{B}$$

- ez a Nem-És-Vagy alak, ezt kétszer negálva:

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

NAND alakot kapjuk, ha az alsó negációt elvégezzük.

A megvalósításhoz az algebrai alakból láthatóan három NAND kapu szükséges: egy-egy NAND kapu valósítja meg az $\overline{A}B$, ill. az $A\overline{B}$ kapcsolatot, majd ezek jeléből egy harmadik állítja elő a függvényt (2.26. ábra, a változók negáltjaihoz az inverttereket már nem rajzoltuk be).

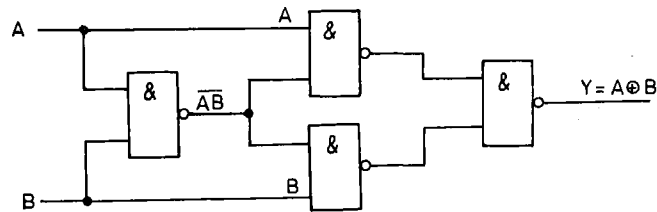


2.26. ábra

Van egy másik lehetőség is, ha a változók negáltja nem áll rendelkezésre:

$$Y = \overline{A}B + A\overline{B} = 0 + \overline{A}B + \overline{A}B + 0 = \overline{A}A + \overline{A}B + \overline{A}B + B\overline{B} = \overline{A}(\overline{A} + B) + B(\overline{A} + \overline{B}) = \overline{A} \cdot \overline{A}B + B \cdot \overline{A}B = \overline{A} \cdot \overline{A}B \cdot B \cdot \overline{A}B$$

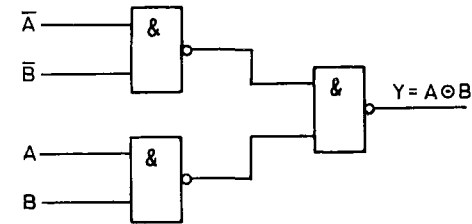
Megvalósítását a 2.27. ábra mutatja.



2.27. ábra

- Ekvivalencia $Y = A \odot B$

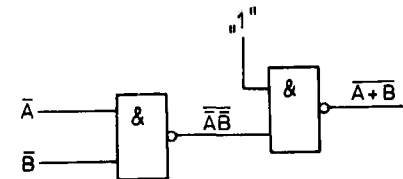
$$Y = \overline{\overline{A}B} + \overline{A\overline{B}} = \overline{\overline{A}B} \cdot \overline{A\overline{B}} \quad (2.28. \text{ ábra})$$



2.28. ábra

- Nem-VAGY, NOR $Y = \overline{A + B}$

$$Y = \overline{A + B} = \overline{A}B = \overline{\overline{A}B} \quad (2.29. \text{ ábra})$$



2.29. ábra

NOR-ral való megvalósítás

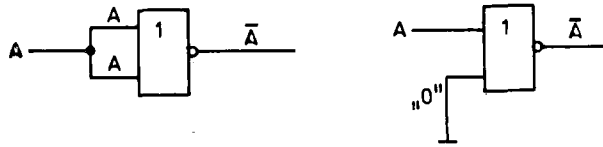
A megvalósítandó függvényeket minden esetben $\overline{A + B}$ alakúra kell hoznunk (A és B helyén ponált, negált vagy egy kifejezés is lehet!)

a) Negáció

A NOR kapu is invertáló jellegű (inverter kimenetű), így a negáció végrehajtható vele, csupán az a kérdés, hogy a felesleges bemenettel mit tegyünk.

- $Y = \overline{A} = \overline{A + A}$ vagyis a nem használt bemenetet az "élő" bemenettel (jelbemenettel) párhuzamosan köthetjük. Tudjuk, hogy a többszörös terhelés miatt (a meghajtó áramkört több bemenet áramával terheljük, nem csak egyet) ez a megoldás villamos szempontból általában nem kedvező (2.30a ábra),

- $Y = \bar{A} = \bar{A} + 0$ vagyis a NOR kapu nem használt másik (többi) bemenetét konstans logikai 0-ra köthetjük (2.30b ábra). Ez természetesen nem csak az invertálásra érvényes, hanem bármely esetre, amikor felesleges, többlet bemenet van. A logikai 0-ra kötésről nem szabad elfeledkezni, mert például a ma legelterjedtebb TTL áramkörök kapubemenetei olyanok, hogy üresen hagyva logikai 1-es szintűnek számítanak. Ezért TTL NOR kapu bemenetét nem szabad üresen, szabadon hagyni, ha nem akarjuk, hogy a kimenet emiatt állandó logikai 0-an legyen.



2.30. ábra

b) Az ÉS művelet megvalósítása NOR kapuval.

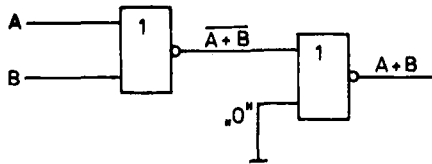
$Y = AB = \overline{\overline{AB}} = \overline{\bar{A} + \bar{B}}$ kétszeres negálással jutunk NOR-hoz (2.31. ábra).



2.31. ábra

c) A VAGY művelet megvalósítása NOR kapuval.

$Y = A + B = \overline{\overline{A + B}}$ vagyis A, B NOR kapcsolatát negáljuk (2.32. ábra).



2.32. ábra

d) Egyéb alapl műveletek megvalósítása NOR kapukkal:

- Kizáró-VAGY

$Y = A \oplus B$ A függvényt addig kell alakítanunk azonos átalakítással, kétszeres negálással és a De-Morgan szabály alkalmazásával, amíg csak NOR kapcsolatot tartalmazó függvényt állítunk elő.

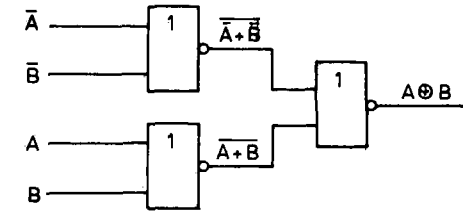
$Y = \overline{AB} + \overline{\bar{A}\bar{B}} = \overline{\overline{\overline{AB} + \bar{A}\bar{B}}} = \overline{(\bar{A} + B)(A + \bar{B})}$ ez utóbbi kifejezésben a szorzást elvégezve:

$$Y = \overline{AB + \bar{A}\bar{B}}$$

(igazoltuk, hogy az antivalencia az ekvivalencia negáltja).

A "belső tagokat" kétszeresen negálva (azért, hogy az ÉS kapcsolatot NOR-rá alakíthassuk):

$Y = \overline{\overline{AB} + \overline{\bar{A}\bar{B}}} = \overline{\bar{A} + \bar{B} + A + B}$ a végeredmény már csak NOR kapcsolatot tartalmaz (2.33. ábra).



2.33. ábra

- Ekvivalencia

$$Y = A \odot B$$

Az előzőhöz hasonló átalakításokkal:

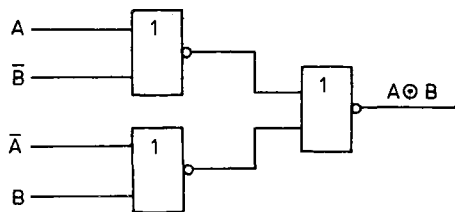
$Y = \overline{\overline{A + \bar{B}} + \overline{\bar{A} + B}}$ eredményre jutunk (2.34. ábra).

- Nem-És, NAND

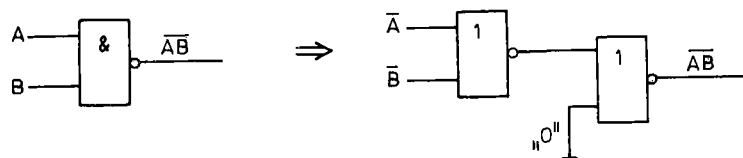
$$Y = \overline{AB}$$

$$Y = \overline{AB} = \overline{\bar{A} + \bar{B}} = \overline{\overline{\bar{A} + \bar{B}}}$$

(2.35. ábra).



2.34. ábra



2.35. ábra

A későbbiekben még foglalkozunk az univerzális kapukból felépülő hálózatok egyszerű tervezési módszereivel, olyanokkal, amelyek nem igénylik a BOOLE-algebrai azonos átalakítások sorozatának (időnként bizonytalan eredményű) végrehajtását, hanem "közvetlenül" NAND vagy NOR, ill. AND-OR-INVERT formát eredményeznek. (Az AND-OR-INVERT, azaz ÉS-NEM VAGY kapu a NOR kapu egy célszerű, gazdaságosan felhasználható változata.)

2.3. Logikai függvények

2.3.1. Logikai függvények megadása

Az eddigiekben a kétváltozós logikai kapcsolatok tárgyalásakor is végeredményben logikai függvényekről volt szó. A függvényt - amely 0 vagy 1 értéket vehetett fel - Y-nal jelöltük. Az Y értéke a logikai változók 0 vagy 1 értékeitől függ, az Y és az A, B, C, ... stb. változók között egyértelmű kapcsolat van, az Y az A, B, C, ... stb. "független változók" függvénye.

A logikai kapcsolatok függvényként való kezelése teszi lehetővé számunkra, hogy algebrai, "matematikai" módszerekkel, megfelelő átalakításokkal, operációkkal - átmenetileg függet-

lenítve magunkat "tényleges" fizikai jelentésétől - viszonylag egyszerű, exakt módon tervezhessünk, analizálhassunk digitális áramköröket. A fő cél általában az egyszerűsítés, ill. egy adott (rendszerint univerzális elemeket tartalmazó) típusválasztékból való áramkör építése.

Egy adott feladat kapcsán a leíró logikai függvény többféle alakjával, megjelenési formájával, megadásával kerülhetünk szembe. Ezek közül a leggyakoribbak:

- az igazságtáblázat,
- az algebrai (BOOLE-) kifejezés,
- a logikai kapcsolási rajz,
- az idődiagram,
- a KARNAUGH-tábla.

Röviden tekintsük át a megadási formákról a legfontosabb tudnivalókat!

Az igazságtáblázat - mint tudjuk - a változók összes lehetséges 0-1 érték kombinációjára megadja az Y "válasz" értékét, tartalmazza az összes 2^n számú lehetőséget. Ugy is mondják, hogy a függvény az "adat-tartományban" (data-domain) van adva. A mai modern műszerek közül a logikai analizátorok képesek egy digitális hálózat állapotainak vizsgálatára, képernyőn való kiírására, adat-tartományban való analizálására, mérésére.

Az algebrai kifejezéssel való felírás közvetlenül a változókkal és a köztük irt logikai műveleti jelekkel történik, vagyis az Y egy BOOLE-egyenlettel van adva. Ez a legkönnyebben érthető, leginkább szemléletes megadás. Hátránya viszont, hogy egy logikai függvénynek igen sok hosszabb-rövidebb ekvivalens felírása lehetséges (éppen a hosszabb kifejezések "rövidítése", egyszerűsítése, minimálása az egyik legfontosabb feladat tervezéskor). Ha valamely függvény megadásához olyan algebrai kifejezést írunk fel, amelyben csak logikai összegezés, logikai szorzás és negáció fordul elő, és ezeket is szabályszerű, kanonikus, ún. teljes normál alakba rendezzük, akkor már csak kétféle ekvivalens van (szorzatok összege és összegek szorzata alakjában). Például a kizáró-VAGY függvénykétféle normál alakja

$$Y = A \bar{B} + \bar{A} B \quad (\text{szorzatok összege}),$$

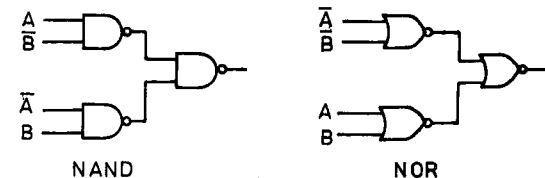
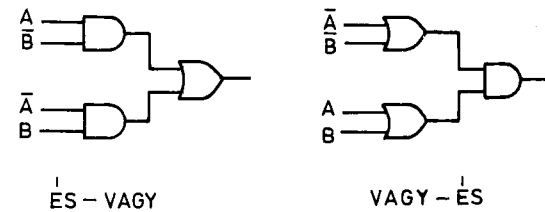
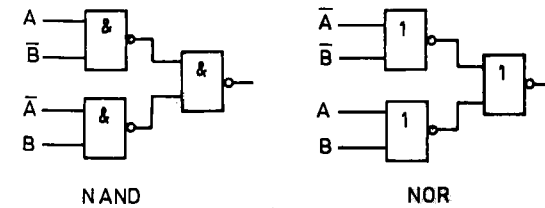
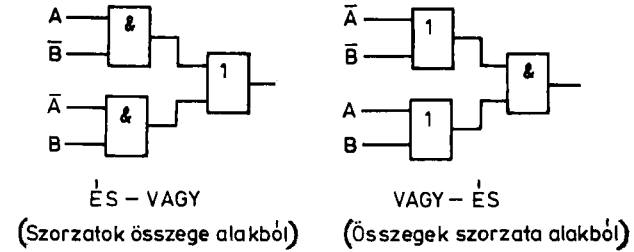
$$Y = (\bar{A} + \bar{B})(A + B) \quad (\text{összegek szorzata}).$$

Az első egyenletből a másodikat pl. a logikai szorzat disztributivitásának (tényezőnkénti "be-összegelhetőségének") felhasználásával kaphatjuk.

A logikai kapcsolási rajz a már megismert kapu-szimbólumokat használja, ezek összekapcsolását ábrázolja. Ennek alapján a logikai függvényt előállító kombinációs hálózat már megépíthető, végeredményben a logikai rajz a logikai áramkörök "tervrajza", kapcsolási terve. Ennél részletesebbet nem is szoktak rajzolni egy digitális berendezéshez, a kapuk "belsőjét" (belső áramköreit) nem ábrázolják, hiszen egy adott áramkör családon belül kapu típusonként ez ugyanis azonos. A sok felesleges ismétlődő részlet áttekinthetetlenné is tenné a rajzokat. Legtöbbször a logikai kapcsolási rajz az elektronika tervezésének végső állomása. A tervezés során az adott feladat alapján általában először elkészítjük az igazságtáblázatot, ennek megfelelően felírjuk a logikai függvényt, amelyet valamely alkalmas módszerrel úgy egyszerűsítünk, hogy a "minimális" hálózatot kapjuk (rendszerint univerzális elemekből, típusválaszték figyelembevételével), és végül az egyszerűsített függvényt megvalósító kapcsolást rajzoljuk fel. Például a kizáró-VAGY kapcsolatot megvalósító hálózat az előzőek szerint ÉS-VAGY, VAGY-ÉS, NAND és NOR változatban a 2.36. ábra kapcsolásaival valósítható meg (a szabványos mellett a b ábrán a másik szokásos rajzjellel is felrajzoltuk a kapcsolást).
Figyeljük meg, hogy az ÉS-VAGY valamint a NAND megvalósítás teljesen azonos strukturájú (bemeneti változók ponált-negált értékei, kapu-elrendezés, huzalozás), az ÉS-VAGY hálózatból a NAND pusztán kapu-átrajzolással megvalósítható. Hasonlóan a VAGY-ÉS hálózatból a NOR hálózat a VAGY, ill. ÉS kapuk NOR-rá való átrajzolásával létrehozható "minden külön szellemi erőfeszítés nélkül".

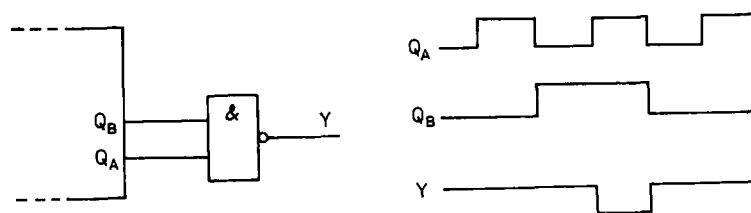
Az idődiagram a logikai hálózat különböző pontjain lévő jelek időfüggvényeit ábrázolja (négyszögjelek alakjában). Legtöbbször a koordinátatengelyeket nem is rajzoljuk meg. Az időfüggvények és a logikai függvények között természetesen egy-

értelmű kapcsolat van, a függvény az időtartományban (time-domain) van adva. Az időfüggvényekre "ránézve" általában nehezen ismerhető fel a logikai függvény, csak az igazságtáblázatra visszavezetve, átírva, egyszerűsítve stb. állapítható meg.

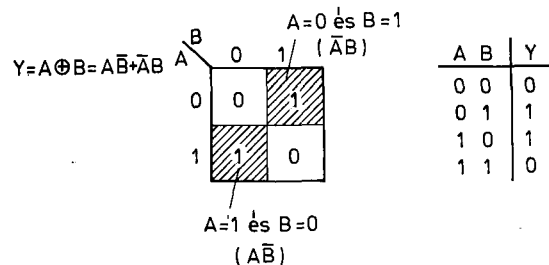


2.36. ábra

A gyakorlatban viszont nagyon sokszor időtartományban dolgozunk. Oszcilloszkóppal a digitális jelek idődiagramjai jelelnithetők meg. A sorrendi hálózatok (amelyeknél a logikai állapotok az előzőleg felvett állapotoktól is függenek) vizsgálatára, működésének leírására, tanulmányozására éppen az idődiagram a legalkalmasabb eszköz (pl. flip-flopok, számlálók, regiszterek, stb.). A 2.37. ábrán példa gyanánt egy olyan idődiagramot láthatunk, amely a NAND áramkör kapuzási funkcióját érzékelteti; a kapu két bemenetére (pl. egy számlálóból) a Q_A és Q_B jel jut, ha bármelyik bemenet 0 értékű, akkor a NAND kimenet 1-es, csak abban az egy esetben lesz 0, ha mindkét bemenet 1-en van.



2.37. ábra



2.38. ábra

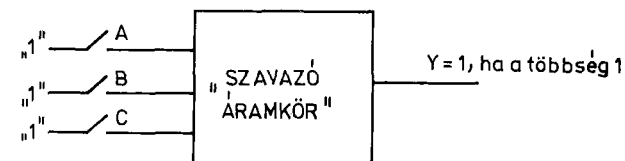
A KARNAUGH-tábla az igazságtáblázat célszerűen - és mint a következőkben látni fogjuk, egyszerűsítésre alkalmasan - átrendezett formája. Minden logikai függvénynek megfelel egy olyan terület a KARNAUGH-táblán, ahol a függvényérték 1-es ($Y = 1$). Ezekbe a "rekeszekbe" 1-est írunk, a többi helyre, ahol $Y = 0$, zérust írunk vagy üresen hagyjuk. Például a kétváltozós kizáró-VAGY függvény akkor 1-es, ha $A = 1$ és $B = 0$,

vagyis az $A = 1$ sor és a $B = 0$ oszlop találkozási pontjába 1-et írunk. A másik 1-es az $A = 0$ -nak megfelelő sor és a $B = 1$ -nek megfelelő oszlop közös rekeszében van (2.38. ábra). Ez a KARNAUGH-tábla a függvényt egyértelműen meghatározza és minden olyan információt megad, amit az igazságtáblázat (amelyet az összehasonlítás kedvéért a tábla mellé rajzoltunk), csak éppen koordináta-rendszerhez hasonló elrendezésben.

2.3.2. Logikai függvények felírása adott feladat alapján

A gyakorlatban legtöbbször a logikai függvények nincsenek közvetlenül adva az előző módszerek valamelyikével, hanem csak a feladat (sokszor "szöveges" formában) ismert, amelyet a logikai egységgel meg kell valósítanunk. Ha a feladatot "spekulatív uton" próbálgatással oldanánk meg, akkor az esetek többségében nagyon nehezen jutnánk eredményre. Hibátlan és viszonylag gyors megoldást csak megfelelő "szisztematikus" módszerrel érhetünk el. Az adott feladat alapján általában először egy - az összes lehetséges "eset", kombinációt összefoglaló - igazságtáblázatot célszerű elkészíteni. Az igazságtáblázatból már fel tudjuk írni a logikai függvényt, amelyet ezután egyszerűsítünk, minimálunk, azért, hogy a megvalósítandó logikai hálózat minél egyszerűbb, olcsóbb legyen. Az egyszerűsített függvény alakból megrajzolhatjuk a végleges logikai kapcsolási rajzot - ezt tekintjük végeredménynek.

Az első felmerülő kérdés az, hogy az igazságtáblázatból, miképpen írható fel a logikai függvény. Nézzük például a következő feladatot:



2.39. ábra

készítsünk egy olyan áramkört, amelynek kimenetén logikai 1 jelenik meg, ha 3 bemenetén a 3 változó (A, B, C) közül a többség (2 vagy 3) logikai 1-en van ("szavazó áramkör")(2.39. ábra). Az igazságtáblázat könnyen megrajzolható (minden sorban, ahol 2 vagy 3 darab 1-es van, az Y = 1 lesz):

A	B	C	Y	
0	0	0	0	(többség 0)
0	0	1	0	(többség 0)
0	1	0	0	(többség 0)
0	1	1	1	(többség 1)
1	0	0	0	(többség 0)
1	0	1	1	(többség 1)
1	1	0	1	(többség 1)
1	1	1	1	(többség 1)

A függvényt az igazságtáblázatból legkönnyebben Nem-És-Vagy műveletekkel realizálhatjuk; sorban megnézzük, hogy Y értéke mikor lesz 1-es. Fentről lefelé haladva az első 1-es a 4. sorban van:

A	B	C	Y
.	.	.	.
.	.	.	.
.	.	.	.
0	1	1	1
.	.	.	.
.	.	.	.
.	.	.	.

Eszerint, ha A = 0 ÉS B = 1 ÉS C = 1, akkor Y = 1. Ezt az esetet a többi közül nyilván egy ÉS kapcsolattal (ÉS kapuval) választhatjuk ki, hiszen az A = 0, B = 1, C = 1-nek egyszerre kell teljesülnie, hogy Y = 1 legyen. Mivel ebben a kapcsolatban A = 0, viszont az ÉS kapcsolat akkor ad 1-et, ha egyidejűleg valamennyi változó 1-es, az A értékét negálnunk kell (azért, hogy a 0-ból 1-es legyen). Ezt a sort az igazságtáblázatból tehát a következő ÉS kapcsolattal választhatjuk ki:

A	B	C	Y	
.	.	.	.	
.	.	.	.	
.	.	.	.	
0	1	1	1	$\bar{A} B C,$
.	.	.	.	
.	.	.	.	
.	.	.	.	

Az Y értéke nem csak ebben az egy esetben lesz 1-es, hanem jelenleg még három esetben. Ezeket az eseteket is ÉS kapcsolattal választhatjuk ki az előzők alapján; ha a kérdéses sorban valamely változó 1-es értékkel szerepel, akkor a szorzatba az illető változó eredeti, ponált értékét írjuk be, ha valamely változó 0-ás értékkel szerepel, akkor a negáltját írjuk be.

A	B	C	Y	A	B	C	$\bar{A} B C$	$A \bar{B} C$	$A B \bar{C}$	$A B C$	Y = $\bar{A} B C + A \bar{B} C + A B \bar{C} + A B C$	
0	0	0	0	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	1	0	0	0	0	0	
0	1	0	0	0	1	0	0	0	0	0	0	
0	1	1	1	$\bar{A} B C$	0	1	1	1	0	0	0	1
1	0	0	0		1	0	0	0	0	0	0	
1	0	1	1	$A \bar{B} C$	1	0	1	0	1	0	0	1
1	1	0	1	$A B \bar{C}$	1	1	0	0	0	1	0	1
1	1	1	1	$A B C$	1	1	1	0	0	0	1	1

Mivel Y értéke több esetben is 1-es; ha VAGY az egyik "eset": $\bar{A} B C$ következik be, VAGY a másik: $A \bar{B} C$, VAGY a harmadik, $A B \bar{C}$, VAGY a negyedik: $A B C$, ezért ezeknek az Y = 1 "eseteket" megvalósító logikai szorzatoknak a VAGY kapcsolata, összege adja a logikai függvényt (hiszen, ha bármelyik "eset" bekövetkezik Y = 1 lesz):

$$Y = \bar{A} B C + A \bar{B} C + A B \bar{C} + A B C.$$

A viszonyokat a kibővített táblázat ábrázolja: az Y függvény az $\bar{A} B C \dots$ stb. rész-függvények (szorzatok) VAGY kapcsolatából áll elő.

Az egyenletről azonnal látszik, hogy egyszerűsíteni lehet a BOOLE-algebra szabályainak alkalmazásával. Például az utolsó és az utolsó előtti tagból AB-t kiemelhetünk:

$$A\bar{B}\bar{C} + ABC = AB(\bar{C}+C) = AB, \text{ mert } C + \bar{C} = 1.$$

Hasonlóan a második és az utolsó tagból AC-t emelhetünk ki (felmerül a kérdés, hogy szabad-e az utolsó tagot újra egyszerűsítésre igénybe venni. A válasz, hogy szabad, hiszen

$A = A + A + A \dots$, vagyis bármely tagot annyiszor szerepeltethetünk, ahányszor indokolt):

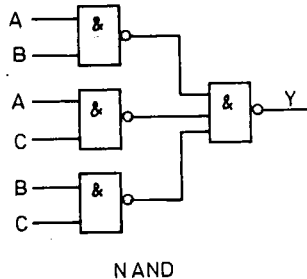
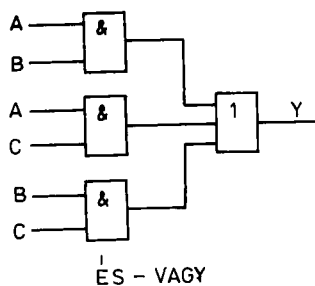
$$A\bar{B}\bar{C} + ABC = AC(\bar{B} + B) = AC$$

Az első és utolsó tag is összevonható, ezekből BC-t emelhetünk ki:

$$\bar{A}BC + ABC = BC.$$

Az egyszerűsített függvény tehát:

$$Y = AB + AC + BC.$$



2.40. ábra

A megvalósító hálózat ÉS-VAGY kapukkal a 2.40. ábrán látható: először az AB, AC, BC, ÉS kapcsolatát állítjuk elő egy-egy két-bemenetű ÉS kapuval, majd ezek kimeneti jelét egy VAGY kapuval "fogjuk össze". A VAGY kapu kimenetén a kitűzött feladatnak megfelelően akkor lesz 1-es, ha a bemenetek többsége (3 közül 2, vagy mind a három) 1-es szinten van. Annak ellenére, hogy a tervezést algebrai módszerrel végeztük, elvonatkoztat-

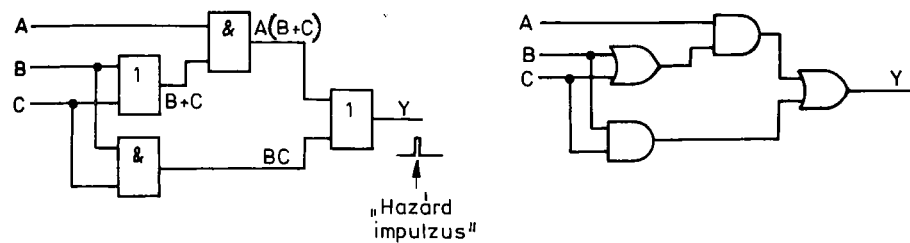
va a fizikai jelentéstől, az eredeti "szöveges feladattól", az eredmény, a logikai ÉS-VAGY hálózat - ha elkészítjük - valószínűleg is működő, minimális (legegyszerűbb) megoldás lesz! A 2.40. ábrán megrajzoltuk a NAND hálózatot is, felhasználva azt a már említett megfigyelést, hogy a NAND hálózat az ÉS-VAGY formából közvetlenül megrajzolható csupán a kapuk "átjelölésével" (az algebrai bizonyítást, a próbát, amelyet a De-Morgan szabály felhasználásával végezhetünk el, az olvasóra bízunk).

A végeredményként kapott

$$Y = AB + AC + BC$$

alaku függvényt - úgy tűnik - még egyszerűbb formában fel lehet írni, egyszerűbb hálózattal meg lehet valósítani, hiszen van még kiemelés, amit végre lehet hajtani az algebrai alakon, pl. az első két tagból az A változót kiemelhetjük

$$Y = A(B + C) + BC.$$



2.41. ábra

A 2.41. ábrán látható a hálózat: a megvalósításhoz ugyanannyi kapu kell, mint az előző esetben (összesen négy), de nincs szükség három bemenetű kapura, mindegyik két bemenetű lehet, ami valóban egyszerűbb megoldásnak számít. Azt viszont látnunk kell, hogy ez utóbbi megoldás nem olyan "szabályos" hálózatot eredményez, mint amilyen a 2.40. ábra ÉS-VAGY hálózata, több "vegyes" kapun halad keresztül a jel (hatása), amíg a bemenettől a kimenetig eljut. Figyeljük meg, hogy esetünkben a B és C jel a hálózat felső részén három kapun halad ke-

resztül, az alsó részen csak kettőn. Mivel minden kapunak (az áramkör típusától függő) véges jelterjedési, késleltetési ideje van, belátható, hogy a B vagy C jel megváltozásakor ennek hatása a két uton végighaladva eltérő idő múlva érezteti hatását a kimeneti jelben. Van tehát egy olyan "pillanat", amikor az egyik uton már "megérkezett" a bemeneti jel hatása, de a másikon még nem, így előfordulhat, hogy átmenetileg nem az igazságtáblázatnak megfelelő jel áll elő a kimeneten: "versenyt futnak" a jelek és a "jó szerencsétől függ", milyen hamis kimeneti tranzienst keletkezik, szaknyelven HAZÁRD JELEN-SÉG lép fel! Mivel ez a házárd impulzus csak nagyon rövid ideig van a kimeneten (csak addig, amíg a hosszabb uton is megérkezik a bemeneti jelek hatása), nagyon nehéz oszcilloszkóppal vagy más műszerrel felderíteni, viszont hatása nagyon zavaró, "titokzatos jelenségek" előidézője lehet. (Gondoljunk arra, hogy ha egy ilyen "hazárdos" kombinációs hálózat egy sorrendi hálózatot vezérel, akkor - mivel ezek általában egész rövid impulzusokra is működnek - teljesen téves működés következhet be annak ellenére, hogy minden egység külön-külön vizsgálva jól dolgozik. Egy számláló pl. lehet, hogy megszámlálja a házárd impulzusokat is, ezért a számlálás eredménye teljesen használhatatlan lesz.)

Előbbiek alapján megállapíthatjuk, hogy célszerűbb a "szabályos" ÉS-VAGY (ill. VAGY-ÉS), kétszintes kapu-hálózatos megvalósítás, mert ennél a bemeneti jelek azonos késleltetéssel jutnak a kimenetre, a működés hazárdmentes (a ma általában használatos CAD: Computer Aided Design = számítógéppel segített tervező programok rutinszerűen házárdmentes hálózatot valósítanak meg). A "szabályos" alakú megvalósítást legtöbbször előnyben részesítjük annak ellenére, hogy - az előző példákhoz hasonlóan - nem nyújt minimális megoldást (nem a legkisebb a kapuk, ill. a kapu bemenetek száma), ha csak nem olyan esetről van szó, amelynél a házárddnak semmilyen jelentősége nincs és fontos a lehető legegyszerűbb megoldás.

2.3.3. Logikai függvények normál alakjai, mintermek, maxtermek

Az előzőek szerint közvetlenül az igazságtáblázat alapján felírt függvény jellegzetes, "szabályos" alakú, például a "szavazó áramkör" esetén:

$$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC.$$

Nincsenek többszörös zárójelek (esetleg "egymásba skatulyázott" zárójelek), több-szintű szorzatok, összegek. A függvények ilyen rendezett, szabályos alakját NORMÁL, kanonikus alak-nak nevezzük. Jelen esetben a függvényt szorzatok összegeként állítottuk elő, az ilyen normál alakot DISZJUNKTIV NORMÁL alak-nak mondjuk (a függvény egy több tagú összeg, azaz diszjunkció). Ha megfigyeljük a példa függvényt, észrevehetjük, hogy az összeg minden tagjában (minden szorzatban) mindhárom változó előfordul. Általában, ha a diszjunktív normál alak mind-egyik tagjában minden egyes változó szerepel, akkor a függvény DISZJUNKTIV TELJES NORMÁL alakban van adva. Az egyszerűsített függvény is normál alakú, de nem teljes normál alakban van:

$$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC \quad \text{diszjunktív teljes normál alak,}$$

$$Y = AB + AC + BC \quad \text{- diszjunktív (nem teljes) normál alak.}$$

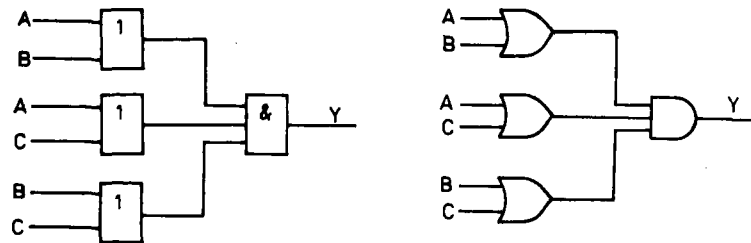
A diszjunktív normál alakból közvetlenül felrajzolható a neki megfelelő ÉS-VAGY hálózat. (ÉS kapuk kimeneti jeleit és közös VAGY kapu bemeneteire vezetjük - l. a 240. ábrát.)

A logikai függvények természetesen nem csak szorzatok összegeként írhatók fel, hanem összegek szorzataként, vagyis KONJUNKTIV NORMÁL alakban is (a függvény több tényező szorzata, konjunkciója), pl.:

$$Y = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+C) \quad \text{- konjunktív teljes normál alak,}$$

$$Y = (B+C)(A+C)(A+B) \quad \text{- konjunktív nem teljes normál alak.}$$

A konjunktív normál alak szintén lehet teljes és nem teljes. A konjunktív normál alakú függvény VAGY-ÉS hálózattal realizálható közvetlenül (VAGY kapuk kimeneti jelei egy közös ÉS kapu bemeneteire mennek, a VAGY kapuk a zárójelekben lévő logikai összegeket valósítják meg, ezek eredményét az ÉS kapu "szorozza össze" (2.42. ábra). Érdemes megjegyezni, hogy a VAGY-ÉS hálózat - lévén az ÉS-VAGY hálózat duálisa - közvetlenül "át rajzolható" NOR-ra, ez a De-Morgan szabály segítségével igazolható, hasonlóan a NAND-hoz.



2.42. ábra

A normál alakok jelentősége igen nagy a logikai függvények, ill. hálózatok egyszerűsítésében, minimalálásában. Éppen a minimalálás érdekében gyakran szükséges, hogy egy nem normál alakban lévő függvényt ekvivalens normál alakban írjunk fel. Minden logikai függvény - megfelelő átalakításokkal, bővítésekkel - előállítható konjunktív és diszjunktív normál alakban. Például írjuk fel a következő függvény kétféle teljes normál alakját!:

$$Y = A(B+C)$$

$$\begin{aligned} A(B+C) &= AB+AC = AB \cdot 1 + AC \cdot 1 = AB(C+\bar{C}) + AC(B+\bar{B}) = \\ &= ABC + AB\bar{C} + ABC + A\bar{B}C = ABC + AB\bar{C} + A\bar{B}C \end{aligned}$$

a diszjunktív teljes normál alak.

$$\begin{aligned} A(B+C) &= (A+0)(0+B+C) = (A+B\bar{B})(A\bar{A}+B+C) = \\ &= (A+B)(A+\bar{B})(A+B+C)(\bar{A}+B+C) = \\ &= (A+B+C\bar{C})(A+\bar{B}+C\bar{C})(A+B+C)(\bar{A}+B+C) = \\ &= (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+B+C), \end{aligned}$$

ez a konjunktív teljes normál alak.

Ehhez hasonló bővítésekkel egészíthetjük ki a függvényeket teljes normál alakúra. Diszjunktív alak esetén a "hiányzó" változókat 1-gyel való szorzással visszük be, konjunktív alak esetén pedig 0 hozzáadásával.

A teljes normál alak jellegzetes összetevő tagjait, ill. tényezőit általánosságban TERM-eknek (meghatározóknak) nevezzük.

A diszjunktív teljes normál alak tagjait (azokat a szorzatokat, amelyekben minden változó vagy "igen", vagy negált értékkel szerepel) MINTERM-eknek nevezzük. Egy minterm a Karnaugh-diagramon minimális területet foglal el (l. a 2.2. fejezetet!).

A konjunktív teljes normál alak tényezői (azok a zárójelben lévő összegek, amelyek minden változót tartalmaznak) a MAXTERM-ek. Egy maxterm, VAGY-kapcsolat lévén, a Karnaugh-diagramon az egységterületnél kisebb maximális területet foglal el.

Ugy is mondhatjuk, hogy a diszjunktív teljes normál alak mintermek összege (mintermes alak), a konjunktív normál alak pedig maxtermek logikai szorzata (maxtermes alak).

Amikor az igazságtáblázatból a logikai függvényt "kiírjuk" szorzat-összeges formában, vagyis az $Y = 1$ sorokból az ismert módon szorzatokat képezünk és ezeket összeadjuk, akkor végeredményben mintermekből állítjuk elő a függvényt. Az igazságtáblázat minden sorából egy-egy minterm képezhető. Amelyik sor mellett 1-es van ($Y = 1$), azt a mintermet szerepeltetjük a függvényben, amelyik mellett 0 van, arra "nincs szükség". Adott n számú változó esetén annyi minterm (és maxterm) van, ahány sor van az igazságtáblázatban, vagyis ahány bináris számot tudunk képezni n jegyből, azaz 2^n darab.

A 2^n mintermet (maxtermet) sorszámokkal (indexekkel) jelölhetjük meg. A számozás úgy történik, hogy a mintermben vagy maxtermben a mindig ABC sorrendben felírt változókat, ill. negáltjukat bináris számjegyeknek tekintjük; a változó igaz értékét 1-esnek, negált értékét 0-nak. Ezek a számjegyek "összeolvasva" adják a minterm vagy maxterm számát, indexét. Az indexet mindig a mintermet jelentő m , ill. a maxtermet jelentő M jobb alsó sarkába írjuk (decimális számmal). A felső sarokban kis számmal jelezzük, hogy hány változós term-ről van szó (egy minterm vagy maxterm egészen más módon, ha pl. 2, vagy ha 3 változós). Egy n változás, i -edik minterm, ill. maxterm jelölése:

$$m_i^n \quad M_i^n$$

Például a 2 és a 3 változós minterm és maxtermek a következők:

	A B	minterm:	MAXterm:	A B C	m_i^3	M_i^3
0.	0 0	$m_0^2 = \bar{A} \bar{B}$	$M_0^2 = \bar{A} + \bar{B}$	0. 0 0 0	$m_0^3 = \bar{A} \bar{B} \bar{C}$	$M_0^3 = \bar{A} + \bar{B} + \bar{C}$
1.	0 1	$m_1^2 = \bar{A} B$	$M_1^2 = \bar{A} + B$	1. 0 0 1	$m_1^3 = \bar{A} \bar{B} C$	$M_1^3 = \bar{A} + \bar{B} + C$
2.	1 0	$m_2^2 = A \bar{B}$	$M_2^2 = A + \bar{B}$	2. 0 1 0	$m_2^3 = \bar{A} B \bar{C}$	$M_2^3 = \bar{A} + B + \bar{C}$
3.	1 1	$m_3^2 = A B$	$M_3^2 = A + B$	3. 0 1 1	$m_3^3 = \bar{A} B C$	$M_3^3 = \bar{A} + B + C$
				4. 1 0 0	$m_4^3 = A \bar{B} \bar{C}$	$M_4^3 = A + \bar{B} + \bar{C}$
				5. 1 0 1	$m_5^3 = A \bar{B} C$	$M_5^3 = A + \bar{B} + C$
				6. 1 1 0	$m_6^3 = A B \bar{C}$	$M_6^3 = A + B + \bar{C}$
				7. 1 1 1	$m_7^3 = A B C$	$M_7^3 = A + B + C$

A négyváltozós függvények 16 termje, az ötváltozós függvények 32 termje stb. hasonlóképpen írható fel.

Mivel a logikai függvények teljes normál alakja mintermek összegéből vagy maxtermek szorzatából áll, a függvények megadhatók mintermekkel, vagy maxtermekkel is, a teljes BOOLE kifejezés felírása helyett. A szorzatok, ill. összegek helyett csak a függvény előállításában szereplő megfelelő indexű termeket írjuk le, például:

$$Y = A \oplus B = \bar{A}B + A\bar{B} = m_1^2 + m_2^2 \quad Y = A \odot B = \bar{A}\bar{B} + AB = m_0^2 + m_3^2$$

$$A \oplus B = (A+B)(\bar{A} + \bar{B}) = M_0^2 \cdot M_3^2 \quad A \odot B = (\bar{A} + B)(A + \bar{B}) = M_1^2 \cdot M_2^2$$

Az előzőekben szerepelt példa függvényünk (3 változós szavazó áramkör) mintermes formában:

$$Y = m_3^3 + m_5^3 + m_6^3 + m_7^3 \quad (= \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC).$$

A felírást még tovább rövidíthetjük, ha az "m", ill. "M" betűket sem írjuk le, hanem csupán műveleti jellel utalunk arra, hogy mintermes (összeges) vagy maxtermes (szorzatos) alakról van-e szó, és a műveleti jel után zárójelben felsoroljuk a függvény előállításában szereplő mintermek, ill. maxtermek sorszámát (indexét). A mintermes, összeges felírás műveleti

jele a szumma: Σ , a maxtermes, szorzatos felírás jele a produktum: Π . A szumma, ill. produktum jele fölött feltüntetjük, hogy hány változós függvényről van szó, például:

$$\text{mintermes alakok: } Y = A \oplus B = m_1^2 + m_2^2 = \Sigma (1,2)$$

$$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC = m_3^3 + m_5^3 + m_6^3 + m_7^3 = \Sigma (3,5,6,7)$$

$$\text{maxtermes alakok: } Y = A \oplus B = M_0^2 \cdot M_3^2 = \Pi (0,3)$$

$$Y = (\bar{A} + \bar{B} + \bar{C})(\bar{A} + B + C)(A + B + C) = M_0^3 \cdot M_3^3 \cdot M_7^3 = \Pi (0,3,7).$$

Általánosságban: a logikai függvények egyféleképpen írhatók fel diszjunktív teljes normál alakban, aminek általános formája a következő:

$$Y = \sum_{i=0}^{2^n-1} f_i m_i^n \quad \text{ahol } f_i \begin{cases} 0 \\ 1 \end{cases} \text{ lehet.}$$

Amelyik minterm "részt vesz" a függvény előállításában, annak f_i együtthatója 1-es, amelyik nem, annak 0 (mert $1 \cdot m_i = m_i$ és $0 \cdot m_i = 0$). Minden minterm-nek van egy f_i együtthatója.

Például az

$$Y = \Sigma (3,5,6,7) \text{ függvény a következő mintermek összege:}$$

$$Y = m_3^3 + m_5^3 + m_6^3 + m_7^3, \text{ ez együtthatókkal részletezve:}$$

$$Y = 0 \cdot m_0^3 + 0 \cdot m_1^3 + 0 \cdot m_2^3 + 1 \cdot m_3^3 + 0 \cdot m_4^3 + 1 \cdot m_5^3 + 1 \cdot m_6^3 + 1 \cdot m_7^3$$

tehát:

$$f_3 = f_5 = f_6 = f_7 = 1$$

$$f_0 = f_1 = f_2 = f_4 = 0.$$

Az f együtthatók megegyeznek az igazságtábla Y oszlopában lévő 0-1 függvényértékekkel:

A	B	C	Y
0	0	0	0 ← f ₀
0	0	1	0 ← f ₁
0	1	0	0 ← f ₂
0	1	1	1 ← f ₃
1	0	0	0 ← f ₄
1	0	1	1 ← f ₅
1	1	0	1 ← f ₆
1	1	1	1 ← f ₇

Az

$$Y = \sum_{i=0}^{2^n-1} f_i m_i^n$$

függvény negáltja általános formában, mintermes alakban

$$\bar{Y} = \sum_{i=0}^{2^n-1} \bar{f}_i m_i^n$$

vagyis azok a mintermek alkotják a negált függvényt, amelyek együtthatója: $f_i = 0$.

A konjunktív normál alak, maxtermes függvényalak igazságtáblázatból való felírása a De-Morgan szabály alkalmazásával, kétszeres negálással történhet. Ha egy mintermekkel (szorzatok összegével) felírt függvényt negálunk, akkor maxtermes alakot kapunk, hiszen a "műveletek negálásakor" a részletszorzatokból részletösszegek (maxtermek) lesznek, az összeges normál alakból pedig szorzatos alak keletkezik. Egy adott függvényt csupán egyszer nem negálhatunk, mert akkor nem a függvénynek, hanem a negáltjának kapnánk meg a konjunktív alakját. Az eredeti érték megtartásához kétszer kell negálnunk:

- először az Y függvény negáltját állítjuk elő mintermes formában úgy, hogy az igazságtáblázatból a nem az 1-es, hanem az Y = 0-ás sorokat "olvassuk ki". Példánknál maradv:

$$\bar{Y} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C},$$

- ezt a mintermes alakot negálva (a függvényt másodszer negálva) a De-Morgan szabály felhasználásával megkapjuk a függvény maxtermes formáját:

$$\bar{\bar{Y}} = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}} = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+C) =$$

$$Y = M_7^3 \cdot M_6^3 \cdot M_5^3 \cdot M_3^3 = \pi^3(3,5,6,7).$$

A kapott konjunktív normál alakot és az igazságtáblázatot összehasonlítva megállapíthatjuk a maxtermes alak felírási szabályát:

Az igazságtáblázatból a 0-kat olvassuk ki, vagyis az Y=0 sorokból képezünk maxtermeket oly módon, hogy a változók negált értékét írjuk be a maxtermekbe (ahol a változó 1-gyel szerepel, ott a változó negáltját). A függvényt az összes Y = 0 sorból képezett maxterm logikai szorzata adja. (Ahogy az előbb láttuk, a 0-ák kiolvasására azért van szükség, hogy a függvény negáltját írjuk fel először mintermes formában, a változókat azért kell negálva kiolvasni, mert a második negáláskor, a De-Morgan szabály alkalmazásakor nem csak a műveletek, hanem a változók is negálódnak.)

A maxtermes függvény igazságtáblázatból történő felírását az általános algebrai alakon is végigkövethetjük:

- a függvény negáltjának felírása diszjunktív normál alakban:

$$\bar{Y} = \sum_{i=0}^n \bar{f}_i m_i^n$$

vagyis a 0-ás sorokat olvassuk ki.

- Második negálás a De-Morgan szabály felhasználásával

$$\bar{\bar{Y}} = Y = \overline{\sum_{i=0}^n \bar{f}_i m_i^n} = \prod_{i=0}^n (f_i + \bar{m}_i^n) = \prod_{i=0}^n (f_i + M_{2^n-1-i}^n).$$

Negálás során a szummából produktum lett és a rész szorzatokból (mintermekből) negálás után rész-összegek (maxtermek) lettek. Az index azért változott meg, mert egy i-edik minterm negáltja

$$\bar{m}_i^n = M_{2^n-1-i}^n$$

egy "1-es komplementum" indexű maxterm (például: $m_0^3 = \overline{ABC} = A+B+C = M_7^3$... és így tovább).

Ebben a konjunkcióban azok a rész-összegek "maradnak életben" és vesznek részt a függvény előállításában, amelyek mellett $f_i = 0$, hiszen $f_i = 1$ esetén az

$$1 + A \equiv 1$$

azonosság érvényesül, tehát a kapott végeredményből (példánkánál maradva):

$$\begin{aligned} Y &= \prod_{n} (f_i + M_{2^n-1-i}^3) = \\ &= (0+M_7^3)(0+M_6^3)(0+M_5^3)(1+M_4^3)(0+M_3^3)(1+M_2^3)(1+M_1^3)(1+M_0^3) = \\ &= M_3^3 \cdot M_5^3 \cdot M_6^3 \cdot M_7^3 = \prod (3,5,6,7) \end{aligned}$$

formája lesz a függvény (vagyis végeredményben az $f_i = 0$ -hoz tartozó sorokat olvastuk ki az igazságtáblázatból, a változók - mintermek - tagadott értékeivel).

2.4. Minimálási módszerek

2.4.1. A minimális kombinációs hálózat

A kombinációs hálózatok - eddigiekben megismert - matematikai leírásának legtöbbször az a célja, hogy exakt módon megtaláljuk a legegyszerűbb, minimális megoldást (kivételt képeznek azok az esetek, amelyekben pl. a működés ellenőrzését, elemzését tüzzük ki célul vagy valamely áramkör család adott típusaival való megvalósítás érdekében alakítjuk át a függvényt).

A minimális, legegyszerűbb hálózat kritériumát, ha meggondoljuk, nem is olyan egyszerű meghatározni. Azt mondhatnánk, hogy a legegyszerűbb megoldás az, amely a legkevesebb kaput tartalmazza. Természetesen a minimális kapu-szám mellett az is fontos, hogy minél kevesebb kapu bemenetre legyen szükség. A dolog azonban nem ilyen egyszerű. Hiába tervezünk pl.

egy akármilyen egyszerű logikai hálózatot olyan kapukból, amelyek nincsenek benne az illető áramköri rendszer típusválasztékában - akár azért, mert olyan fajta kapu nincs is, akár mert az előírt bemenet-számú kapu nem létezik (pl. a TTL-ben nincs hatbemenetű NOR kapu). A típusválaszték figyelembe vétele tehát a saját érdekünkben nagyon fontos. Ezenkívül egy adott feladat áramköri realizálásakor mindenekelőtt meg kell győződnünk arról, hogy valóban fel kell-e kapukból építenünk, a kérdéses hálózatot, nincs-e a felhasznált áramkör családban olyan "kész" MSI vagy LSI (MSI=Medium Scale Integrated - közepesen integrált, LSI=Large Scale Integrated - nagymértékben integrált) áramkör, amely kimondottan az adott célra készült. Az is lehet, hogy nincs a célunknak kifejezetten megfelelő típus, de néhány MSI tok egyes összekapcsolásával nagyszámú kaput helyettesíthetünk. Erre nincs szisztematikus tervezésmódszer. Nagyobb sorozatszámú készülékek, berendezések célszerű építőeleme lehet az LSI kategóriába sorolható programozható, logikai áramkörök ma már népes családjának valamelyik tagja: pl. a PLA (Programable Logic Array = programozható logikai elrendezés), az ULA (Uncommitted Logic Array = előre meg nem határozott logikai elrendezés) vagy pl. az ún. CUSTOM IC (a felhasználó által "megrendelt" IC, nálunk a Berendezés Orientált Áramkörök, BOÁK a leggyakrabban használt elnevezés) - minderről a későbbiekben részletesen szó lesz. Mindegyik típus közös jellemzője, hogy nagyszámú kaput és sokszor egyéb, akár lineáris áramkört integrálnak le egyetlen IC-ben (az alkatelemek száma természetesen véges, így az egyszerűsítésnek itt is jelentősége van). Az összekapcsolást, a kapcsolási rajzot (legtöbbször előírt kódolással) a felhasználó adja meg a gyártónak.

Az egyszerűsítésben sok más szempont is szerepet játszik. A huzalozás lehetőleg legyen minél egyszerűbb a tokok és a tokokat hordozó áramköri lemezek között, a parazita hatások, a jelkésleltetési idő csökkentése, valamint a költségek csökkentése érdekében. Az energia fogyasztás lehetőleg kicsi legyen a kellő működési sebesség biztosítása mellett. Az energia fogyasztás és sebesség egymásnak legtöbbször ellentmondó követelmények. Különböző áramkör családokra egymástól el-

térő sebesség és fogyasztás a jellemző. A gyorsabb bipoláris (NPN - PNP tranzistorokból felépített) áramkörök általában többet fogyasztanak, ilyen pl. az ECL (Emitter Complded Logic = emittercsatolt logika) vagy a TTL (Transistor Transistor Logic). A tervezérlésű (NMOS, PMOS, CMOS: komplementer MOS) családok sokszor extrém kislevegysztásuak (CMOS = 10 nW kapunként), de lassubbak. A célnak megfelelő áramkör családot rendszerint már a tervezés első fázisában kiválasztjuk. A kis energia fogyasztáson kívül fontos még a megbízhatóság, sokszor a kis helyfoglalás stb.

A végső cél az esetek többségében az, hogy az adott rendszer minimális költséggel hozzuk létre és üzemeltessük - az előírások és követelmények maradéktalan teljesítése mellett. A minimális költség elérésének sok feltétele van - ezek közül említettünk az imént néhányat, de valamennyi összetevő matematikai formába öntése szinte lehetetlen. Így talán közelebb járunk az igazsághoz, ha azt mondjuk, hogy egy adott feladatot megvalósító hálózat akkor optimális, ha minimális számú áramköri elemet (IC tokot) használ fel. Leegyszerűsítve, a kombinációs hálózatok esetében ez azt jelenti, hogy a minimális hálózat feltétele, hogy minimális számú kaput tartalmazzon. A minimális számú kaput tartalmazó változatok közül azt tekintjük minimális hálózatnak, amelyben a legkevesebb kapu bemenet van, ill. amely a rendelkezésre álló típusválaszték figyelembe vételével áramkörileg a legegyszerűbben valósítható meg. A minimális kapu és kapu bemenetre vonatkozó szabály természetesen nem érvényes változatlan formában az MSI és LSI áramkörökkel való építkezés esetén, ilyenkor nem "számoljuk" a kapukat (bár ilyenkor sem közömbös, hogy egy adott feladathoz mekkora IC chip-területet használunk fel).

A tervezés során a logikai függvényeket általában normál alakban írjuk fel és egyszerűsítjük és az eredményt is normál alakban kapjuk (emlékezzünk: a normál alakból realizált hálózat lesz hazárdmentes!). Egy logikai függvény az előzőeknek megfelelően akkor minimál alakú, ha benne minimális számú tag vagy tényező van és ezekben a tagokban, ill. tényezőkben minimális számú változó fordul elő, egyetlen egyet sem lehet elhagyni anélkül, hogy a függvény meg ne változna. Természete-

sen többféle minimál ekvivalens létezik. Ezek közül azt választjuk, amelyet áramköri szempontból legegyszerűbben meg tudunk valósítani. A továbbiakban az adott feladat alapján felírt függvény átalakítása, egyszerűsítése során mindig ilyen minimál alakú függvény előállítására lesz a cél. Magát az egyszerűsítő eljárást, amelynek eredményeképpen a minimál alakot kapjuk meg minimálásnak nevezzük:

- Az algebrai úton való minimálást mondhatjuk a "hagyományos" módszernek; a logikai függvényen sorozatosan addig végzünk azonos átalakításokat, amíg a kívánt alakú, egyszerűsített formához eljutunk. Ennek nagy hátránya - amint arról valószínűleg már meggyőződünk - hogy nagyon munkaigényes, sok a tévedési lehetőség. (Ha kb. 5%-os hibavalószínűséggel dolgozunk, akkor 20 művelet elvégzése után valószínű, hogy már egy hibát ejtettünk, ami megkérdőjelezi az egész tevékenység értelmét). Sok összefüggést, azonosságot kell fejben tartani és felismerni, a kapott végeredményről nem tudhatjuk biztosan, hogy a legegyszerűbb-e és a több ekvivalens eredmény közötti választásig sokszor el sem jutunk.

- A grafikus minimálás a BOOLE-algebrai átalakításokon alapuló minimálásnál sokkal biztosabb, áttekinthetőbb és kevesebb munkát igénylő módszer, amely a már bemutatott KARNAUGH-táblával végezhető. A KARNAUGH-tábla nem más, mint egy célszerűen átrendezett igazságtábla, amelyen az egyszerűsítési lehetőségek szinte első ránézésre láthatók és a tévedés valószínűsége is sokkal kisebb. Hátrány viszont, hogy 4-nél több változó esetén már nehezkessé válik a kezelése. A KARNAUGH-tábla a minimáláson kívül gyors és szemléletes ellenőrzésre, sorrendi áramkörök állapotainak ábrázolására is hasznos segéd-eszköz, ezért ezzel foglalkozunk részletesen.

- Egyéb, főként táblázatos módszerekhez akkor érdemes folyamodni, amikor sok (négy-öttnél több) változós függvényrel dolgozunk vagy amikor algoritmust kívánunk készíteni az egyszerűsítési művelethez, lehetővé téve a számítógépes feldolgozást. Legismertebb a Quine-McCluskey eljárás, ennek lényegével is megismerkedünk.

2.4.2. Az algebrai egyszerűsítés

Annak érdekében, hogy egyszerűbb vagy célunknak megfelelő alakú kifejezést kapjunk, a logikai függvényen azonos átalakításokat végzünk, felhasználva a BOOLE-algebra alaptételeit, szabályait, amelyeket a 2.2.4. fejezetben áttekintettünk; pl. a tagadás, ill. a kettős tagadás törvényét, a De-Morgan szabályt, a logikai összeadás és szorzás kommutativitását, asszociativitását, disztributivitását, az abszorpciós törvényeit, a dualitás törvényét stb. A következőkben néhány példán bemutatjuk a tételek, szabályok alkalmazását. A példák egy része azonosság, amelyek bizonyításából okulhatunk, és amelyek később, "bonyolultabb" esetekben - már, mint bizonyított tételek - jól alkalmazhatók.

- A "kiemelés"-sel való egyszerűsítés alap esetei (amely a "zárójel beszorozhatóság" előzőkben felírt sorrendjének felcserélése, vagyis a disztributivitás felhasználása):

$$\begin{aligned} A + AB &= ? & A(A+B) &= ? \\ A + AB &= A(1 + B) = A \cdot 1 = \underline{A} & A(A+B) &= \underline{A} \text{ (a dualitás miatt)} \\ AB + A\bar{B} &= ? & (A+B)(A+\bar{B}) &= ? \\ AB + A\bar{B} &= A(B+\bar{B}) = A \cdot 1 = \underline{A} & (A+B)(A+\bar{B}) &= \underline{A} \text{ (dualitás).} \end{aligned}$$

A "kiemelés" módszerét használjuk leggyakrabban egyszerűsítésre, kifejezések "rövidítésére". Alapelv, hogy a "szomszédos" tagok (tényezők) közös része kiemelhető, a megmaradó rész pedig "kiesik". "Szomszédos"-nak akkor mondunk két tagot (vagy tényezőt), ha bennük ugyanazok a változók szerepelnek, de egy és csak egy változó az egyikben ponált, a másikban negált értékű, pl.:

$$\begin{aligned} ABC + A\bar{B}\bar{C} &= AB(C+\bar{C}) = AB \cdot 1 = \underline{AB} \text{ vagy ennek duálja:} \\ (A+B+C)(A+B+\bar{C}) &= \underline{A + B}. \end{aligned}$$

Mindegyik esetben a szomszédos kifejezésekben levő eltérő állapotú változó egyszerűsíthető, "tüntethető el". (Vigyázat! A kiemelésnek az a módja csak szomszédos tagok, ill. tényezők esetén hajtható végre; két tag, ill. tényező össze-

vonásakor egy változó "tűnik el"!). A későbbiekben, függvények egyszerűsítésekor, amikor szomszédos tagokat találunk, az összevonást "automatikusan", gyorsítottan hajtjuk végre az eltérő (egyikben ponált, másikban negált) változó elhagyásával. (Emlékezzünk: a "szavazó áramkör" tervezésekor ugyanezt a módszert követtük, sőt egy tagot többször is felhasználtunk, mivel több egyszerűsítendő taggal is szomszédosnak bizonyult.)

- Bizonyítandó:

$$A + \bar{A}B = A + B \quad (\text{vagyis az első tag negáltja elhagyható a második tagból}).$$

Bizonyításkor az egyenletnek csupán az egyik oldalát alakítjuk addig, amíg a másik oldallal egyező nem lesz.

Bizonyítás a disztributív tulajdonság (összegezés tényezőnként való elvégezhetősége) alapján, az eredetileg $A + BC = (A + B)(A + C)$ alakban felírt azonosság felhasználásával:

$$\begin{aligned} A + \bar{A}B &= A + B \\ (A + \bar{A})(A + B) &= A + B \\ 1 \cdot (A + B) &= A + B \\ \underline{A + B} &= \underline{A + B} \end{aligned}$$

Bizonyítás kétszeres negálással, a De-Morgan szabály alkalmazásával (gyakran célravezető módszer):

$$\underline{\underline{A + \bar{A}B}} = \underline{\underline{\bar{A} \cdot \bar{A}\bar{B}}} = \underline{\underline{\bar{A} \cdot (A + \bar{B})}} = \underline{\underline{\bar{A}A + \bar{A}\bar{B}}} = \underline{\underline{0 + \bar{A}\bar{B}}} = \underline{\underline{A + B}}$$

Bizonyítás a dualitás tételével (ha egy egyenlet duálja igaz, akkor az eredeti is igaz):

$$\begin{aligned} A + \bar{A}B &= A + B & \text{duálja az} \\ A \cdot (\bar{A} + B) &= AB & \text{egyenlet, ez pedig igaz, mert} \\ \underline{\underline{A\bar{A} + AB}} &= \underline{\underline{AB}} \end{aligned}$$

- Egyszerűsítendő:

$$AC + BC + \bar{A}B = ?$$

Ez az eset arra példa, hogy első pillanatban nem vehető észre az egyszerűsítési lehetőség, ilyenkor a függvényt inkább bővítünk kell, valamelyik tagot vagy akár többet is úgy kell

kiegészítenünk, hogy ugyanaz a változó (ponált vagy negált értékkel) mindegyik tagban szerepeljen. Jelen esetben a középső tagot célszerű A-val bővítenünk - természetesen úgy, hogy a kifejezés azonos maradjon:

$$AC + (A+\bar{A})BC + \bar{A}B = AC + ABC + \bar{A}BC + \bar{A}B.$$

Igy már láthatók a kiemelési lehetőségek:

$$AC + ABC + \bar{A}BC + \bar{A}B = AC(1 + B) + \bar{A}B(1 + C) \quad \text{vagyis végeredményben:}$$

$$AC + BC + \bar{A}B = \underline{AC + \bar{A}B} \quad \text{tehát az eredeti függvényben a középső BC tag "felesleges" - ez az, amit "ránézésre" nélkül felfedezni.}$$

- Bizonyítandó:

$$(A + B)(\bar{A} + C) = \bar{A}B + AC.$$

Az egyik tényezőben az A változó, a másikban a negáltja szerepel. Ha az egyenlőség igaz, akkor ilyen esetekben nem kell valamennyi tagot valamennyi taggal összeszoroznunk, hanem "rossz diák módjára" csak a "beltagokat" és a "kültagokat".

$$(A + B)(\bar{A} + C) = \cancel{A\bar{A}} + \bar{A}B + AC + BC \quad \text{a fennmaradó rész éppen egyezik az előző példával, így:}$$

$$(A + B)(\bar{A} + C) = \underline{\bar{A}B + AC} \quad \text{tehát igaz a "rossz diák szorzat" (vigyázat! csak akkor, ha ugyanazon változó ponált-negált értéke fordul elő egy-egy tényezőben).}$$

- Egyszerűsítendő:

$$\bar{A} + B + \bar{C} + \bar{A}BC = ?$$

Példa arra, hogy mielőtt a részletesebb kifejtéshez hozzáfekcendénk, érdemes áttekintően szemügyre venni a teljes kifejezést; ebben az esetben észrevehető, hogy az egyik rész a má-

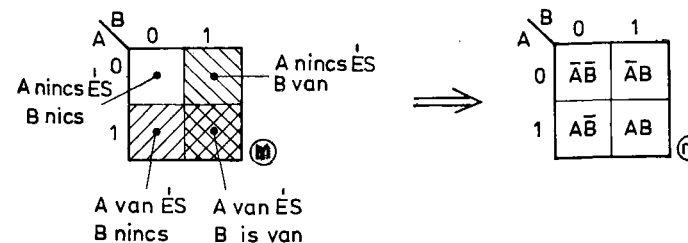
sik negáltja, így VAGY kapcsolatuk végeredményként 1-et ad:

$$\bar{A} + B + \bar{C} + \bar{A}BC = \bar{A}BC + \bar{A}BC = \underline{1}.$$

A bemutatott néhány típus-példával áttekintésünk nem lehet teljes, további gyakorlásra feltétlenül szükség van (gyakorlatok, szakirodalom alapján). Az azonos átalakítások végrehajtása közben ne tévesszük szem elől, hogy az alapösszefüggések és a most bemutatott összefüggések nemcsak egyedi változókra igazak, hanem több változóból álló tagokra, tényezőkre, kifejezésekre is. A lehetséges egyszerűsítések felismerése csak kelendő gyakorlattal, megfelelő intuícióval lehetséges.

2.4.3. A grafikus egyszerűsítés

Említettük előnyét: gyorsabb, biztosabb eredményt adó, szemléletes, kevesebb munkát igénylő módszer, érdemes vállalni a megismerésével járó egyszeri többletmunkát. Jelenleg a KARNAUGH-táblás egyszerűsítés a legelterjedtebb, ezzel foglalkozunk. A KARNAUGH-táblával már az alapműveletek tárgyalásakor megismerkedtünk (2 változóig); a változók és negáltjuk területét a széleken levő, változókhoz tartozó 0-1 kódolás jelöli ki.



2.43. ábra

Az emlékeztetőnek felrajzolt 2.43. ábrán a kétváltozós min-term (m) tábla látható; mind a négy terület-résznek egy-egy minterm felel meg aszerint, hogy az 1-gyel megjelölt sorban, vagy oszlopban a változó ponált, ill. 0-val megjelölt sorban vagy oszlopban a változó negált értéke "van jelen", vagyis mindegyik mintermnek egy és csak egy helye van a minterm táb-

lázatban. Bármely minterm sorszámát egyszerűen a megfelelő sor és oszlop koordináták számjegyeinek "összeolvasásával" kaphatjuk meg binárisan (az összeolvasást természetesen a változók ábécé sorrendjében kell végeznünk - emlékezzünk a mintermek számozási, index-képzési szabályára - 2.44. ábra).

A \ B	0	1
0	00	01
1	10	11

A \ B	0	1
0	m_0^2	m_1^2
1	m_2^2	m_3^2

2.44. ábra

Kettőnél több változó esetére a számozáshoz többféle feltételt kell teljesítenünk. Egyrészt követelmény, hogy minden mintermnek egy és csak egy helye legyen a KARNAUGH-táblában, megtartva a koordináták szerinti számozás, ill. helykijelölés elvét. Másrészt, azért, hogy a táblát egyszerűsítésre lehessen használni a változók területeit úgy kell kijelölni, hogy az egymás melletti mintermek szomszédosak legyenek, vagyis az egymás melletti sorok vagy oszlopok csak egyetlen változóban térjenek el egymástól (nem fordulhat elő, hogy egyik sorban két változó ponált értékű, a mellette lévő sorban pedig mindkettő negált értékű!). Ezen elvek alapján a háromváltozós KARNAUGH-tábla, amelyben 8 rekesz van, a 2.45. ábra szerinti lehet. Mivel most 4 oszlop van, ezeket egy-egy számpárral jelöljük, amelyek első jegye természetesen az először felírt, jelen esetben B változóra, második jegye az utána írt C változóra vonatkozik. A szomszédos oszlopok (és sorok) csak egy változóban különbözhetnek egymástól (még a két szélső is), ezért a KARNAUGH-tábla sorait és oszlopait a GRAY-kód sorrendjében számozzuk (hiszen az egylépéses GRAY-kód tulajdonsága az, hogy a szomszédos számok csak egyetlen helyiértéken különböznek egymástól!).

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

← Nem bináris sorrend! (GRAY)

2.45. ábra

A négyváltozós KARNAUGH-tábla, amelynek 16 rekesze van, az előzőekkel egyezésben a 2.46. ábra szerinti. A rekeszek sarkába beirtuk a széleken lévő számjegyek összeolvasásából adódó minterm számokat. (Tanulás közben célszerű a kettő-, három-, négyváltozós KARNAUGH-táblákat hely számokkal együtt fejből is lerajzolni!)

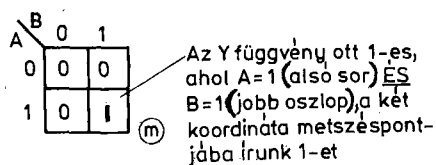
CD \ AB	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

2.46. ábra

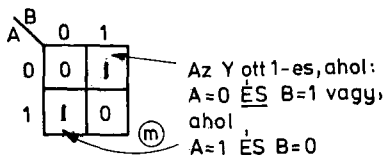
A függvényeket a KARNAUGH-minterm táblán úgy ábrázoljuk, hogy a függvény előállításában résztvevő mintermek rekeszeibe 1-et, a többi rekeszbe 0-át írunk. Végeredményben a f_i együtt-hatókat kell beírniuk a megfelelő (i-edik) minterm helyére. Ha összehasonlítjuk a KARNAUGH-táblát az igazságtáblázattal, akkor észrevehetjük a közeli rokonságot; az igazságtábla bal oldala, amely a változók érték-kombináció-sorait tartalmazza, a KARNAUGH-tábla szélein, a függőleges és vízszintes oldalakon van jelen (csak éppen koordináta-rendszer szerű elosztásban és nem bináris, hanem GRAY-kódban), az igazságtábla jobb oldala, Y oszlopa, amely az f_i függvényértékeket tartalmazza, a KARNAUGH-tábla megfelelő rekeszébe írt 0-ák és 1-ek formájában található meg. Ezért könnyű a KARNAUGH-tábla használata; a feladatot megadó igazságtábla könnyedén átültethető minterm táblába.

Példaképpen a 2.47. ábrán felrajzoltuk a kétváltozós ÉS, valamint kizáró-VAGY függvény minterm táblás ábrázolását. Sokszor, ha nincs szükség rá, a 0-kat nem is írjuk be a táblázatba, hiszen az 1-esek egyértelműen meghatározzák a függvényt. A többször idézett "szavazó áramkör" ("majoritás áramkör") függvényének minterm táblája így a 2.48. ábra szerinti.

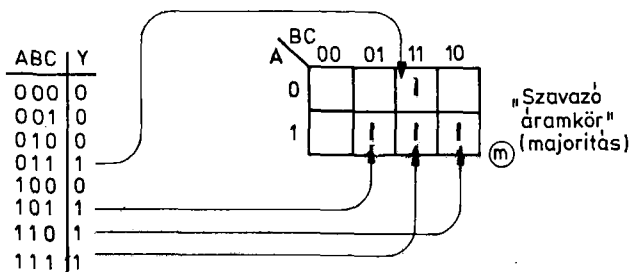
Y = AB		
AB	Y	
00	0	
01	0	
10	0	
11	1	



Y = A ⊕ B		
AB	Y	
00	0	
01	1	
10	1	
11	0	

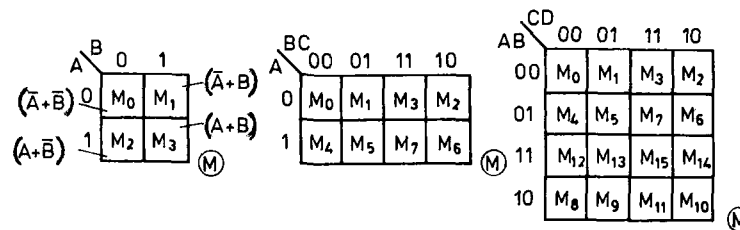


2.47. ábra



2.48. ábra

A MAXTERM tábla (M) elve azonos a minterm táblával. Ebbe is annyi rekeszt rajzolunk, ahány maxterm van adott változó esetén (2^n). A széleken a változókat hasonlóan jelöljük ki és ezzel mindegyik maxterm helyét rögzítjük a táblában (lásd a 2.49. ábra kettő-, három-, és négyváltozós tábláit, a nagy M a maxterm-et jelöli). A maxterm tábla megszerkesztéséhez arra a szabályra kell emlékeznünk, amelyet a maxtermes alak felírásáról tanultunk (2.3.3. fejezet). Eszerint a függvényt kétszer kell tagadnunk: az igazságtáblázatból az Y = 0 sorokat kell kiolvasnunk, valamint az egyes sorokban a kiolvasáskor a változókat is tagadnunk kell (a 0-val szereplő változókat IGEN értékűnek, az 1-gyel szereplő változókat NEGÁLT értékűnek kell vennünk). A sorokból így módon képzett MAXTERMÉK szorzata adja a függvényt. Ennek alapján készíthetjük el a függvények Maxterm tábláját is. Ha az igazságtábla adva van, akkor cél-



2.49. ábra

szerü első lépésben a minterm táblát megrajzolni, majd ebből áttérni a maxterm táblára.

A minterm táblából maxterm táblára átrajzolás szabálya az előbbieket alapján:

1. A széleken a változókat negáljuk, azaz a számozásban 0-1 cserét hajtunk végre.

2. A tábla belsejében is felcseréljük az 1-eseket és 0-akat, vagyis egyszerűen a minterm tábla valamennyi számjegyét (a széleken és a belsejében is) negálnunk kell.

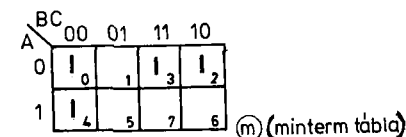
A maxterm tábláról minterm táblára való áttérés szabálya is természetesen ugyanez.

Például rajzoljuk meg az

$$Y = \sum_{(0,2,3,4)}^3$$

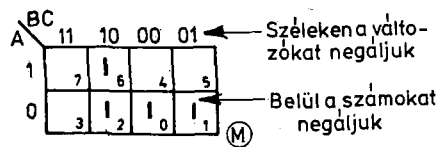
függvény maxterm tábláját!

Először felrajzoljuk az adott diszjunktív alaknak megfelelő minterm táblát (2.50. ábra):



2.50. ábra

Ebből kétszeres negálással rajzoljuk meg a maxterm táblát (2.51. ábra):



2.51. ábra

A függvény maxtermes alakja ennek alapján:

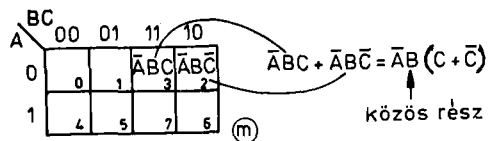
$$Y = \prod (0,1,2,6)$$

A "szereplő" maxtermek sorszámát a maxterm tábla 1-esei koordinátáinak összeolvasásából kapjuk meg.

A maxterm tábla megrajzolásához természetesen nem kell feltétlenül minterm táblát rajzolni kiindulásul. Ha például egy függvény konjunktív alakja adott, akkor abból közvetlenül felrajzolható a maxterm tábla. Az igazságtáblázatból is rajzolhatunk közvetlenül maxterm táblát némi gyakorlattal, a minterm tábla "kihagyásával". Egy KARNAUGH-tábla megrajzolásakor mindig célszerű megjelölni, hogy minterm (m) vagy maxterm (M) tábla-e, mert ez egyébként nem látszik róla.

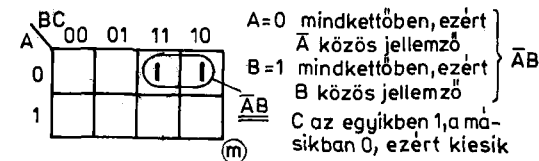
Egyszerűsítési szabályok

Az egyszerűsítés alapelve a szomszédos term-ekből a közös tényező (közös tag) kiemelése. A KARNAUGH-táblák szélein a változókat éppen e célból jelöltük ki (Gray-kódban való számozással) úgy, hogy a szomszédos sorok és oszlopok csak egyetlen változóban különbözzenek egymástól. Így a minterm táblában bárhol, két egymás melletti (egymás alatti)rekeszben olyan mintermek vannak, amelyek "majdnem egyformák", csupán egy változó fordul elő bennük eltérő értékkel (az egyikben ponált, a másikban negált formában). Például az egymás melletti 2-es 3-as minterm csak a C változóban tér el egymástól (2.52. ábra).



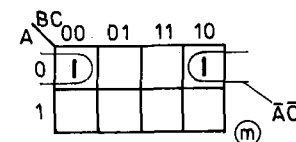
2.52. ábra

Az a változó, amelyik az egyik mintermben PONÁLT értékkel, a másikban NEGÁLT értékkel fordul elő, a mintermek összegezésékor "kiesik", és a két minterm közös része marad meg. Ha tehát valamely függvényben a 2-es és a 3-as minterm is szerepel (a minterm táblában a 2-es és a 3-as rekeszben 1-es van), akkor ezek összevonhatók. Az összevonást úgy jelöljük, hogy az illető 1-eseket egy hurokkal közrefogjuk és melléírjuk az összevonás eredményét, a "rövidített" szorzatot (un. prim-implikáns-t - lásd a 2.53. ábrát).



2.53. ábra

Összevonáskor természetesen nem írjuk le külön az $\overline{A}B(C + \overline{C}) = \overline{A}B$ összefüggést, csak a kiemelés eredményét. Ezt pedig úgy állapíthatjuk meg, hogy (célszerűen ABC sorrendben) megkeresünk a hurok "közös jellemzőit", azokat a változókat, amelyek mindegyik mintermben egyforma értékkel fordulnak elő (jelen esetben mindkét 1-es az A = 0 sorban van, tehát az \overline{A} "közös jellemző", ezenkívül mindkét 1-esre érvényes, hogy B = 1, tehát B is "közös jellemző"). A "közös" változókat megfelelő értékkel szorzat formájában leírjuk, azt a változót pedig, amely egyik helyen 1-gyel, a másikon 0-val szerepel, elhagyjuk, azaz egyszerűsítünk ($C + \overline{C} = 1$). Ez az összevonási lehetőség a KARNAUGH-táblában minden szomszédos négyzetre fennáll, akár egymás mellettiek, akár egymás alattiak. Szomszédosnak számítanak és összevonhatók a sorok és oszlopok szélén lévő 1-esek is. Például "egymás melletti" a 0-ás és a 2-es minterm (2.54. ábra).

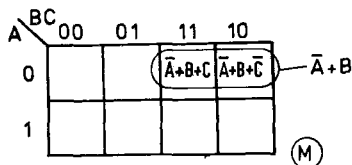


2.54. ábra

A maxtermek összevonásának elve és szabálya egyezik amin-
termekével. Például a 2-es és 3-as szomszédos maxterm össze-
vonható (2.55. ábra); hiszen:

$$(\bar{A}+B+C)(\bar{A}+B+\bar{C}) = \bar{A} + B + \cancel{C} = \bar{A} + B$$

vagyis a konjunktív alakban a szomszédos maxtermek összeszor-
zásakor csak a közös rész marad meg. A maxterm táblából is te-
hát a hurok "közös jellemzőit" kell kiirmunk - természetesen
maxterm (összeg) formában.

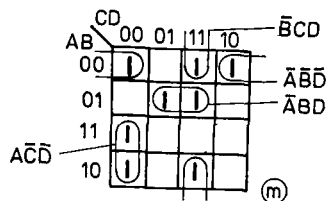


2.55. ábra

A kiemelés elve tovább vihető kettőnél több minterm vagy
maxterm összevonására is. A KARNAUGH-táblában - megfelelő alak-
zatban - 4, 8 általában

$$2^n$$

darab 1-es is összevonható.



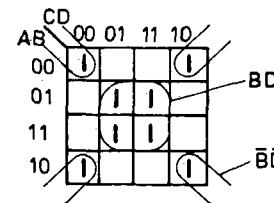
2.56. ábra

A lehetséges összevonások a minterm és maxterm táblában
4-nél nem több változóra a következők (a bizonyítást, ill. a
"próbát" az olvasóra bizzuk):

- 2 egymás melletti, alatti term összevonható (2.56. áb-
ra);
- 2 term összevonásakor 1 változó "kiesik".

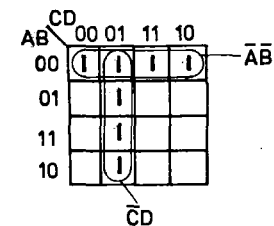
- 4 darab, négyzetet alkotó term összevonható (2.57. áb-
ra).

(A négy sarokban lévő 1-es is négyzet alakzatnak számít).
Ilyenkor 2 változó "esik ki".



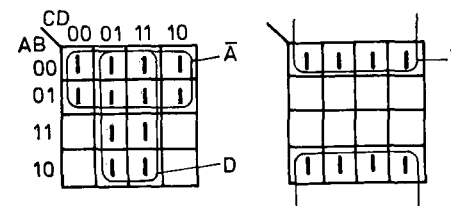
2.57. ábra

- Teljes sorok, valamint teljes oszlopok összevonhatók
(2.58. ábra);



2.58. ábra

- 2 (4, 8, ..., 2ⁿ) szomszédos teljes sor vagy oszlop ösz-
zevonható (2.59. ábra);



2.59. ábra

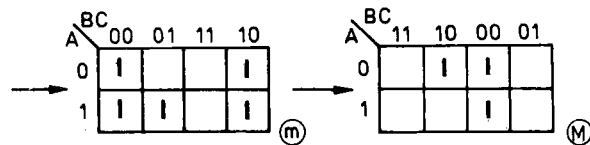
- Előfordulhat, hogy a tábla valamennyi rekeszében 1-es
van, ilyenkor az "egész tábla összevonható" az eredmény azo-
nosan egyenlő 1-gyel.

Függvények minimalása KARNAUGH-táblával

1. Első lépésben megrajzoljuk a minterm vagy maxterm táblát, aszerint, hogy az egyszerűsített függvényt diszjunktív (mintermes) vagy konjunktív (maxtermes) alakban akarjuk kapni. Sokszor mindkét táblát megrajzoljuk és az egyszerűbben realizálható eredményt választjuk.

Amennyiben a függvény igazságtáblázattal van megadva, a KARNAUGH-táblákat az ismert módon állítjuk elő (az igazságtábla 1-eseit írjuk be a minterm táblába, ebből kétszeres negálással rajzoljuk meg a maxterm táblát). BOOLE-kifejezéssel adott függvényeket először normál, ill. teljes normál alakúra kell hoznunk bővítéssel (lásd 2.3.3. pontot). Ezután már elkészíthetjük a KARNAUGH-táblát.

Vegyük példának a következő, igazságtáblázattal adott háromváltozós függvényt (amelyből közvetlenül megrajzolható a minterm tábla, abból pedig a maxterm tábla, 2.60. ábra):



2.60. ábra

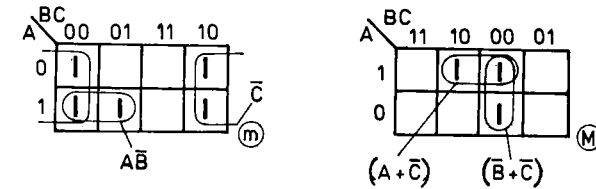
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

2. Megvizsgáljuk és bejelöljük az egyszerűsítési, összevonási lehetőségeket. A cél az, hogy:

- valamennyi 1-est egy-egy hurokkal lefedjük,

- az egyes hurkokat úgy jelöljük ki, hogy a lehető legtöbb 1-est "kerítsék körül", hiszen így esik ki a legtöbb változó.

A hurkok "közös jellemzőit" (a prim-implikánsokat) a hurkok mellé írjuk. A hurkokkal való lefedést célszerű olyan sorrendben végezni, hogy először azokat az "egyedülálló" 1-eket keressük meg, amelyek nem vonhatók össze másokkal, majd azokat az összevonható párosokat, amelyekből 4-es csoport nem készíthető, majd azokat a 4-eket, amelyekből 8-as csoport nem készíthető stb. (Amikor már mindegyik 1-est körülkerítettük a lehető legnagyobb hurokkal, meg kell állnunk, akármiilyen jó lehetőségeket látunk még, mert ezután már csak feleslegesen bővítjük a függvényt.)



2.61. ábra

Egy-egy minterm vagy maxterm több hurokban is benne lehet, többször is igénybe vehetjük egyszerűsítésre, ha ezáltal nagyobb hurkokat tudunk kijelölni. A többszörös igénybevétel nem jelent bővítést ($A + A + A + \dots = A$ és $A \cdot A \cdot A \dots = A$), hanem a nagyobb hurok miatt éppen, hogy egyszerűsítést. (Példánk Karnaugh-tábláit az egyszerűsítésekkel a 2.61. ábra mutatja.)

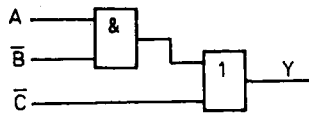
3. A hurok mellé írt prim-implikánsokból összeállítjuk a végeredményt, az egyszerűsített normál alakú függvényt.

$$Y = A\bar{B} + \bar{C} = (A + \bar{C})(\bar{B} + \bar{C}).$$

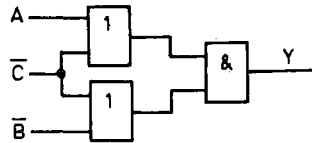
4. Végül megrajzoljuk a feladatot megvalósító logikai hálózatot - vagy NÉV rendszerben (az eredmény diszjunktív vagy konjunktív alakjából, aszerint, hogy melyik egyszerűbb), vagy univerzális (NAND, NOR, AOI) kapukkal. Az utóbbihoz még utólagos BOOLE átalakításokra lehet szükség, a közvetlenül KAR-

NAUGH-táblából történő tervezés módszereit egy következő fejezetben taglaljuk (a példa eredményül kapott hálózatot a 2.62. ábra mutatja).

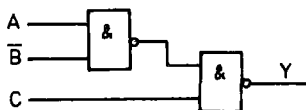
NÉV: $Y = A\bar{B} + \bar{C}$



$Y = (A + \bar{C})(\bar{B} + \bar{C})$

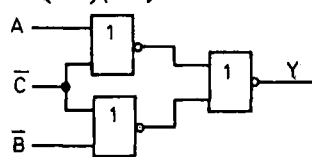


$Y = \overline{A\bar{B} + \bar{C}} = \overline{A\bar{B}} \cdot \overline{\bar{C}}$



(NAND)

$Y = \overline{(A + \bar{C})(\bar{B} + \bar{C})} = \overline{A + \bar{C}} + \overline{\bar{B} + \bar{C}}$



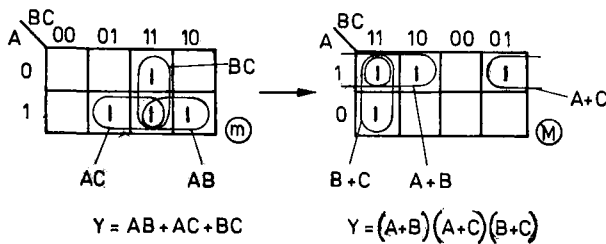
(NOR)

2.62. ábra

A szabályokat illusztráljuk néhány példán:

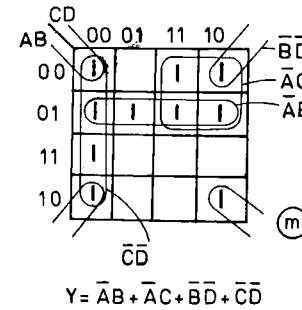
- állítsuk elő a háromváltozós "szavazó áramkör" egyszerűsített diszjunktív és konjunktív normál alakját KARNAUGH-táblák segítségével! (2.63. ábra);

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

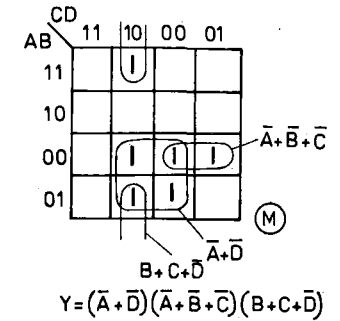
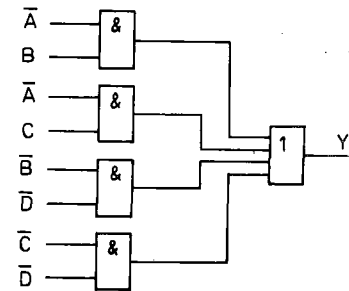


2.63. ábra

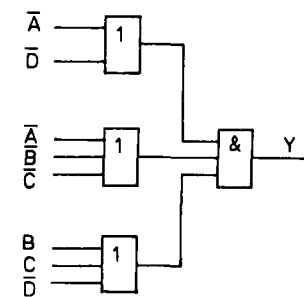
- minimáljuk az $Y = \sum_{i=0,2,3,4,5,6,7,8,10,12}^4$ függvényt (2.64. ábra);



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



$Y = (\bar{A} + \bar{D})(\bar{A} + \bar{B} + \bar{C})(B + C + \bar{D})$



2.64. ábra

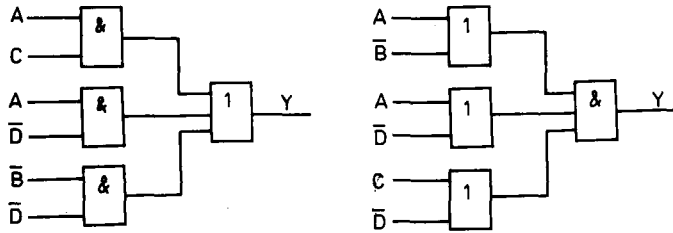
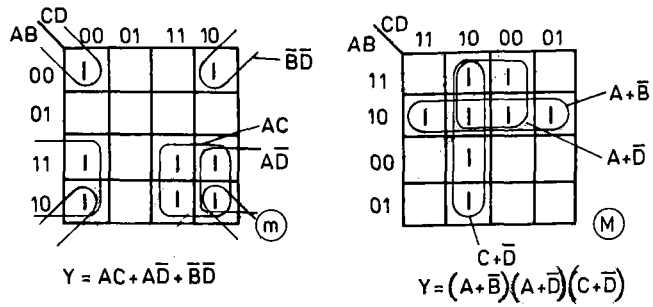
- Valósítsuk meg egyszerűsítve ÉS, VAGY kapukkal az

$$Y = \bar{B} \left[\bar{A} \bar{D} + (C \oplus D) \right] + A \left[\bar{D}(B \oplus C) + BC \right]$$

függvényt!

$$\begin{aligned} Y &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} = \\ &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} = \\ &= m_0^4 + m_2^4 + m_3^4 + m_8^4 + m_{10}^4 + m_{11}^4 + m_{12}^4 + m_{14}^4 + m_{15}^4 \end{aligned}$$

A Karnaugh-táblák és az eredményül kapott hálózatok (2.65. ábra).

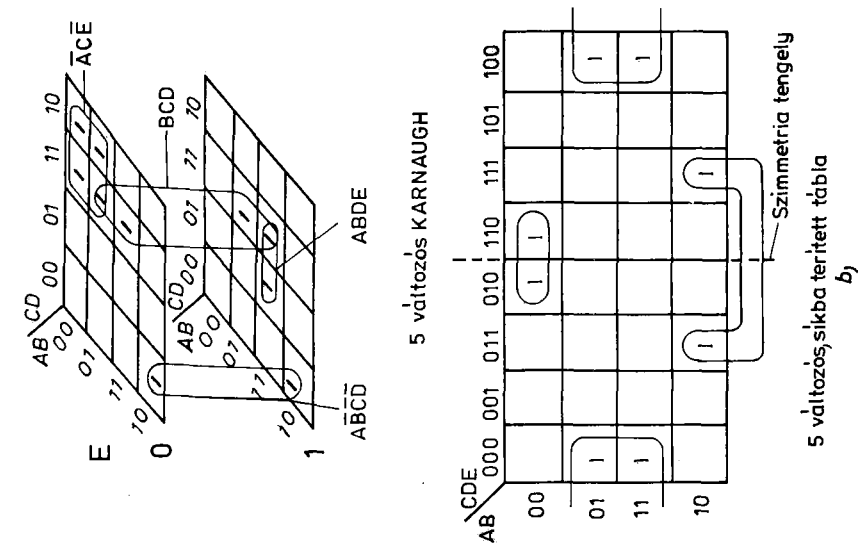
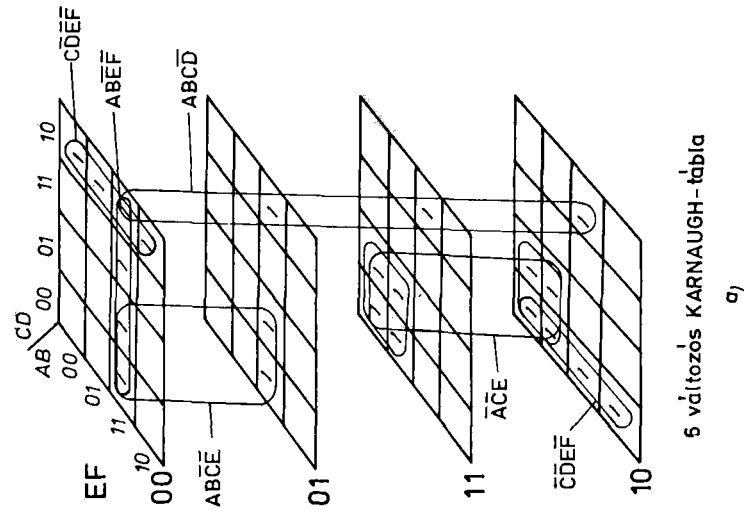


2.65. ábra

Több változós KARNAUGH-táblák

Az eddigiekben csak legfeljebb négyváltozós függvényekkel foglalkoztunk. Ha a táblákat 3 dimenzióban rajzoljuk meg, ill. képzeljük el, akkor - kissé nehézkesen ugyan, de öt-, és hatváltozós függvényeket is minimálhatunk. Az A, B, C, D változókra ugyanolyan táblát rajzolunk, mint ami szokásos, csak egymás "fölé" többet. Egy-egy sík tartozik az újabb "E" változó 0 és 1 értékéhez, ill. 4 sík az E, F változók értékkombinációihoz (2.66. ábra).

Az egyszerűsítés elve ilyenkor is változatlan, csak most az egyes táblákon végrehajtott összevonásokon kívül függőleges irányban is összevonhatjuk az egymás alatti szomszédos, azonos területrészeket (lásd a 2.66a ábrát), ez utóbbiakat is a már ismert szabályok szerint. A 2.66b ábra 5 változóra egy lehetséges síkbeli elrendezést mutat; ebben a táblában a szimmetrikusan elhelyezkedő rekeszek, alakzatok is szomszédosak, összevonhatók (2^n term!).

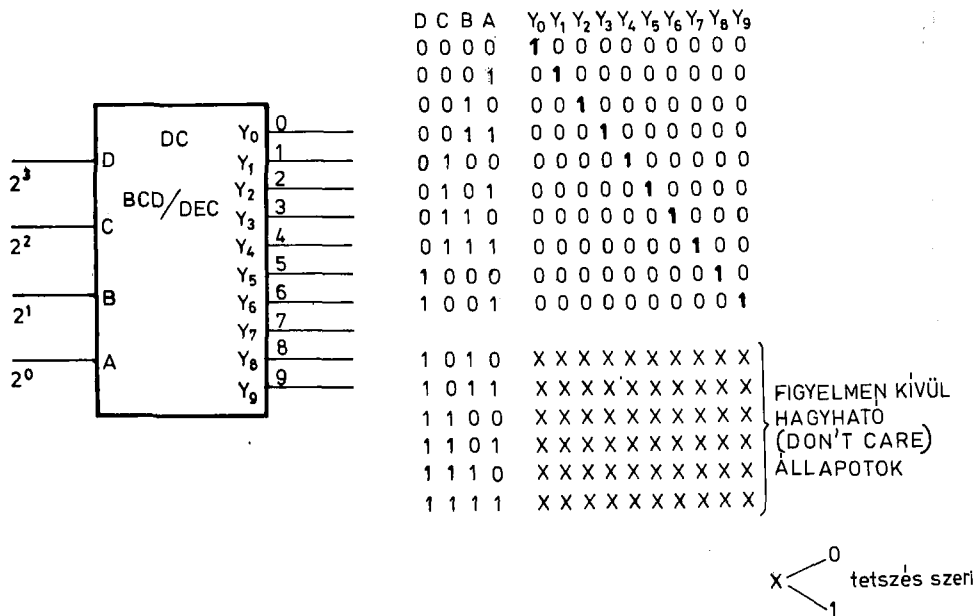


2.66. ábra

A figyelmen kívül hagyható (DON'T CARE) term-ek

A kombinációs és sorrendi hálózatoknál is gyakran előfordul, hogy vannak olyan bemeneti változó kombinációk, olyan term-ek, amelyek normális működés során nem fordulhatnak elő. Ezeket a "felesleges", elő nem forduló term-eket eredeti nevükön DON'T CARE (figyelmen kívül hagyható) term-eknek vagy REDUNDÁNS term-eknek nevezzük.

Készítsünk például egy BCD-decimális dekódót! Ennek az a feladata, hogy a 4 bites BCD bemeneti jel decimális megfelelőjét állítsa elő a 10 db kimeneten (2.67. ábra).

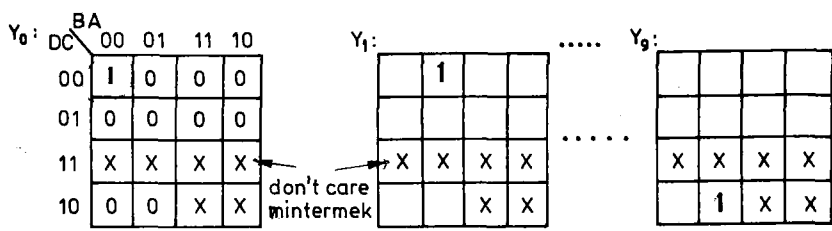


2.67. ábra

Ahányas BCD számjegy érkezik a bemenetre, annyiadik Y kimeneten kell logikai 1-nek megjelennie. A többi kimenet 0-ban kell, hogy legyen (például, ha a bemenetre 0101 érkezik, akkor az ötödik, Y₅ kimeneten kell logikai 1-nek lenni, a többin pedig logikai 0-nak). Az igazságtábla az ábrán látható. Nyilvánvaló, hogy 9-nél nagyobb számoknak jelen esetben nincs fizikai értelmük, hiszen a BCD kód 0-tól 9-ig terjed és ennek

felel meg a dekódoló 10 db kimenete is. Ezek szerint a 9-nél nagyobb számot jelentő A,B,C,D bemeneti jelkombinációk ennél az áramkörnél nem is fordulhatnak elő, ezekre az áramkör működését nem kell definiálnunk. Az igazságtáblázatban a kimeneti függvényekhez tartozó X-ek azt jelentik, hogy mivel ezek a sorok nem is fordulhatnak elő, ezért tetszés szerint 0-át vagy 1-et gondolhatunk a helyükre. A függvény utolsó 6 mintermje DON'T CARE vagy redundáns minterm, a hozzájuk tartozó fiktív függvényértékeket tetszés szerint felvehetjük, ahogy nekünk kellemebb. Ilyenkor ezt a lehetőséget célszerű arra használni, hogy az X-eknek olyan értéket tulajdonítunk, amivel a függvények a legegyszerűbbek lesznek, vagyis a don't care mintermeket (vagy maxtermekeket) egyszerűsítésre használhatjuk fel.

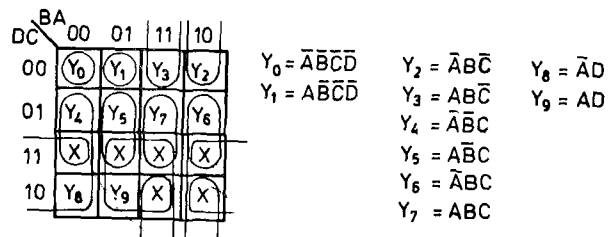
A BCD - decimális dekódoló tipikus több bemenetű, több kimenetű hálózatra is példa. Az eddig szokásos egy Y kimenet helyett itt több - jelen esetben tíz: Y₀Y₁Y₂...Y₉ - van. Végeredményben több kombinációs hálózatot egyesítettünk egyetlen blokkban. Mindegyik kimenetnek külön-külön igazságtáblázata és logikai függvénye van. Valamennyi Y függvényt egyenként fel kell írunk és minimálnunk. Érdekes közben arra is figyelni, hogy vannak-e a függvényeknek közös részei, közös implikánsai - ezeket csak egyszer kell megvalósítanunk, így gazdaságosabb lesz a megoldás (sokszor az egyes függvények egyszerűsítésekor nem is a legnagyobb hurkokat jelöljük ki azért, hogy minél több közös implikáns legyen).



2.68. ábra

Jelen esetben 10 db KARNAUGH-táblát kell rajzolnunk az egyszerűsítéshez. A függvényeknek közös része biztosan nincsen, hiszen mindegyik Y csak egyetlen esetben lesz 1-es és ezek nem esnek egybe (2.68. ábra).

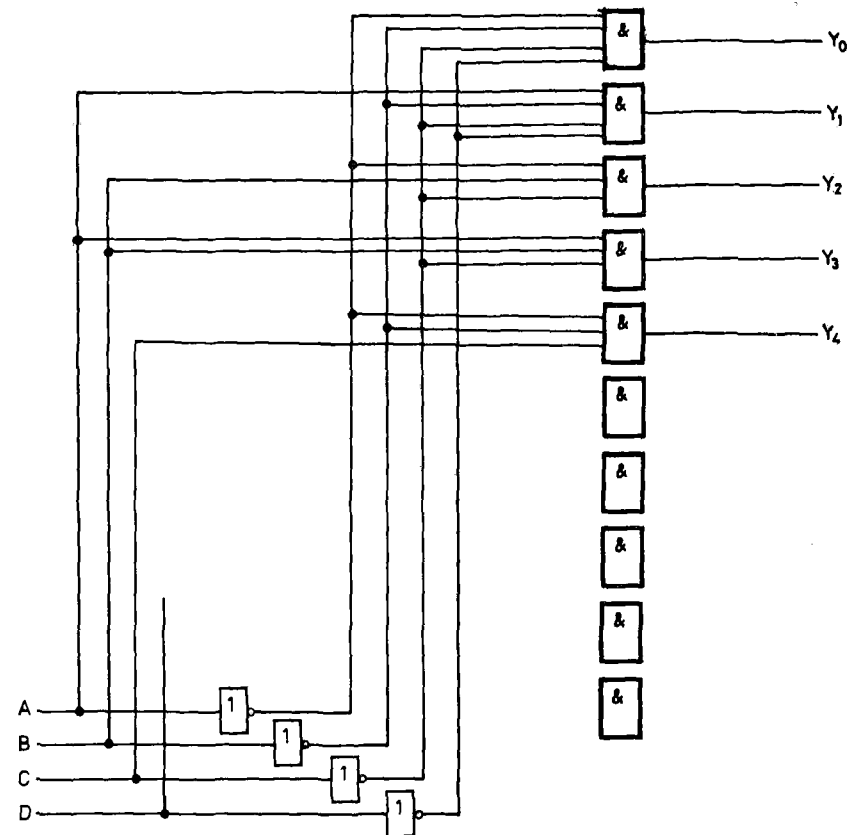
Tekintve, hogy mindegyik függvény csak egy 1-est tartalmaz és az mindig más rekeszben van és a don't care mintermek is ugyanott vannak, 10 minterm tábla helyett csak egyetlen "egyesített" táblát érdemes rajzolni; a megfelelő rekeszbe mindig azt az Y-t írjuk, amelyik ott 1-es értéket vesz fel. (2.69. ábra)



2.69. ábra

Az Y függvények kiírásakor meg kell vizsgálnunk, hogy nem lehet-e valamelyik X-szel jelölt term-mel összevonni; ilyenkor az illető don't care mintermet 1-esnek tekintjük, és az összevonás következtében egyszerűbb lesz a függvény. Az Y₀ és Y₁-et nem lehet semmivel összevonni, de az Y₂Y₃Y₄Y₅Y₆Y₇-et már összevonhatjuk egy-egy X-szel, így ezekből a függvényekből 1 változó kiesik. Az Y₈ és Y₉ négyes csoportban vonható össze, ezekből 2 változót tudunk ily módon kiküszöbölni. A don't care mintermek felhasználásával végül is 10 kapu bemenetet megtakarítottunk (egyszerűsítés nélkül mindegyik függvényt csak 4 bemenetű ÉS kapuval lehetett volna megvalósítani). Természetes, hogy az egyszerűsítés során "nem kötelező" don't care term-et igénybe venni, ha a hurok nem lesz nagyobb, a függvény nem lesz egyszerűbb, ilyenkor X-et 0-nak tekintjük.

A dekódoló logikai rajzát a 2.70. ábra mutatja (fejezzük be a rajzot, egészítsük ki Y₅-től Y₉-ig!)



2.70. ábra

Modern integrált áramkörök "belsejét" rendszerint nem egyszerűsítik a don't care mintermek felhasználásával, egyrészt azért, mert ott nincs jelentősége a kisebb kapu bemenet számnak, másrészt, mert a felhasználó szempontjából mégiscsak kedvezőbb, ha a tiltott kombinációkban is előre meghatározott módon működik az áramkör, például a kimenetek letiltódnak. A decimális számok kijelzésekor általában a 9-nél nagyobb számoknál az integrált dekódor "kioltja" a kijelzőt, ezt a készülékek tervezésekor jól fel lehet használni (pl. a felesleges jegyek, nullák kioltására).

A don't care mintermek felhasználásával való egyszerűsítés természetesen nemcsak a dekódolókra alkalmazható, hanem minden olyan kombinációs és sorrendi hálózatra, amelyben egyes

vezérlési kombinációk tiltottak, nem értelmezhetők (a don't care term-ekkel egyszerűsíthető sorrendi hálózatok tipikus példája az RS flip-flop - lásd ott).

2.4.4. Táblázatos egyszerűsítés:
a QUINE-McCLUSKEY-módszer

A táblázatos, "szisztematikus" (vagyis matematikailag az "egyenes úton" eredményt adó, intuiciót nem igénylő) megoldási módok közül a Quine-McCluskey-eljárás a legelterjedtebb. "Hétköznapi", egyszerű feladatokhoz nem nagyon használjuk, összetett, sok változós esetekben, számítógépes tervezéshez ajánlható. Az egyszerűsítés alapelve itt is a szomszédos min-termekek megkeresése, csak nem grafikusán, hanem táblázatos formában.

Mintermek - beláthatóan - akkor szomszédosak, akkor vonhatók össze, ha sorszámuk bináris kifejezésének HAMMING távolsága 1 - éppen ezek megtalálása érdekében készítünk táblázatokat. Az összevonhatóság szükséges feltétele, hogy az indexek különbsége 2 hatványa legyen (1, 2, 4, 8....), pl.:

$$m_2^3 + m_3^3 = \overline{A}B\overline{C} + \overline{A}BC = \overline{A}B \quad (3-2=1=2^0, \text{ az 1-es helyiértéken lévő, jelenleg } C \text{ változó egyszerűsödik ki, marad el}),$$

$$m_2^6 + m_3^6 = AB\overline{C} + \overline{A}B\overline{C} = \overline{C} \quad (6-2 = 4, \text{ a 4-es helyiértéken lévő } A \text{ változó marad el}).$$

Az egyszerűsítés, táblázatkészítés menetét egy egyszerű példán mutatjuk be.

A célszerű lépések a következők:

1. A függvény kiegészítése teljes (diszjunktív) normál alakra, annak érdekében, hogy a függvényben résztvevő mintermek indexei ismertek legyenek, pl. minimaljuk az

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

függvényt.

A kiegészítés "mechanikusan" úgy történhet, hogy a részletszorzatok mindegyikéhez bináris számokat rendelünk - ugyan-
gy, ahogy a mintermek számának megállapításakor tettük: az ábécé sorrendben leírt szorzatban a ponált változónak 1-et, a negáltkak 0-át feleltetünk meg. A szorzatban jelen nem lévő változó helyére "egyszerre" 0-át és 1-et írunk: \emptyset jelöléssel, ami azt jelenti, hogy az ilyen jelet tartalmazó bináris számot ezen a helyiértéken 0-val és 1-gyel is fel kell írunk, tehát:

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

010 \emptyset	01 \emptyset 1	100 \emptyset	10 \emptyset 1	011 \emptyset					
0100	0101	0101	0111	1000	1001	1001	1011	0110	0111
dec: 4	5	5	7	8	9	9	11	6	7

Igy a függvény (a többször előforduló indexeket természetesen csak egyszer szerepeltetve, hiszen $A + A + \dots = A$).

$$Y = \sum_{i=4}^9 (4, 5, 6, 7, 8, 9, 11) \quad \text{normál alakú.}$$

2. A mintermek csoportosítása aszerint, hogy indexük bináris értékében hány 1-es van (hány ponált változó van, ezt nevezhetjük index-sulyoknak):

0	-	
1	4	0100
	8	1000
2	5	0101
	6	0110
	9	1001
3	7	0111
	11	1011

3. A lehetséges összevonások végrehajtása (ez több lépésben történik): Először felülről haladva összehasonlítjuk egy-egy adott index-sulyu csoport elemeit az alatta lévő cso-

port elemeivel minden kombinációra. Ha találunk olyan számpárokat, amelyek különbsége 2 valamelyik hatványával egyenlő (és a lejjebb lévő szám nagyobb, mint a felette lévő), akkor ezeket egy következő rovatban összegyűjtjük. Minden tagot össze kell hasonlítani legalább egyszer, ennek megtörténtét célszerű emlékeztetőül megjelölni, "kipipálni". A kipipált taggal már nem szükséges foglalkozni, hacsak nem használható fel másutt is. Az egyszerűsítésbe be nem vonható számokat is meg kell jelölnünk (valamilyen megkülönböztető jellel, pl. csillaggal).

0	-	-	← (0-ás és 1-es csoport összehasonlítása),
1	4 ✓ 0100 8 ✓ 1000	4,5 (1)	← 1-es és 2-es csoport összehasonlítása,
2	5 ✓ 0101 6 ✓ 0110 9 ✓ 1001	8,9 (1) 6,7 (1)	
3	7 ✓ 0111 11 ✓ 1011	9,11 (2)	← 2-es és 3-as csoport összehasonlítása.

Jelen esetben minden tagot felhasználhattunk összevonásra. A második lépcsőben az így kialakult csoportokat hasonlítjuk össze felülről lefelé. Csak azok a szám-kettősök vonhatók össze, amelyek mellett a zárójelben ugyanakkora szám van (vagyis az előző összevonásnál azonos különbség volt köztük), így biztosíthatjuk csak, hogy az összevont szám-kettősök első tagja, ill. második tagjai között azonos különbség legyen. Példánkban a 4,5 szám-kettős vonható össze a 6,7 kettőssel, a 8,9 és a 9,11 kettős nem (8-9 között 1 lenne a különbség, de 9-11 között 2!), ezek a függvény tovább nem egyszerűsíthető alkotói (prim-implikánsai) lesznek:

0	-	-	-
1	4 0100 8 1000	4,5 (1) ✓	-
2	5 0101 6 0110 9 1001	8,9 (1) ✗ 6,7 (1) ✓	4,5,6,7 (1) ✗
3	7 0111 11 1011	9,11 (2) ✗	

4. Az így "megmaradó" tagok (implikánsok) segítségével fel kell írni az egyszerűsített függvényt:

$$\begin{aligned} \text{✗ } 4,5,6,7 : & \left. \begin{array}{l} 0100 \\ 0101 \\ 0110 \\ 0111 \end{array} \right\} 01\phi\phi \rightarrow \bar{A}B \quad (C \text{ és } D \text{ kiesik}) \\ \text{✗ } 8,9 : & \left. \begin{array}{l} 1000 \\ 1001 \end{array} \right\} 100\phi \rightarrow A\bar{B}\bar{C} \quad (D \text{ kiesik}) \\ \text{✗ } 9,11 : & \left. \begin{array}{l} 1001 \\ 1011 \end{array} \right\} 10\phi 1 \rightarrow A\bar{B}D \quad (C \text{ kiesik}) \end{aligned}$$

$$Y = \bar{A}B + A\bar{B}\bar{C} + A\bar{B}D \quad \text{az eredmény.}$$

Több változó esetén az eljárás hasonló, csak természetesen a táblázatnak még több lépcsőből kell állnia, az egyszerűsítést a megismert elven folytatni kell.

2.5. Kombinációs hálózatok megvalósítása univerzális műveleti elemekkel

2.5.1. NAND hálózat tervezése KARNAUGH-táblával

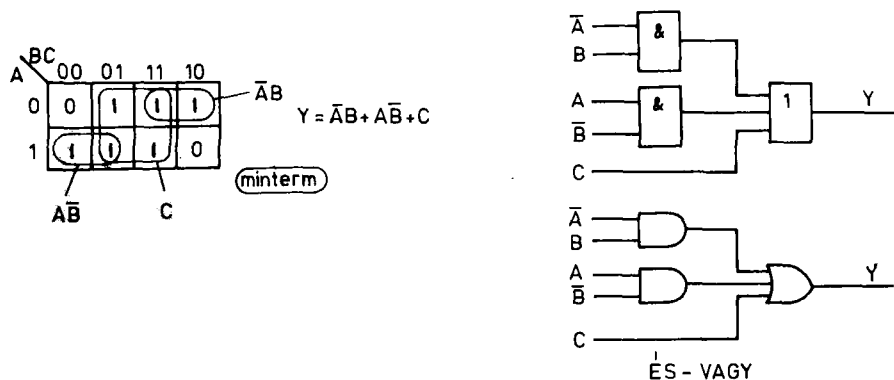
Célunk az, hogy a KARNAUGH-táblából közvetlenül meg tudjuk rajzolni a kizárólag NAND kapukból álló hálózatot. Feltételezzük, hogy a változó-negáltak is rendelkezésre állnak vagy

már a rendszerben jelen vannak, vagy már előzőleg inverterekkel előállítottuk őket. Mindig kanonikus függvényalakot valósítunk meg.

A módszert egy példa segítségével a legkönnyebb megérteni. Legyen a megvalósítandó háromváltozós függvényünk:

$$Y = \sum^3 (1, 2, 3, 4, 5, 7).$$

Ennek minterm táblája és AND-OR (ÉS-VAGY) megvalósítása (2.71. ábra):



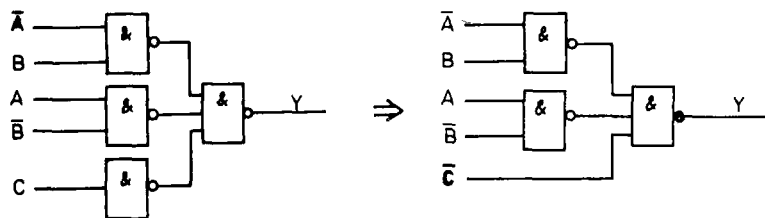
2.71. ábra

Negáljuk kétszer a kapott függvényt:

$$Y = \overline{\overline{AB} + \overline{A\bar{B}} + C} = \overline{\overline{AB} \cdot \overline{A\bar{B}} \cdot \overline{C}}$$

NAND NAND INV.

Az eredmény már csak NAND kapcsolatot tartalmaz, megvalósítása a 2.72. ábrán látható:



2.72. ábra

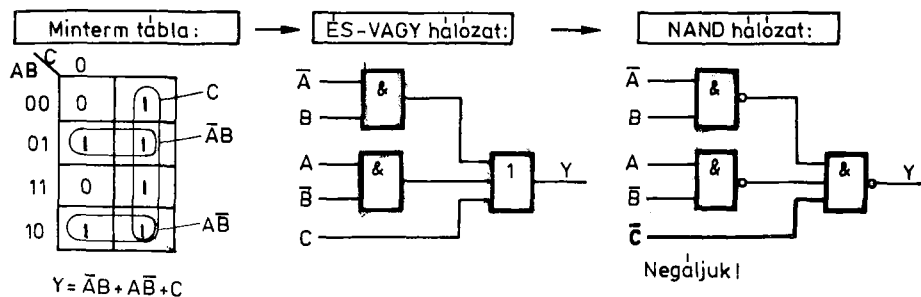
Összehasonlítva a 2.71. ábrán kapott ÉS-VAGY hálózatot és a 2.72. NAND változatot, észrevehetjük, hogy szerkezet szempontjából teljesen megegyeznek, csak éppen az ÉS-VAGY kapukból NAND lett. A bemeneten a változók is ugyanazok, kivéve azt a változót, amelyik közvetlenül a kimeneti kapuhoz megy - ezt negálni kellett a NAND hálózat felrajzolásakor. Minthogy a De-Morgan szabály minden diszjunktív normál alakú függvényre a példához hasonlóan alkalmazható, kimondhatjuk a NAND hálózat tervezésének szabályát:

- felrajzoljuk a függvény minterm tábláját,
- a minterm tábla 1-eseit összevonjuk,
- az összevonások eredménye alapján megrajzoljuk az ÉS-VAGY hálózatot.

Ezután következik az átalakítás NAND-re:

- az ÉS-VAGY hálózat valamennyi kapuját átjelöljük NAND-re, miközben az elrendezést nem változtatjuk,
- azokat a változókat, amelyek közvetlenül a kimeneti (legutolsó) kapuhoz mennek, negáljuk.

A teendőket a 2.73. ábrán foglaltuk össze:

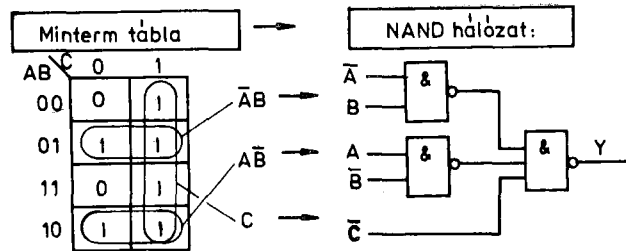


2.73. ábra

Természetesen az ÉS-VAGY (AND-OR) hálózatot nem feltétlenül szükséges megrajzolni, megfelelő gyakorlattal közvetlenül a minterm táblából is előállíthatjuk a NAND hálózatot:

- az összevont 1-es hurkokat egy-egy NAND-kapuvál megvalósítjuk,
- majd ezeknek a NAND kapuknak a kimeneteit egy "gyűjtő" közös NAND kapura vezetjük, ennek kimenetén áll elő a függvény.

- Ha a minterm táblában olyan 1-es csoportot vonunk össze, amelynek eredménye egyetlen változó, akkor ezt negálva közvetlenül a kimeneti, "gyűjtő" kapura vezetjük (2.74. ábra):



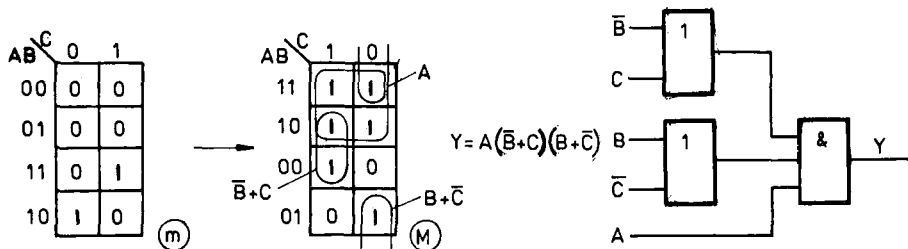
2.74. ábra

2.5.2. NOR (NOT-OR vagy OR-INVERT) hálózat tervezése KARNAUGH-táblával

A NOR művelet a NAND duálja, így a tervezés lépései is hasonlóak. Különbség csak az, hogy most a MAXTERM táblát használjuk fel, mivel a konjunktív normál alakból kétszeres negálással NEM-VAGY, NOR alakot kaphatunk. Például valósítsuk meg a következő függvényt:

$$Y = \sum^3 (4,7)$$

Felrajzoljuk a minterm táblát és ebből a maxterm táblát (2.75. ábra):

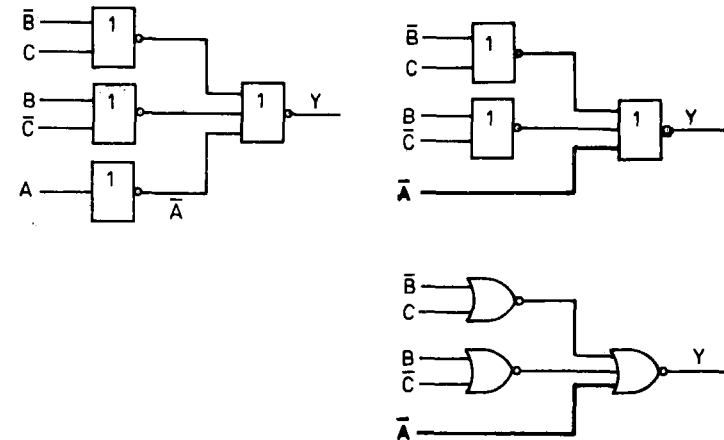


2.75. ábra

Negáljuk kétszer a maxterm táblából kapott egyszerűsített konjunktív alakot:

$$Y = A(\overline{\overline{B+C}})(\overline{\overline{B+C}}) = \overline{\overline{A+(B+C)}} + (\overline{B+C})$$

Ez már csak NOR kapcsolatokat tartalmaz (2.76. ábra).

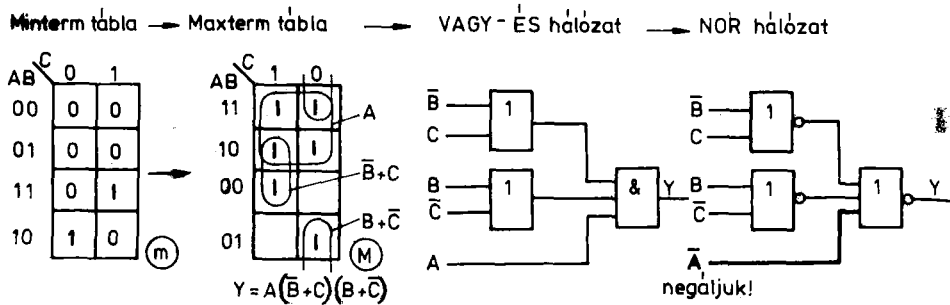


2.76. ábra

Általában a megoldás menete tehát:

- felrajzoljuk a függvény maxterm tábláját (legtöbbször a minterm táblából),
- a maxterm tábla 1-eseit összevonjuk,
- az összevonás alapján megrajzoljuk a VAGY-ÉS hálózatot. Ezután következik az átalakítás NOR-ra:
- a VAGY-ÉS hálózat valamennyi kapuját átjelöljük NOR-ra,
- azokat a változókat, amelyek közvetlenül a kimeneti kapuhoz mennek, negáljuk.

A teendőket a 2.77. ábra foglalja össze:

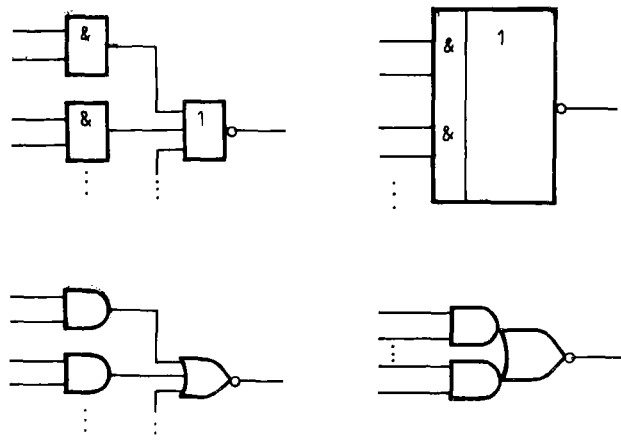


2.77. ábra

Természetesen itt sem szükséges a VAGY-ÉS hálózatot megrajzolni. A maxterm táblából az 1-es hurkokat egy-egy NOR kapuval megvalósítjuk és azok kimeneteit egy közös, "gyűjtő" NOR kapura vezetjük. Amennyiben olyan hurkot jelöltünk ki, amelynek eredménye egyetlen változó, azt negálva kell a "gyűjtő" kapura vezetnünk.

2.5.3. AND-OR-INVERT (ÉS-NEM-VAGY) hálózat tervezése

Az AND-OR-INVERT (AOI) kapu úgy fogható fel, mint egy "kibővített" NOR kapu. A NOR kapu bemeneteit egy-egy ÉS kapuval bővítettük ki (2.78. ábra):



2.78. ábra

Ez a TTL AOI kapukra áramköri szempontból is igaz, az AOI belső felépítése szinte teljesen megegyezik a NOR kapuéval. Ebből az is következik, hogy egy AOI kapu késleltetési ideje (amíg a vezérlőjel a bemenetről a kimenetre jut) kb. ugyanannyi, mint egy NOR kapué és ez nagy előny, hiszen az AOI két fokozatu kapu áramkör, de jelkésleltetése csak egyszeres! Ennek köszönhetően AOI rendszerben gyorsabb működésű áramköröket tudunk felépíteni. Az MSI integrált áramkörök belsejét is rendszerint AOI kapukból építik fel, mert így lesznek a lehető leggyorsabbak és áramköri szempontból is a legegyszerűbbek, legkisebb helyfoglalásuak.

Az AOI áramkörökkel való tervezés esetén nehéz a minimális hálózat kritériumát kimondani. A NAND és NOR logikánál a minimális kapuszám és a kapu bemenet szám a cél, a típusválaszték figyelembevételével. Viszont, ha AOI kapukkal építünk hálózatot, akkor még nagyobb körültekintéssel figyelembe kell venni a típusválasztékot, azt, hogy egy-egy tokmit (hány ÉS kaput, hány bemenettel) tartalmaz. Egyes AOI kapukhoz kívülről további bővítő ÉS kapuk csatlakoztathatók. Olyan ésszerű hálózatot kell terveznünk, amely minimális tokot igényel. Nagyon fontos tudnivaló, hogy egy AOI kaput egyetlen blokk gyanánt kell kezelni, az ÉS kapuk kimenetei nincsenek kivezetve, nem lehet őket elvezetni, jelüket felhasználni.

Az AOI kapukon kívül a bipoláris áramhuzó rendszerekben (pl. TTL) gyakran használt nyitott kollektoros NAND kapuk kimeneteinek párhuzamos összekötésével létrehozott ún. huzalozott-VAGY áramkörök is AOI kapcsolatot valósítanak meg (lásd ott).

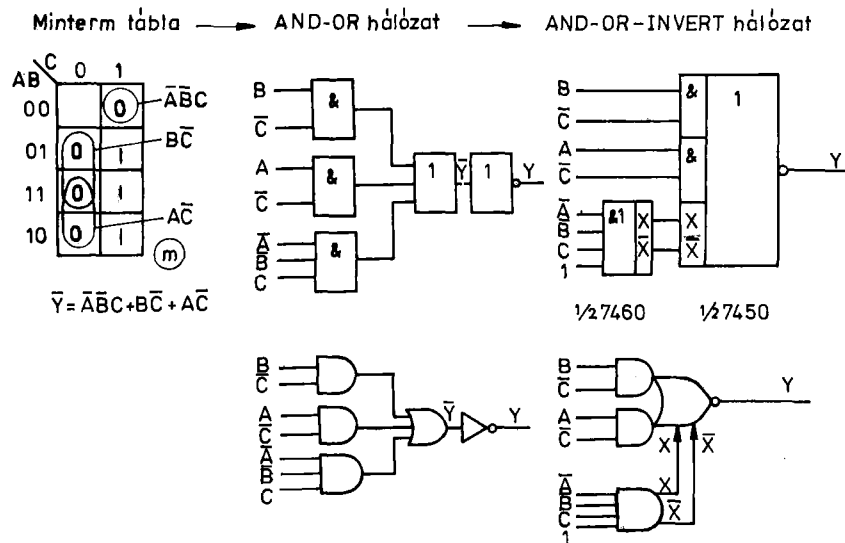
Az AOI hálózat "majdnem azonos" az ÉS-VAGY hálózattal, csak éppen a kimenet invertálva van. Ha a minterm tábla 1-eseit valósítanánk meg AOI hálózattal, akkor a függvény negáltját kapnánk (amit csak egy inverterrel lehetne "vissza-negálni"). Logikus tehát, hogy tervezéskor először ÉS-VAGY hálózattal a függvény negáltját kell előállítanunk. Ezután a függvény negáltját szolgáltató ÉS-VAGY hálózat kimenetét invertáljuk, így előállítjuk a kívánt függvényt, még hozzá AND-OR-INVERT hálózattal.

A tervezés menete tehát a következő:

- felrajzoljuk a KARNAUGH minterm táblát.
- A táblában a 0-kat vonjuk össze implikáns hurkokba (hiszen a függvény negáltját realizáljuk első lépésben).
- Megrajzoljuk az összevonásoknak megfelelő ÉS-VAGY hálózatot.
- A kimenetre rajzoljuk az invertálást jelentő null-kört, így a függvény negáltja helyett magát a függvényt kapjuk AOI rendszerben.
- A típusválaszték alapján megkeressük a legegyszerűbb megvalósítási formát.

Például állítsuk elő AOI hálózattal a következő függvényt (2.79. ábra):

$$Y = \sum_{(0,3,5,7)}^3$$



2.79. ábra

A feladat megoldásához pl. TTL-ben egy "fél" 7450-es tok kell, és ehhez egy "fél" 7460-as bővítő ÉS kapu, vagyis viszonylag gazdaságos a felhasználás, a NAND megvalósítás gazdaságtalabb lenne. (Gondoljunk arra, hogy az AOI kapukon az ÉS-NOR

összekötések token belül, "készen vannak", míg a különálló, pl. NAND kapukból történő hálózat építéskor a két "kapu-szint" közötti összekötéseket is meg kell valósítanunk, ami a huzalozást bonyolultabbá teszi - ezért is javasolható az AOI kapu felhasználása minden olyan helyen, ahol a típusválaszték ezt lehetővé teszi.)

3. A LOGIKAI HÁLÓZATOK ÉPÍTŐELEMEI, LOGIKAI ÁRAMKÖR CSALÁDOK

Az eddigiekben vázlatosan áttekintettük digitális áramkörök működésének leírásai lehetőségeit, alapvető tervezési elveit. Ebben a fejezetben azokkal az eszközökkel (hardverrel) foglalkozunk, amelyek a kitűzött feladatokat ténylegesen megvalósítják. Ismertetésünk nem lehet teljes; csak a villamos eszközökkel foglalkozunk (pl. pneumatikus, hidraulikus logikákkal nem) és ezek közül is a leggyakrabban, a mindennapi életben használatos fajtákat említjük meg. A "különleges", speciális elemeket katalógusból és a szakirodalomból kell megismerünk (ez vonatkozik a szinte napról-napra megjelenő legújabb eszközökre, alkatrészekre is).

3.1. Elektromechanikus logikák

A kapcsolókból, jelfogókból felépülő logikai áramkörök tekinthetők a "legősibbnek", ezekre alkalmazták először a kapcsolási algebrát. "Elektromechanikus logika" alatt általában jelfogókból felépülő áramkört értünk.

Az érintkezős logikai hálózatoknak ma is van létjogosultsága, vannak alkalmazási területei, nem jelenthetjük ki, hogy korszerűtlen lévén, új berendezésbe jelfogót nem építünk be. Kétségtelen, hogy lassúsága ($n \cdot 10$ ms meghúzási idő), korlátozott élettartama ($n \cdot 10^4 \dots 10^5$ számú meghúzás), nagy helyfoglalása, nagy meghúzási áram fogyasztása ($n \cdot 1 \text{mA} \dots 1 \text{A}$) folytán a legtöbb helyen nem veszi fel a versenyt a mai félvezető építőelemekkel, ezért kimondottan digitális gyengeáramu berendezésekbe ma már ritkán építenek jelfogós logikát. Az is

biztos, hogy például egy hálózati motorindító mágneskapcsoló tartó áramkörét (mely egy jelfogós sorrendi logikai hálózatnak tekinthető) senkinek sem jutna eszébe valamiféle integrált áramkörös kapcsolással helyettesíteni, hiszen a tartás egyetlen többlet munkaérintkezővel megvalósítható és az integrált áramkörhöz még külső tápellátás is kellene. (Az erősáramu jelfogónak is létezik ma már az elektronikus megfelelője: az "SR": Solid-state Relay, szilárdtest jelfogó, amely kisfeszültséggel vezérelhető, rendszerint fény-csatolású és több tíz amperes, többszáz voltos hálózati áramkörbe iktatható, mozgó alkatrészeket nem tartalmaz, robbanásbiztos.)

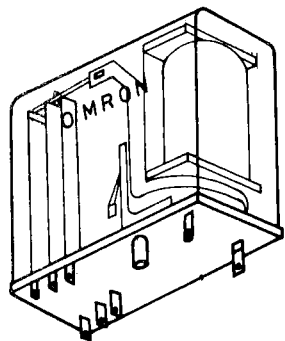
A jelfogók (és különböző módosulataik) ma még például a következő területeken nélkülözhetetlenek:

- erősáramu vezérlésekben (nagy áramok, viszonylag kevés logikai művelet, kis sebességigény);
- távbeszélő technikában (a tiszta elektronikus, mozgó alkatrész nélküli berendezések elterjedéséig);
- elektronikus berendezésekben nagyobb teljesítményű egységek meghajtására (izzó, motor, perifériális egységek, stb.);
- elektronikus berendezésekben, ahol egyes jelvezető "melegpontokat" kell összekötni földtől és a vezérlő körtől függetlenül, galvanikusan jól elszigetelve, miközben esetleg ezek a pontok nagy feszültségen is vannak. Ilyenkor az elektronikus kapcsolók (bipoláris tranzisztor, JFET, MOSFET) felhasználása ugyyszólván lehetetlen. Az analóg jelet kapcsoló jelfogók szorosán véve nem is tekinthetők logikai eszközöknek, de a kapcsoló eszközök családjába tartoznak.

A jelfogós áramkörök tervezése "külön tudomány", amelynek elsajátítását nem tekintjük célunknak, csak a legfontosabb alapismereteket tárgyaljuk a kiviteli formákról és alapvető felhasználásokról.

Ami a "gyengeáramu" jelfogókat illeti, ma már sokféle célra, sokféle kivített gyártanak, pl. a több amperes, több tíz vagy több száz voltos, több érintkezős univerzális elektromechanikus típusokat (3.1. ábra), a kevesebb érintkezős (rendszerint két MORZÉ-s) miniatűr reléket, amelyek közül a legkisebbeket dual-in-line tokban vagy éppen TO-5-ös tranzisztor tokban helyezik el (3.2. ábra). Ott, ahol hosszú élettartamra,

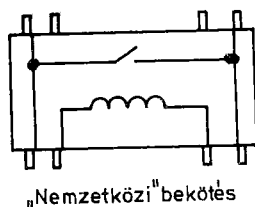
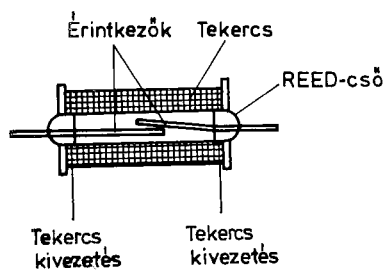
nagy üzembiztonságra, valamint az érintkezők között nagyon kis átmeneti ellenállásra van szükség kis áramerősség mellett, a REED-relé felhasználása a legcélszerűbb. Az üzembiztos és meg- lehetősen gyors működést a külvilágtól elzárt érintkezőknek kö- szönheti és annak, hogy minimális mozgó alkatrészt tartalmaz (3.3. ábra).



3.1. ábra



3.2. ábra



3.3. ábra

Sokszor előfordul, hogy egy érintkezős hálózat kimeneti jelét elektronikus logikai áramkörbe vezetjük. Ilyenkor nagy problémát okoz az érintkezők pergése. Amikor a jelfogó meghuz vagy elenged, az érintkezők nem zárnak, ill. bontanak azonnal, egy ideig bizonytalanul hol összeérnek, hol megszakadnak. A pergés eredménye, hogy az egyszeres 0-1 vagy 1-0 átmenet helyett sok határozatlan impulzus keletkezik, ami az őt követő elektronikus logikai hálózatot - különösen, ha az sorrendi há-

lózat - megzavarja (3.4. ábra). A pergést ezidő szerint csak a higany-nedvesítésű REED-relé küszöböli ki. Ezek ugyan lassabban működnek, de nagyobb áramot bírnak el és pergésmentesek. Minden más esetben elektronikus úton kell a pergésmentesítésről gondoskodnunk!



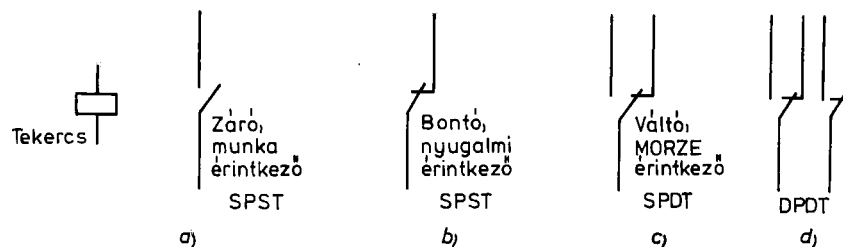
3.4. ábra

A jelfogók (és mechanikus kapcsolók) érintkező alaptípusaival már foglalkoztunk:

- a záróérintkező vagy munkaérintkező meghuzáskor, ill. működtetéskor hoz létre villamos kapcsolatot (szabványos rajz- jele a 3.5a ábrán).

Szakirodalomban, katalógusokban található elnevezése: a SPST (Single Pole Single Throw);

- a bontóérintkező vagy nyugalmi érintkező működtetéskor megszakít (3.5b ábra, SPST);



3.5. ábra

- a váltó vagy MORZE érintkező, a két alaptípus szokásos egyesített kombinációja: működtetéskor a mozgó középérintkező a nyugalmi érintkezőt elengedi, a munkaérintkezőt zárja (3.5c ábra).

Szokásos nemzetközi megjelölése: SPDT (Single Pole Double Throw).

Az is fontos, hogy a bontás és zárás milyen sorrendben történik, ebből a szempontból az SPDT kapcsoló kétféle lehet:

- BBM (Break Before Make), azaz először megszakít, azután zár, tehát egy egészen rövid ideig a középérintkező seholva sem kapcsolódik.

- MBB (Make Before Brake) vagyis először összeköt, azután bont, tehát egy egészen rövid ideig a középérintkező mindkét szélsőhöz hozzáér (egy pillanatra mindhárom pont összeér).

Jelfogók MORZE érintkezője általában BBM típusu, a mechanikus (és elektronikus) kapcsolók BBM és MBB típusúak is lehetnek, az adott feladathoz megfelelő nekünk kell kiválasztanunk (például egy méréshatárváltó kapcsoló jobb ha MBB típusu, így a kapcsolóra csatlakozó erősítő bemenete egy pillanatra sem marad szabadon, de pl. egy digitális berendezéshez csatlakoztatott kapcsoló, amely logikai 0, vagy 1 szintet ad, jobb ha BBM működésű, különben lenne egy olyan pillanat, amikor mindhárom pontot összezárva rövidrezárná a 0, ill. 1 szintet adó forrást!);

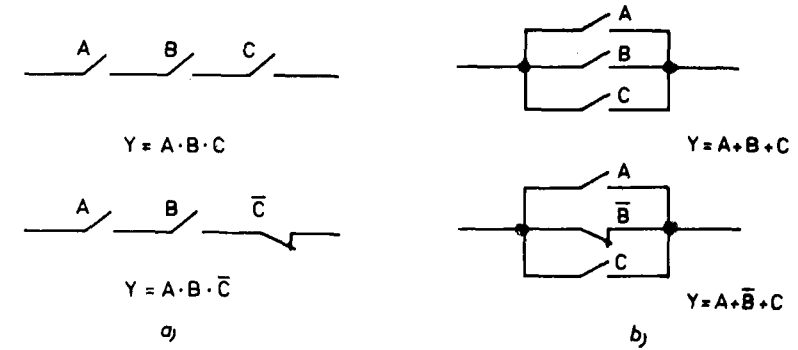
- a váltó érintkezőből egy-egy jelfogóra vagy kapcsolóra sokszor kettőt helyeznek el (DPDT: Double Pole Double Throw, d. ábra), esetleg többször kettőt (n-DPDT).

Léteznek összetett jelfogó érintkező (kapcsoló) összeállítások, amelyek adott sorrend szerint végeznek megszakítást, összekapcsolást. Egyes feladatokhoz középállású vagy polarizált (meghúzó áramiránytól függő irányban elmozduló középérintkezőjű) jelfogót alkalmazunk - ezekkel itt nem foglalkozunk részletesen.

Röviden foglaljuk össze az alap-függvények jelfogós realizálását!

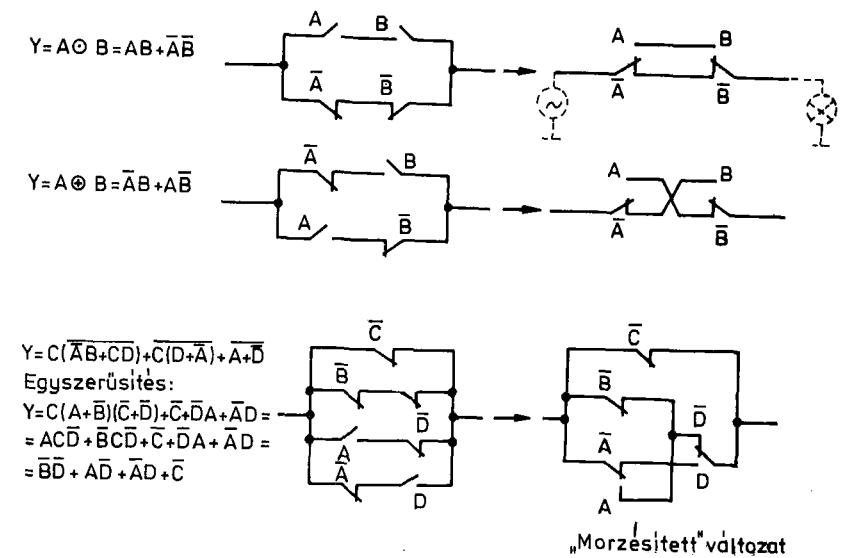
Egy-egy logikai változó azt az eseményt jelenti, hogy a hozzá tartozó jelfogó meghúz. A rajta lévő munkaérintkezők zárnak, a nyugalmi érintkezők bontanak. Egy A-val jelölt jelfogó munkaérintkezőit - mivel A bekövetkezésekor zárják az áramkört - A-val nyugalmi érintkezőit - mivel A bekövetkezésekor megszakítják az áramkört - \bar{A} -tal jelölhetjük.

Több változó ÉS kapcsolását - mint tudjuk - az A, B, C,stb. jelfogók munkaérintkezőinek soros kapcsolásával realizálhatjuk, hiszen az érintkezők áramköre akkor záródhat, ha A ÉS B ÉS C... bekapcsolt állapotban van (3.6a ábra).



3.6. ábra

Több változós VAGY kapcsolat a munkaérintkezők párhuzamos kapcsolásával hozható létre, mert így, ha bármely jelfogó meghúz (bármely változó 1-es értéket vesz fel), az áramkör már záródik (3.6b ábra). Ha az ÉS kapcsolatban vagy a VAGY kapcsolatban valamely változó negált értéke szerepel, akkor természetesen neki egy nyugalmi érintkezőt rajzolunk.



3.7. ábra

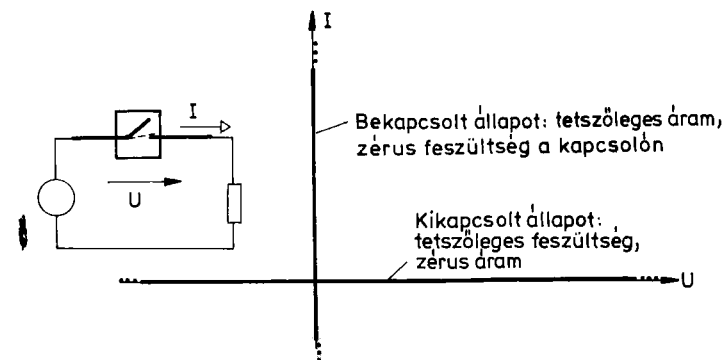
Amennyiben nem tisztán ÉS, ill. VAGY kapcsolat megvalósításáról van szó, hanem bármely logikai függvényről, akkor az egyszerűsített függvényt Nem-És-Vagy, NÉV rendszerben felírva egyszerűen megrajzolható a vegyes kapcsolású érintkező hálózat (3.7. ábra).

Sokszor lehetőség van a "morzésítésre", azaz ugyanazon jelfogó munka- és nyugalmi érintkezőjének morze-érintkezővel való helyettesítésére, ha az illető munka- és nyugalmi érintkezők egyik pontja össze van kötve vagy összeköthető, ahogy az a 3.7. ábra ekvivalencia, antivalencia áramkörében és például a hálózatban látható. (Érdemes megjegyezni, hogy az antivalencia/ekvivalencia kapcsolás két kapcsolóval, áramforrással, izzólámpával felépítve nem más, mint az ún. "alternatív", "lépcsőházi" kapcsolás: két különböző helyen tetszés szerint ki-be kapcsolhatjuk a világitást, nem szükséges kiegészítő jelfogók, stb. felhasználása, csak két-érintkezős kapcsoló és huzalozás kell.)

3.2. Diódás logikai áramkörök, diódák kapcsoló üzeme

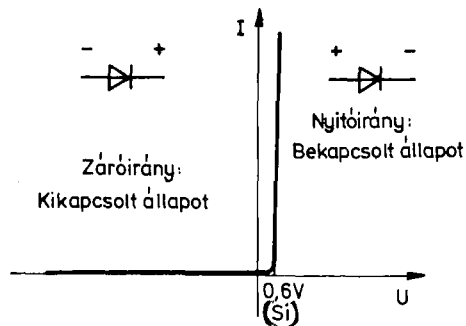
Ettől a fejezettől kezdve a félvezetős kapcsoló áramkörökkel foglalkozunk. A félvezető eszközök kapcsoló üzemi működése sok szempontból eltér az eddig tanult és "megszokott" lineáris működéstől. A lineáris üzemben az adott munkapontba beállított eszközön a vezérlőjel okozta változások kismértékűek, lineáris helyettesítőképet rajzolhattunk. Kapcsolóüzemben két "szélső", határozott állapot előállítása a cél: egy jelet átengedjünk vagy ne engedjünk át, ill. egy adott pont feszültségét (áramát) a bemeneti jeltől (jelektől) függően két szélsőséges IGEN (1) vagy NEM (0) szintre állítsuk be. Ennek érdekében a félvezető eszközöket általában a lineáris üzemmóddhoz képest sokkal nagyobb jellel "tulvezéreljük", a cél rendszerint az, hogy a PN átmenet, bipoláris, térvezérlésű tranzisztor a lehető legjobban "hasonlítson" az ideális kapcsolóhoz. Ennek feszültség-áram karakterisztikája a 3.8. ábra szerinti: kikapcsoláskor bármennyi is a feszültség, az ideális

kapcsoló nem enged át áramot, szakadássá válik, bekapcsoláskor viszont bármekkora áram esetén zérus a kapcsolón maradó feszültség, az ideális kapcsoló ilyenkor rövidzár. Természetes, hogy a valóságos kapcsolók esetében ezekhez az ideális állapotokhoz képest engedményeket kell tennünk. Leginkább az elektromechanikus kapcsolókkal lehet az ideális karakterisztikát (bizonyos határadatok tiszteletben tartásával) megközelíteni, az elektronikus kapcsolók fajtától függően az ideális-tól kisebb-nagyobb eltérést mutatnak. Ez az ára a nagyobb megbízhatóságnak, hosszabb (gyakorlatilag végtelen) élettartamnak, sokkal gyorsabb működésnek, kis helyfoglalásnak, integrálhatóságnak, stb. A továbbiakban ezeknek a szempontoknak az alapján minősítjük az elektronikus kapcsoló eszközöket.



3.8. ábra

A dióda kapcsolóként való felhasználásakor a legelső kérdés, ami felmerülhet: hol lehet a dióda ki-be kapcsolását vezérelni, hiszen nincsen kivezetve vezérlő elektród, amelynek villamos állapotával befolyásolni tudnánk, vezessen-e dióda vagy sem. A válasz: a dióda (PN átmenet) a rákapcsolt feszültség előjelétől teszi függővé, hogy kikapcsol vagy bekapcsol. Bekapcsolt (nyitóirányban előfeszített) állapotban az átfolyó áramtól kevéssé függő néhány tizedes voltnyi feszültség marad a diódán (Si esetén kb. 0,6 V, Schottky diódánál kb. 0,4 V, GaAs-nél kb. 1,6 V), ezzel "közelíti" a rövidzárát (3.9. ábra).



3.9. ábra

Kikapcsolt (záróirányban előfeszített) állapotban a feszültségtől elvileg széles határok között független maradékáram folyik, ez a mai PN átmeneteknél gyakorlatilag elhanyagolható, (hacsak a hőmérséklet nem különösen nagy vagy a várt specifikáció nem túl kiélezett), vagyis a dióda gyakorlatilag szakad. A diódát felhasználó kapcsolásoknak olyanoknak kell lenniük, hogy a feszültség-polaritás segítségével "közölni tudják" az illető diódával, nyisson ki vagy zárjon le. Erre példa a digitális áramkörökben legtöbbször alkalmazott VAGY kapu és az ÉS kapu. Magától értetődő hátrányai ellenére (nem állítható diódákkal elő valamennyi logikai kapcsolat, mivel invertálni nem tudnak, működésük lassu stb.) érdemes foglalkozni a diódás áramkörökkel, mivel ismeretük a mai áramkörök megértéséhez is szükséges, másrészt mert egyes helyeken ma is használják őket (kódolók, vonal-elválasztó funkciók, programozható logikák stb.). A diódás logikai áramkörök gyakori rövidítése a DDL (Diode Diode Logic).

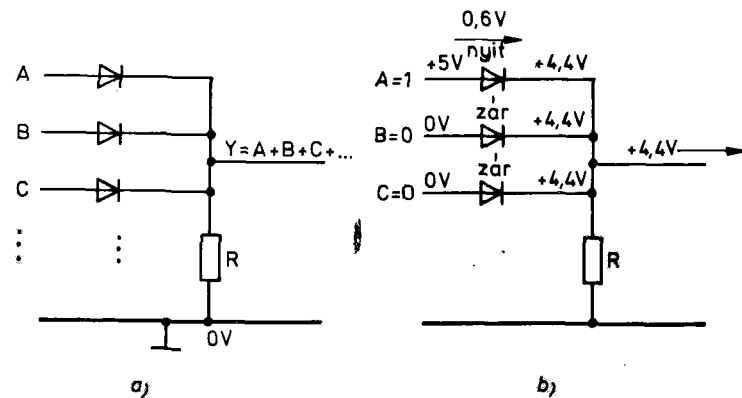
3.2.1. A diódás VAGY kapu (DDL OR-GATE)

A logikai jelszintek rögzítése az első feladat, mielőtt áramkört rajzolnánk. A mai rendszerek többsége pozitív szintű; a logikai 1-nek megfelelő U_1 feszültség pozitívabb, mint a logikai 0-nak megfelelő U_0 (rendszerint 0 V körüli) feszültség:

$$U_1 > U_0 \rightarrow \text{pozitív szintű logika.}$$

Az U_1 és U_0 térérszezeje általában nagy (pl. $+2V < U_1 < +5V$ és $0V < U_0 < +0,8V$). Negatív szintű a logikai rendszer akkor, ha az 1-nek megfelelő U_1 feszültség (tartomány) negatívabb U_0 -nál. A továbbiakban - ha külön nem említjük - pozitív szintű logikát feltételezünk.

A VAGY áramkörnek - az eredeti definíciónak megfelelően - logikai 1-et kell a kimenetén előállítania, ha a bemenetek közül akár egyetlen egyre is logikai 1-es feszültség szint érkezik. Ezenkívül fontos, hogy a bemenetek működés közben ne hassanak egymásra. Ezeknek a feltételeknek tesz eleget a 3.10. ábra pozitív szintű diódás VAGY kapuja.



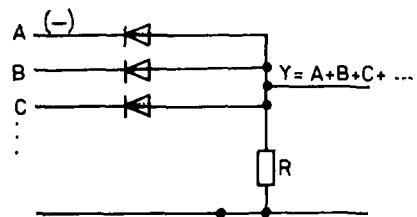
3.10. ábra

Vezérlés nélkül (0 V-os bemeneti feszültségeknél) az ellenállás a kimeneti pontot 0 V-ra huzza. Ha akár az A, akár a B...stb. bemenetre pozitív feszültség érkezik, akkor kinyit a hozzá tartozó dióda (mivel az anód pozitív feszültséget kap) és felhuzza a kimeneti közös pontot is pozitív feszültségre. Eközben a 0 V-tal vezérelt bemenetekhez tartozó diódák "automatikusan" lezárnak, kikapcsolnak (kimenet, vagyis katód+ feszültségen, anód 0 V-on, 3.10b ábra), ezek a bemenetek "leválasztódnak" az áramkörrel, vagyis a bemenetek valóban függetlenek, nem hatnak egymásra (lényegében ezért van szükség a diódákra, maga a VAGY működés egyszerű huzal-összeköttetéssel is létrejönne, de az egyik bemenet "elhuzná" a másikat!). Ahhoz

tehát, hogy a kimeneten (0,6 V-al csökkentett) pozitív logikai 1 feszültséget kapjunk, elegendő egyetlenegy bemenetnek pozitívnak lenni, ez már "magával viszi" a kimenetet is (természetesen, ha több bemenet 1 szintű, akkor a kimenet ugyanugy 1 szintű). A kimeneten csak akkor van 0, ha valamennyi bemenet 0 vezérlésű.

A VAGY kapu az öt meghajtó generátor (meghajtó hálózat) számára "fogyasztó jellegű, pozitív értelmű" terhelést képvisel, ami azt jelenti, hogy a meghajtó generátor feszültségét csökkenteni igyekszik, úgy, mint a "szokásos" terhelő ellenállások (szemben az ÉS kapuval, amely ellenkező értelmű terhelést képvisel).

Negatív logikai szint esetén az elv és a megvalósítás ugyanaz, de mivel a negatív bemeneti jelnek kell nyitnia a diódát, a bemenetekhez a katódok csatlakoznak, azaz a diódákat fordítva kell elhelyezni (3.11. ábra).



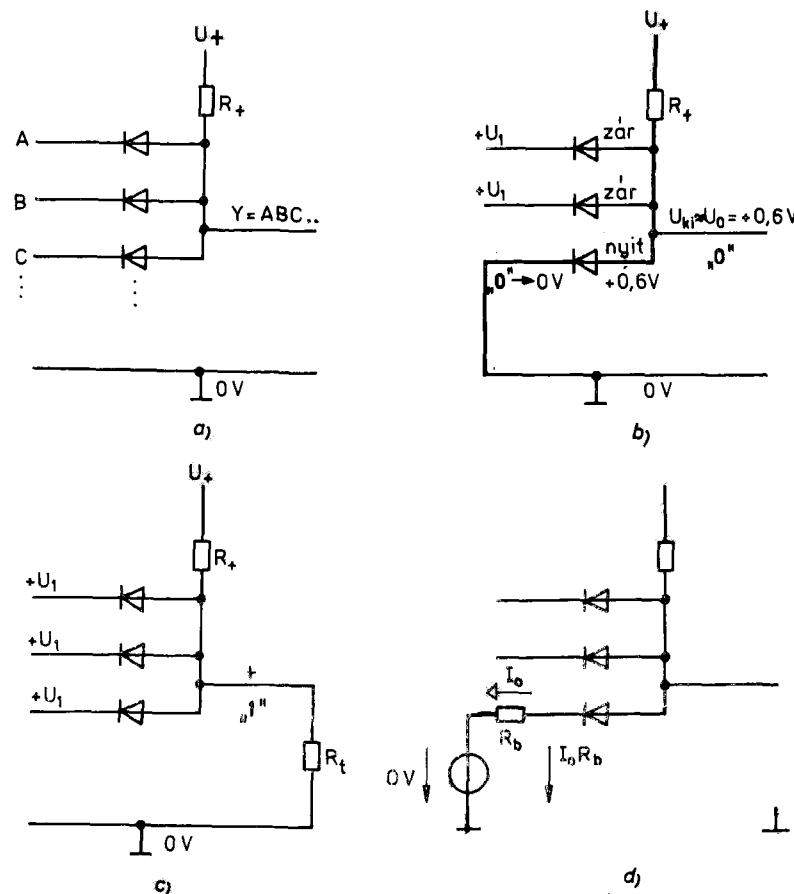
Negatív szintű VAGY kapu

3.11. ábra

3.2.2. A diódás ÉS kapu (DDL AND-GATE)

Az ÉS kapcsolat igazságtáblázatának megfelelően az áramkör kimenetén akkor jelenhet meg 1-es, ha valamennyi bemenet 1-ben van, amit úgy is meg lehet fogalmazni, hogy ha a bemenetek közül egyetlenegy is 0 szinten van, akkor már a kimenet is 0 szintű kell legyen. Ennek az előírásnak tesz eleget az 3.12a ábrán látható POZITÍV SZINTŰ ÉS-kapu. Az esetek többségében az U_+ tápfeszültség a logikai 1-nek megfelelő U_1 -gyel

körülbelül megegyező vagy annál néhány volttal nagyobb feszültség szokott lenni. Az R_+ ellenállás a kimeneti pontot igyekszik felhuzni a + tápfeszültségre.

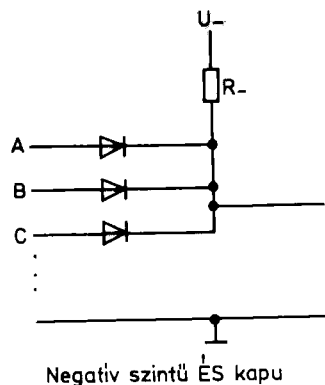


3.12. ábra

Ha a bemenetek közül egyetlenegy is földpotenciálra kötünk (3.12b ábra), akkor az illető bemenethez tartozó dióda katódja 0 V-ra kerül, miközben anódját az R_+ ellenállás huzza pozitív irányba. A nyitóirányú előfeszítés következtében az anód, vagyis a kimeneti pont feszültsége kb. +0,6 V lesz, ami logikai 0 szintnek felel meg, vagyis a földpotenciálra kötött bemenet a kimeneti pontot 0 V környezetébe lehuzza. A lo-

gikai 1-gyel vezérelt bemenetek diódái eközben záróirányu előfeszítést kapnak, hiszen anódjuk negatívabb a $+U_1$ -re kapcsolt katódnál (3.12b ábrán a két felső dióda). Ezek tehát a működésbe nem szólnak bele, a bemenetet nem terhelik (elválasztják a bemeneteket egymástól). A kimenet csak akkor lesz pozitív feszültségű az R_+ ellenállás felhuzó hatása következtében, amikor valamennyi bemenet pozitív feszültségen van, így tehát a kapu az ÉS funkciót valósítja meg (3.12c ábra).

Az ÉS kapu működéséhez, az előbbiek értelmében, tápfeszültségre is szükség van: a pozitív szintű kapuhoz pozitív feszültségre, a negatív szintűhöz negatívra (és természetesen ez utóbbinál a diódák is ellenkező polaritásúak, lásd a 3.13. ábrát).



3.13. ábra

Az ÉS kapu bemenete által képviselt terhelés eltér a "megszokott" terhelési viszonyoktól. Ha valamelyik bemenetet 0-val vezéreljük, azaz földre kötjük, akkor az U_+ -ra kötött R_+ ellenállás teljes I_0 árama a 0-ra menő vezetéken folyik a pozitív táp felől a föld felé. Amennyiben a bemenetre nem ideális rövidzár kapcsolódik, hanem egy valóságos generátor (3.12d ábra), akkor annak a belső ellenállásán folyik át az I_0 áram, és azon $I_0 R_b$ feszültséget hoz létre. Így annak ellenére, hogy a generátor forrásfeszültsége jelen esetben $U_0 = 0$ V, a kapocsfeszültség $+I_0 R_b$ pozitív feszültség lesz. A logikai 0-val vezérelt ÉS kapu bemenetet tehát az R_+ ellenállás a pozitív

feszültség irányába huzza, és a táptól a föld felé a meghajtó generátoron keresztül áram folyik, amelyet huzóáramnak (sink-current) nevezünk. Ez a huzóáram a kapocsfeszültséget növelni igyekszik, ezért az ÉS kapu bemenete logikai 0-ban negatív terhelést jelent az őt meghajtó hálózat számára. A logikai 1-ben - értelemszerűen attól függően, hogy U_+ nagyobb-e az U_1 logikai 1 szintnél vagy pedig kb. egyforma értékű vele - vagy szintén huzóáram folyik, vagy egészen kicsi záróirányu dióda áram, amely pozitív irányu. A logikai 0 vezérléskor jelenlévő bemenő huzóáram szint-torzító hatásának minimális értéken tartása érdekében az ÉS kaput meghajtó áramkörnek lehetőleg kis belső ellenállásúnak kell lennie.

Összegezve: a diódás ÉS kapu bemenetének 0-ra viteléhez, áram szükséges, amely a szokásos terhelő árammal ellenkező irányu, negatív, huzóáram.

Az előbbiekből következik, hogy ha az ÉS kapu valamelyik bemenetét üresen hagyjuk, akkor az nem logikai 0, hanem logikai 1-es vezérléssel egyenértékű, mert az üres bemeneten keresztül nem záródhat a huzóáram útja. Ez könnyen érthető, ha arra gondolunk, hogy valamennyi kapubemenetet üresen hagyva az R_+ felhuzó ellenállás a kimenetet akadálytalanul pozitív feszültségre, azaz logikai 1-re huzza.

Az ÉS kapu terhelési viszonyait azért fontos megértenünk, mert a mai digitális áramkör technikában legelterjedtebben felhasznált TTL (Transistor Transistor Logic) integrált áramkörök bemenete hasonló a diódás ÉS kapuhoz, azaz a TTL is áramhuzó logika.

Az R és R_+ ellenállás méretezésével külön nem foglalkozunk, bizva abban, hogy adott specifikációra (megengedett bemenő huzóáram, kimenő terhelőáram, megkivánt legkisebb kimeneti U_1 feszültség....) az Ohm-, és Kirchhoff-törvények ismeretében, azok következetes alkalmazásával nem okoz nehézséget az ellenállásérték meghatározása.

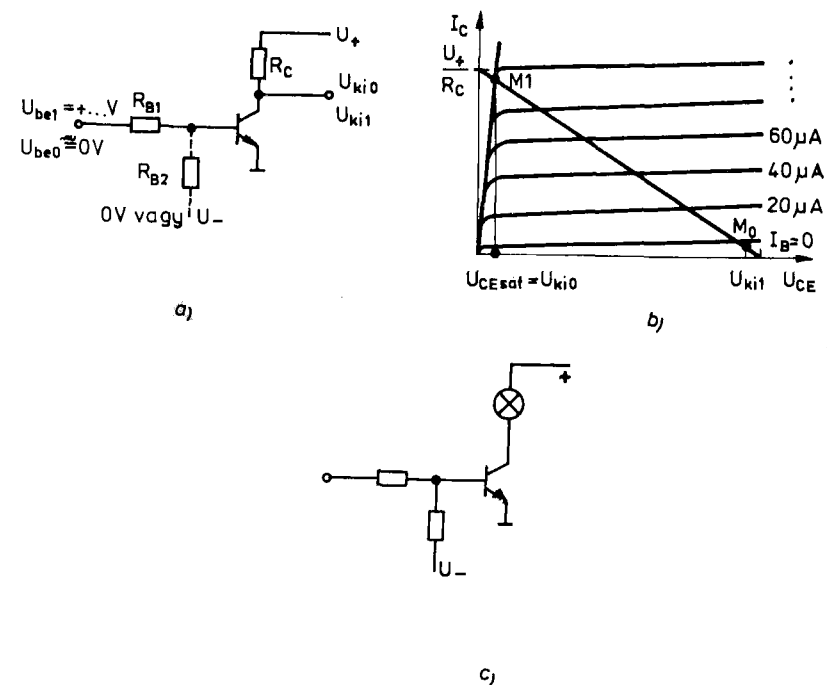
3.3. A bipoláris tranzisztorok kapcsolóüzeme, inverter

Az aktiv elemes logikai áramkör családok tanulmányozásához célszerű először a bennük levő aktiv elemek kapcsolóüzemű működését részletesen megismerni. Ehhez leginkább alkalmas a legegyszerűbb áramkör, az egyetlen aktiv elemet tartalmazó inverter, mivel a logikai áramkörök funkcionális egységeiben általában ez az erősítő elem és felépítése alapvetően jellemző egy-egy áramkör családra. Jelenleg a bipoláris tranzisztorok kapcsolóüzemét vizsgáljuk, ez fontos valamennyi bipoláris áramkör család (TTL, ECL, I^2L) működésének megértéséhez, elvárható villamos jellemzőinek becsléséhez. Tudnunk kell, hogy a különböző áramkör családok integrált áramkörös kivitelben kapható inverterei nem egyetlen tranzisztorból épülnek fel, hanem minden esetben a kapu-áramkörök egy-bemenetű változatai (amelyekben természetesen megtalálható az invertálást végző alapáramkör). Éppen ezért a gyakorlatban ritka, hogy invertert méreteznünk kellene, az viszont már többször előfordul, hogy valamilyen nagyobb teljesítmény igényű eszköz (jelző-izzó, LED, jelfogó, huzómágnes) meghajtásához az inverterrel egyező működésű áramkört kell készítenünk, adott specifikációra.

A bipoláris tranzisztor vezérelhető kapcsolóként való felhasználásakor a bázis-emitter diódára adott vezérlő jel dönti el, hogy kollektor-emittere között közel szakadást vagy közel rövidzárt képvisel-e:

Zérus vagy negatív bázis-emitter előfeszítés esetén a kapcsoló kikapcsol: a kollektor-emitter között - a környező áramkörtől szinte függetlenül - csak egészen kicsi (mai Si eszközöknél gyakorlatilag elhanyagolható) maradékáram ($\ll I_{CE0}$) folyik, a kollektor és az emitter között gyakorlatilag szakadás van. Nyitóirányú előfeszítéskor a vezérlő feszültség és a bázisellenállás által meghatározott (a lineáris üzemmódhoz képest rendszerint nagyra választott) bázisáram folyik, a tranzisztor kinyit, "összeköti az emitterét és a kollektorát", az áram a kollektor és az emitter között szinte akadálytalanul folyhat, a tranzisztoron kis feszültség marad.

A viszonyokat legjobban a telítési üzemi inverteren tanulmányozhatjuk. Ez lényegében földelt emitteres kapcsolás, amely nagyszintű bemeneti jelet dolgoz fel, éppen ezért hiányoznak a lineáris erősítőtechnikában megszokott munkapontbeállító elemek (bázisosztó, emitterellenállás), amelyek az erősítő tranzisztor ott aktiv üzemben tartják. A legegyszerűbb, NPN tranzisztoros ún. "telítési", "határolt" (szaturált, saturated) üzemmódu inverter a 3.14a ábrán látható.



3.14. ábra

A vezérlést R_{B1} -en keresztül kapja, R_{B2} a biztonságos lezáráshoz segíti hozzá a tranzisztor, a bázist 0 V-ra vagy negatív zárófeszültségre huzva. A mai kis visszáramú tranzisztorok legtöbbször nem igénylik a beépítését. Pozitív szintű logikát feltételezve a bemenetre vagy U_{be0} kb. 0 V-os logikai 0 feszültség, vagy U_{bel} , pozitív néhány voltos logikai feszültség érkezik. A kétféle vezérlés esetére az egyenáramú működés a következő:

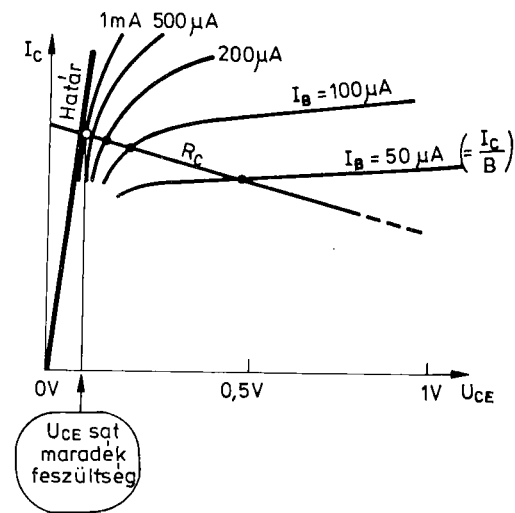
Logikai 0 (≈ 0 V) feszültségű vezérlés esetén a tranzisztor bázis-emitter diódája lezár, a kollektoráram gyakorlatilag 0-ra csökken, a munkapont, M_0 a munkaegyenesen az $I_B = 0$ karakterisztika vonal alatt van (3.14b ábra). Ilyenkor az I_{CER} ($R=R_B$) kollektor maradékáram folyik, amely kisebb, mint az $I_B = 0$ -hoz tartozó I_{CEO} , és amely kisteljesítményű szilícium kapcsoló tranzisztoroknál általában elhanyagolhatóan kicsi. Ez az R_C , munkaellenálláson átfolyva elhanyagolhatóan kis feszültséget hoz létre rajta, így a kimeneti logikai 1 feszültség: U_{kil} terhelés nélkül gyakorlatilag megegyezik a tápfeszültséggel. Végeredményben a logikai 0 bemeneti jelre logikai 1 a válasz a kimeneten.

Ha a lezárt tranzisztor kollektorárama nem elég kisértékű, akkor az U_{kil} kimeneti feszültség is kisebb, ez pedig kedvezőtlen a kapott logikai 1 feszültség szempontjából. Ilyenkor (különösen Ge tranzisztoros áramkörben) célszerű az R_{B2} ellenállást elhelyezni.

Meghajtó áramkör esetében, amikor a munkaellenállás helyén valamilyen "fogyasztó" (izzó, LED, stb.) van (3.14c. ábra) a nagy áramu tranzisztor esetleges bizonytalan lezárása azt eredményezi, hogy pl. a jelzőlámpa, amelynek ebben a vezérlési helyzetben ki kellene aludnia, halványan világít. A viszonylag nagy kollektor maradékáram és U_{CE} feszültség miatt a tranzisztor disszipációja is nagyobb lesz, ami a maradékáram további exponenciális növekedését idézi elő, ami tovább növeli a disszipációt stb. A folyamat végeredménye az is lehet, hogy az izzó gyakorlatilag teljes fényel világít és a tranzisztor tönkremegy. Ezért fontos - különösen meghajtó áramkörök esetében, melyet gyakran magunk tervezünk - biztos lezárásról gondoskodni (katalógusadatok figyelembevételével, ha szükséges R_{B2} elhelyezésével és U_{alk} alkalmazásával). R_{B2} és U_{alk} sokszor olyankor is szükséges, amikor a gyors lezárás a fontos.

Logikai 1 ($+U_{be1}$) feszültségű vezérlés esetén a biztos átkapcsolás érdekében a tranzisztor telítéssel (szaturált) üzemmódban működik (ez az ún. "szaturált" logikai áramkör családokra érvényes, de a "nem szaturált" áramkörökről is szó lesz a későbbiekben). A pozitív bemeneti feszültség hatására az R_{B1}

(ill. R_{B1} és R_{B2}) által meghatározott nagyságú bázis nyitóáram indul meg. Az ennek hatására létrejövő kollektoráram a munkaellenálláson feszültséget hoz létre. Mivel a bázisáram - kapcsolóüzemről és telített üzemről lévén szó - többszöröse annak, mint ami lineáris erősítő üzemben szokásos, a kollektoráram is olyan nagy, hogy a munkaellenálláson gyakorlatilag a teljes tápfeszültség lép fel, a tranzisztoron max. 100 mV nagyságrendű maradékfeszültség (szaturációs feszültség) van (3.14b ábra M_1 munkapont). A kimenet tehát gyakorlatilag 0 V-on, logikai 0 szinten van. Éppen ezért, hogy a kimenet feszültsége minél közelebb legyen 0 V-hoz, vagyis, hogy a tranzisztor minél jobban kinyisson, célszerű a bázisáramot a lineáris üzemben szokásos értéknél jóval nagyobbra választani. Így a munkapont - mivel nagy bázisáramhoz tartozó $U_{CE} - I_C$ karakterisztikán van, közelebb kerül a határegyeneshez, a maradék feszültség a 0 V-hoz (3.15. ábra).



3.15. ábra

A telítéssel üzemmód egyik alapvető ismerve ezek szerint:

$$I_B \gg \frac{I_C}{\beta}$$

vagyis a bázisáram sokkal nagyobb, mint az adott kollektoráramhoz a lineáris üzemmódban szükséges érték. A soros bázisáram-korlátozó ellenállást R_{B1} -et a bázisáram megválasztása után Ohm-törvényéből számíthatjuk:

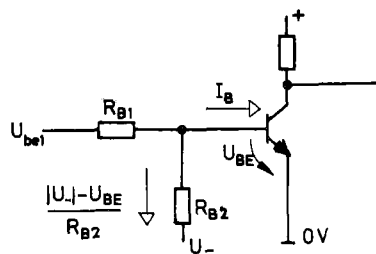
$$R_{B1} = \frac{U_{be1} - U_{BE}}{I_B} \approx \frac{U_{be1} - 0,6 \text{ V}}{I_B}$$

(Ha negatív feszültségre kapcsolódó R_{B2} is van (3.16. ábra), akkor, mivel:

$$I_B = \frac{U_{be1} - U_{BE}}{R_{B1}} - \frac{|U_-| + U_{BE}}{R_{B2}}$$

ebből

$$R_{B1} = \frac{U_{be1} - U_{BE}}{I_B R_{B2} + |U_-| + U_{BE}} R_{B2}$$



3.16. ábra

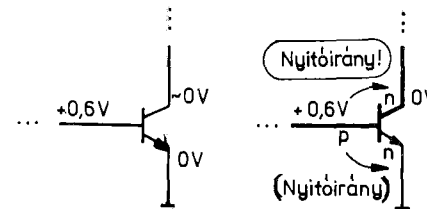
Ezenkívül célszerű R_{B1} , R_{B2} arányát úgy megválasztani, hogy $U_{be} = 0 \text{ V}$ -nál a bázis zárófeszültség $-2 \dots -3 \text{ V}$ körül legyen:

$$U_- \cdot \frac{R_{B1}}{R_{B1} + R_{B2}} = -2 \dots -3 \text{ V}$$

A két utóbbi egyenletből R_{B1} és R_{B2} meghatározható.

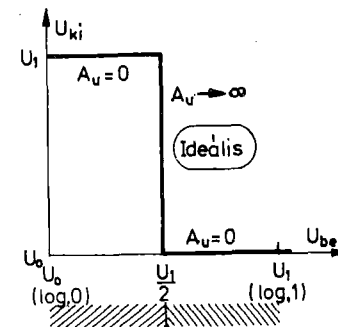
A telítéses üzemmód másik ismérve, hogy nyitáskor, logikai 1-gyel vezérelve a tranzisztor elektródjai egészen sajátos potenciálon vannak: NPN tranzisztor esetén a bázis feszültsége a 0 V-os emitterhez képest kb. +0,6 V, a kollektor feszültsége 10...100 mV nagyságrendű, gyakorlatilag 0 V (3.17. ábra). NPN rétegsorrendnél ez azt jelenti, hogy mivel a P bá-

zisréteg +0,6 V-on van, valamint az N emitter és az N kollektor egyaránt 0 V-on van, nemcsak a bázis-emitter dióda, hanem a kollektor-bázis dióda is nyitó irányban van előfeszítve. Ez lényegesen eltér a lineáris üzemmódban szokásos záróirányú kollektor-bázis feszültségű munkaponti beállítástól. Az ilyenkor fellépő, a tranzisztor lezárását lassító töltéstárolási effektussal később foglalkozunk, a dinamikus működés tárgyalásakor.

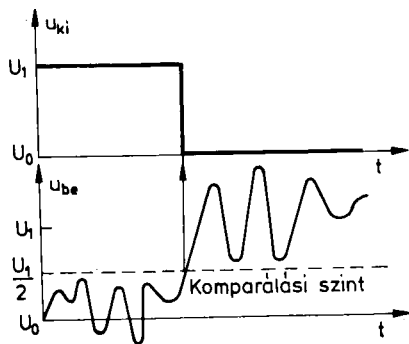


3.17. ábra

Az egyenáramu működés fontos jellemzője az $U_{ki} - U_{be}$ transzfer karakterisztika vagy komparációs jelleggörbe. Ez az adott kapcsolás (rendszer) kimeneti feszültségét adja meg a bemeneti feszültség függvényében. Az ideális inverter olyan, hogy a bemeneti feszültség tartományban a logikai 1 vezérlő feszültség felénél van a határ, a "komparálási szint", az ennél kisebb feszültséget logikai 0-nak, az ennél nagyobbat logikai 1-nek érzékeli, és ennek megfelelően szolgáltatja a kimenetén az invertált jelet (3.18. ábra).

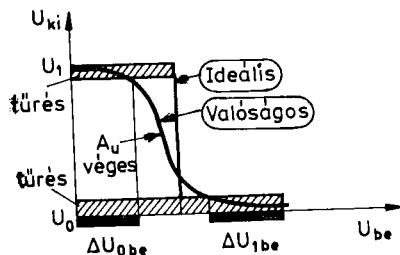


3.18. ábra



3.19. ábra

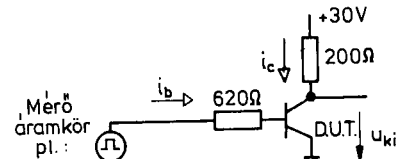
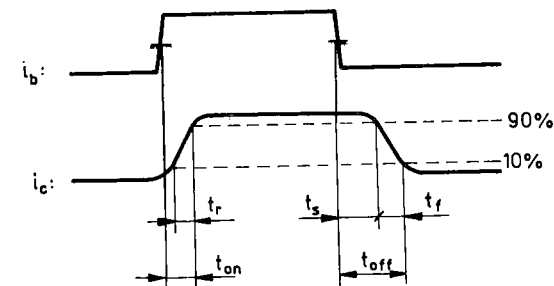
Igy a legnagyobb az áramkör zajvédetség, hiszen a logikai 0 és a logikai 1 vezérlőfeszültségre max $U_1/2$ csúcserőtelű zajfeszültség superponálódhat anélkül, hogy a kimeneti szint tévesen állna be (3.19. ábra). A valóságos inverterek transzfer karakterisztikája annál jobb, minél jobban megközelíti az ideálist. A váltás, az átmenet logikai 0-ból logikai 1-be a valóságban nem ugrásszerű (a feszültségerősítés nem végtelen nagy) és a komparációs szint sincs egészen pontosan az U_1 feszültség felénél (3.20. ábra). Ezek miatt a zajvédetség is csökken, mert kisebb lesz a megengedett logikai 0 és logikai 1 vezérlőfeszültség tartomány (3.20. ábra ΔU_{0be} és ΔU_{1be}).



3.20. ábra

A bipoláris tranzisztoros inverter dinamikus működését nem analizáljuk részletesen, csak a felhasználó szempontjából legfontosabb jellemzőket, specifikációkat említjük. Inverte-

reket ui. vagy készen, integrált formában használunk fel, amelyek specifikációját a katalógusokból megtudhatjuk, vagy esetleg speciális célra mi építünk. Ilyenkor előzetes nagyságrendi kalkuláció után, szükség esetén méréssel állapítjuk meg a dinamikus jellemzőket.

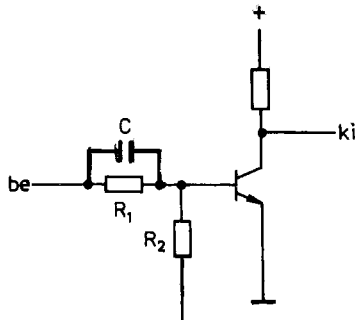


- t_{on} bekapcsolási idő
- t_{off} kikapcsolási idő
- t_r felfutási idő (koll.áramra vonatkozóan)
- t_f lefutási idő (koll.áramra vonatkozóan)
- t_s tárolási idő(!)

3.21. ábra

Bármely áramkörcsaládról is legyen szó, a benne levő tranzisztor véges határfrekvenciájú elem. Lineáris üzemben a lineáris kisjelű helyettesítő képeket és az f_α , ill. f_β határfrekvenciákat használtuk. Kapcsoló üzemben a kisjelű helyettesítő képek nem használhatók, mert mint tudjuk, elemeik munkapontfüggőek és amíg a munkapont M_0 -ból M_1 -be megy (l. a 3.14b ábrát), értékük több nagyságrenddel megváltozik. Itt a kapcsolási idők, valamint a fel-, és lefutási idő megadása célszerű. Kapcsoló tranzisztorokra ezeket az időadatokat a katalógusok egy adott és mellékelte mérőkapcsolásra vonatkozóan közlik (3.21. ábra). A bemenő bázisáramot ideális négyzet alakú

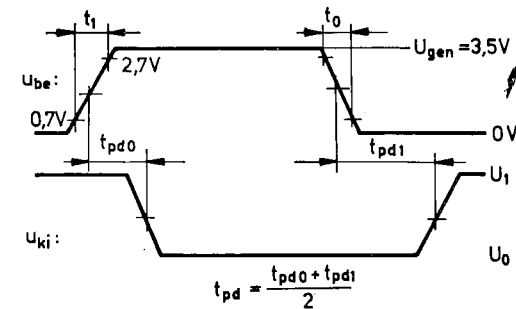
kunak feltételezve megadják, hogy mennyi idő alatt kapcsol be a tranzisztor (t_{on}), vagyis, hogy a bemeneti jel élétől számítva mennyi idő alatt éri el a kollektor árama a végérték 90%-át. Ugyanígy megadják a kikapcsolási időt (t_{off}), azt az időt, ami alatt a bemenet lezárását követően a kollektor árama eléri a névleges érték 10%-át. A felfutási idő (t_r) a kollektor áram 10%-ról 90%-ra növekedésének ideje, a lefutási idő a 90%-ról a 10% eléréséig eltelt idő (adott két bázisáram szélsőérték között). Gyors kapcsoló tranzisztorokra ezek 10ns, néhányszor 10 ns nagyságrendűek. A mellékelt mérőkapcsolásban a kollektor áramra a munkaellenálláson levő feszültségből, a bázisáramra a bemeneti feszültségből (feszültség-ugrásból) és a bázis ellenállásból következtetnek (a meghajtó generátor fel-, ill. lefutási idejét legtöbbször 1 ns alatti értékre írják elő). Vigyázat! a katalógusban közölt adatok csak a katalógusban mellékelt mérőkapcsolásban működő tranzisztorra érvényesek, más kapcsolásban mások lesznek a kapcsolási idők is!



3.22. ábra

A "diszkrét elemes" (tranzisztoros) és a korai integrált áramkörös technikában a telítésszerű inverterek működési sebességét gyorsító kondenzátorral növelték. A bázisáram-korlátozó ellenállással párhuzamosan kapcsolt viszonylag kisértékű (10...100 pF) kondenzátor (3.22. ábra) a bemeneti feszültség pozitív értékre ugrásának pillanatában rövidzárként viselkedik és nagy árammal tulvezérli a tranzisztort, így a bekapcsolási idő csökken. Ezután a bázisáram az ellenállás által meghatározott értékre áll be. A vezérlőfeszültség nullára ug-

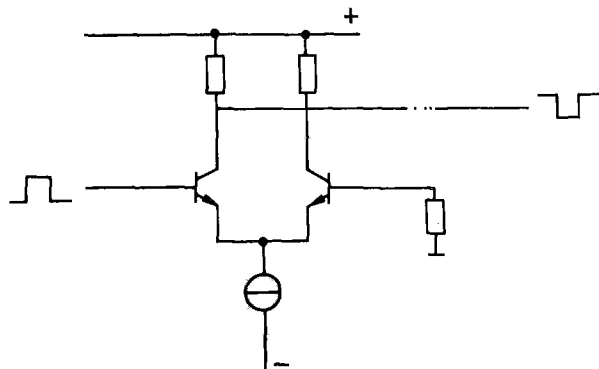
rásakor - a kondenzátor az első pillanatban "telep" lévén, +0,6 V-ról lezárásba viszi a bázist, így a tárolási idő is csökken. Hátrány, hogy a gyorsító kondenzátor növeli az inverter nagyfrekvenciás AC zavarérzékenységét, mivel a nagyfrekvencián gyakorlatilag rövidzárként viselkedő kondenzátor a bázisra vezeti a logikai jelre szuperponálódott zavarokat, és téves átkapcsolás jöhet létre.



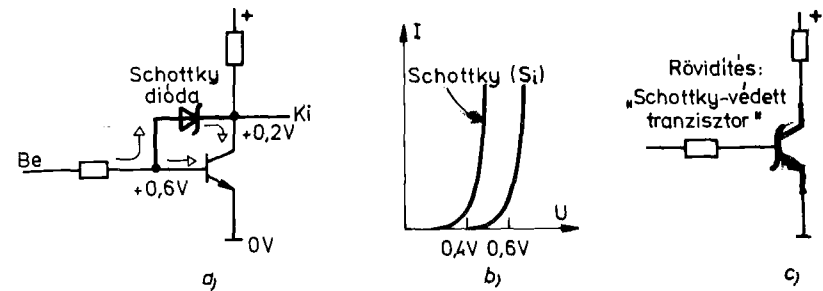
3.23. ábra

Az integrált áramköri inverterek (ÉS kapuk, valamint egyéb építőelemek) időadatait a tranzisztorokétól kissé eltérően adják meg a katalógusok. Egy példát a 3.23. ábra mutat. Az értékek egy megadott mérőkapcsolásra érvényesek, amelyben a bemenetre kapcsolt generátor feszültségét és jelváltáskor a fel-, ill. lefutási idejét előírják (U_{gen} , t_1 és t_0). A kimenetre adott "válasz" késési ideje a bemeneti jelátmenet 50%-os pontjától a kimeneti jel 50%-os pontjáig számít, elnevezése: jelterjedési késleltetési idő (propagation delay). A kimenet 0-ba menetéhez (t_{pd0}) és 1-be menetéhez (t_{pd1}) tartozó késleltetési idő rendszerint nem egyforma, ezeket külön megadják az adatlapokon. Van, amikor az átlagos jelterjedési késleltetési időt adják meg, ez a kétkésleltetési idő számtani középértéke. Logikai áramkörök felhasználása szempontjából a késleltetési idő a legfontosabb sebességre vonatkozó adat, és nem tévesztendő össze a jel felfutási és lefutási idővel, amelyet legtöbbször nem is közölnek. A telítésszerű inverterek t_{off} , ill. t_{pd1} kikapcsolási ideje rendszerint sokkal hosszabb a többi időnél. Java részét a t_s tárolási idő (storage time) teszi ki

Amikor a tranzisztor vezet és tulvezérelt, telitéses munkapontban van, akkor - mint láttuk - a kollektor-bázis dióda nyitó irányban van előfeszítve (a kollektor is emitterként viselkedik). Emiatt a bázisban igen sok töltéshordozó halmozódik fel. Ha a tranzisztort hirtelen lezárjuk (a bázisra zérus vagy záró feszültséget adunk), akkor a kollektoráram mindaddig nem csökken nullára, amíg a bázisból a tárolt nagymennyiségű "fölös" töltés el nem távozik. A telitéses üzemenek tehát ez a hátránya: a kikapcsolási idő a tárolási idővel meghosszabbodik. Ez általában csak akkor okoz problémát, ha olyan gyors áramköröket akarunk építeni, amelyek késleltetése ns, vagy ns alatti nagyságrendben van. Ma már ilyen igen gyors rendszerek is léteznek. Ezekben a tranzisztorok telítésbe vezérlését valamilyen módon megakadályozzák ("nemszaturált", - non saturated - logikák). Egyik megoldás az olyan kapcsolási elrendezés, amely eleve megakadályozza a telítési áram kialakulását, a kollektor-bázis dióda kinyitását pl. differenciál erősítőhöz hasonló, "majdnem lineáris" működéssel. Ezzel később találkozunk az ECL (Emitter Coupled Logic = emitter csatolt logika) tárgyalásakor (3.24. ábra). A másik megoldás szerint az inverter kapcsolást egy diódával úgy egészítik ki, hogy a kollektor-bázis dióda ne tudjon kinyitni (3.25a ábra). Ehhez olyan fajta dióda kell, amelynek nyitófeszültsége kisebb a kollektor-bázis (Si) dióda nyitófeszültségénél. Ilyen a Schottky dióda, amely kb. 0,3...0,4 V-nál nyit ki (3.25b ábra).



3.24. ábra

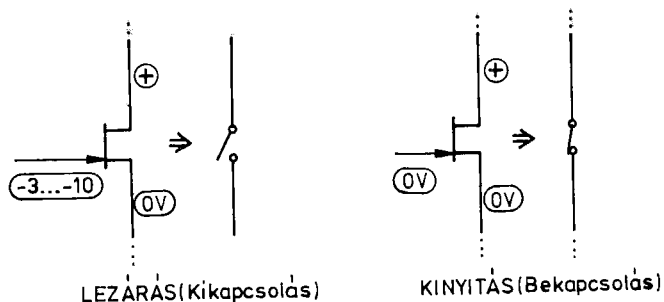


3.25. ábra

Amikor az inverter bemenetére pozitív logikai 1 feszültség érkezik, a bázis feszültsége 0,6 V körüli lesz. A kollektor feszültsége nem csökkenhet közel 0 V-ra, mert a "megfogó dióda" kinyit és mivel anódja a bázison +0,6 V-on van, a kollektor potenciálját nem engedi +0,2 V-nál lejjebb csökkenni. A Schottky dióda a kinyitás után a soros bázis korlátozó ellenállás áramának nagyrésztét is elvezeti a kollektor felé, így a bázisáram csak akkora lesz, amekkorának a kb. +0,2 V-os kollektorfeszültség eléréséhez kell lennie (és a dióda árama is a kollektoráramhoz adódik). A tranzisztor tehát nem lesz telítésben, így lezárása sokkal rövidebb idő alatt végbemehet. A lezárást a dióda nem akadályozza, mert a pozitív kollektorfeszültség záróirányba feszíti elő. A Schottky diódás integrált áramkör változatokat ma már elterjedten alkalmazzák (pl. Schottky-TTL, Schottky védett I²L, stb.), működésüket a későbbiekben tárgyaljuk. A "hétköznapi" sebesség követelményeknek általában megfelelnek az olcsóbb telitéses áramkörök is, az ismertetett nem telített változatokat akkor használjuk, ha különlegesen gyors vagy viszonylag kis felvett teljesítmény mellett normál sebességű működésre van szükség (pl.: Low-power Schottky TTL: kisteljesítményű Schottky TTL). Az integrált áramkörökben a Schottky diódákat egyetlen fázisban diffundáltatják a védett tranzisztorok kollektor-bázis átmenetéhez, nem jelennek meg külön alkatrészként. Ezeket a "Schottky-védett tranzisztorokat" rövidített jelöléssel rajzolják a kapcsolásban (3.25c ábra).

3.4. A térvezérlésű eszközök kapcsolóüzeme, MOS inverter

A térvezérlésű tranzisztorokat akár digitális, akár analóg áramkörökben egyaránt nagyon gyakran használják kapcsolóként, tekintve, hogy a FET-ek működési elvükből következően a vezérlő GATE feszültségtől függő ellenállásoknak tekinthetők. A kapcsolóüzem két szélsőséges esete: teljesen lezárva gyakorlatilag végtelen r_{kik} , teljesen kinyitva tipustól függően néhány ohm ... néhány kiloohm r_{bek} csatorna ellenállást képvisel a FET.



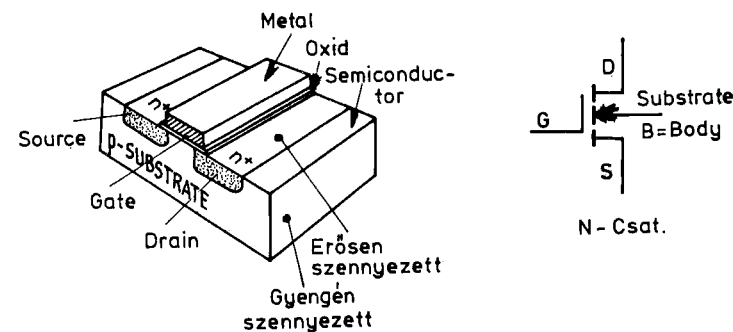
3.26. ábra

A záróréteges FET, JFET "teljes kinyitásához", azaz minimális r_{bek} eléréséhez 0 V-os Gate-Source (bázis-emitter) vezérlőfeszültség szükséges. Lezáráshoz a típusra, példányra jellemző elzáródási feszültségnél nagyobb abszolút értékű, N-csatornás típusnál negatív (több voltos, esetleg 10 V-os) feszültséget kell a Gate és a Source közé adnunk. A kapcsolókét pólusa, két "vége" a Drain (kollektor) és a Source (emitter), működés közben N-csatornás típus esetén mindig a Drain-nek kell pozitívabbnak lennie. A Source potenciálját 0 V-nak véve a lezáráshoz, ill. nyitáshoz szükséges viszonyokat mutatja a 3.26a és b ábra.

A záróréteges FET digitális áramkörökben nem jól alkalmazható. Ha invertert készítenénk belőle, akkor a működtetéshez (N-csatorna esetén) pozitív tápfeszültségre lenne szükség, a bemeneti vezérléshez viszont zérus, ill. negatív feszültség. Így módon egy inverter kimeneti (0 V vagy +táp)feszültsége

nem lenne alkalmas közvetlenül egy ugyanilyen inverter fokozat (0 V-os és negatív feszültséget igénylő) vezérlésére. Márpedig digitális áramkörökben alapvető követelmény, hogy egy áramkör kimenete képes legyen egy következő fokozat bemenetét közvetlenül, szint-áttétel nélkül meghajtani. Ezért a JFET fő alkalmazási területe a precíz analóg kapcsoló (kedvezően kis r_{bek} érhető el), bár a vezérlés itt sem egyszerű (részletesen l. a későbbiekben!).

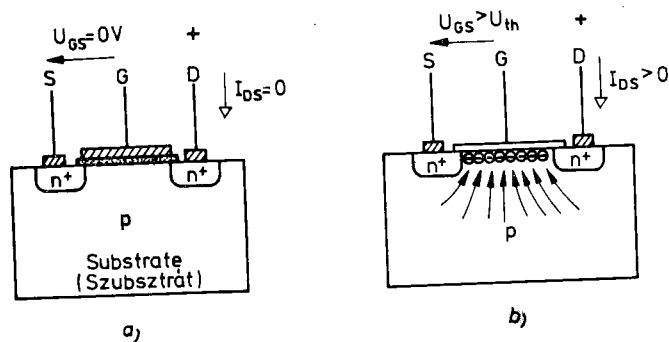
A MOSFET, szigetelt vezérlőelektródás FET a mai térvezérlésű digitális áramkörök szinte kizárólagos alkateleme. A MOS elnevezés a rétegsorrendre utal, Metal (fém) a vezérlő elektróda, Oxid a szigetelés a vezérlő elektróda alatt és Semiconductor (félvezető) az áramvezető csatorna. Általánosabb a MIS-FET (Metal-Insulator-Semiconductor) vagy az IG-FET (Insulated Gate FET) megnevezés, amely a működési elv lényegére mutat; a vezérlő elektróda el van szigetelve a félvezetőtől. A mai eszközök túlnyomó részében oxid a szigetelő, ezért az irodalomban, katalógusokban a MOS megjelölést használják leggyakrabban (még akkor is, ha más anyagok is jelen vannak a szerkezetben: nitrid, poliszilikon stb.).



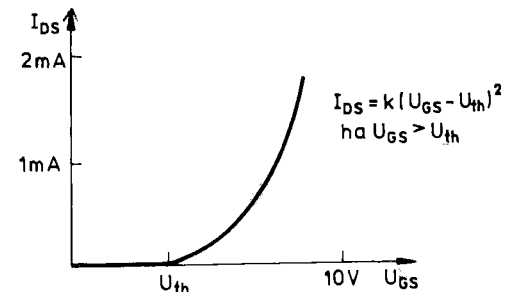
3.27. ábra

Aktiv, kapcsoló tranzisztorként az előbb említett összekapcsolhatóság érdekében rendszerint növekményes (dusitációs, enhancement) típusu tranzisztort alkalmaznak. Az N-csatornás tranzisztor szerkezetét és rajzjelét a 3.27. ábrán idézzük fel. Működésére jellemző, hogy 0 V-os gate-source előfeszítésnél a

csatorna (a drain-source közötti tartomány) gyakorlatilag szakadást képvisel. Hiába pozitív a drain potenciálja, áram gyakorlatilag nem jön létre, a drain-szubsztrát átmenet záróirányu, és nincs töltéshordozó, amely áramot szállítana (3.28a ábra). A gate-source közé pozitív, a küszöb szintet (U_{th}) meghaladó feszültséget adva a tranzisztor vezet, a "csatorna", amely a pozitívan feltöltött Gate alatt összegyűlt töltéshordozók (elektronok) hatására alakult ki, néhány 100Ω ... néhány $k\Omega$ bekapcsolási ellenállást mutat (3.28b ábra). Az U_{GS} - I_{DS} tranzsfer karakterisztikát a 3.29. ábra idézi fel; amíg a vezérlőfeszültség kicsi, a kollektoráram, I_{DS} zérus, ha a vezérlőfeszültség eléri a küszöbfeszültséget (threshold: U_{th}), akkor az áram (állandó U_{DS} -et feltételezve) négyzetes törvény szerint növekedni kezd, a tranzisztor kinyit. A növekményes tranzisztor tehát 0 V körüli bemeneti vezérlőfeszültségre lezár és kinyitására ugyanolyan előjelű (N-csatornás esetén pozitív) feszültségre van szükség, mint amilyen a kimeneti U_{DS} feszültség, ill. a tápfeszültség előjele. Ez teszi lehetővé, hogy az egymás után levő fokozatokat, egységeket közvetlen csatolással összekössük, az adott fokozat kimeneti jele, drain-source feszültsége egyenlő lehet a következő fokozat bemeneti, gate-source feszültségével.

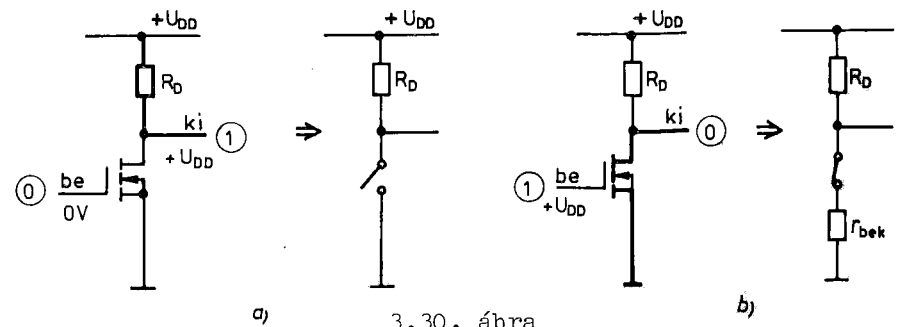


3.28. ábra



3.29. ábra

Az N-csatornás "NMOS" növekményes tranzisztoros egyszerű inverter működését a 3.30. ábra szemlélteti. Ha a bemenetre 0 V körüli feszültséget adunk, akkor a tranzisztor lezár (lásd a tranzsfer karakterisztikát), gyakorlatilag szakadássá válik, kikapcsol. A kimenet feszültsége így egyenlő lesz a $+U_{DD}$ tápfeszültséggel. Valós terheléssel itt nem kell számolni, ha feltételezzük, hogy az inverter újabb MOS áramkört hajt meg (egyenáramu szempontból üresjárásra dolgozik, 3.30a ábra).

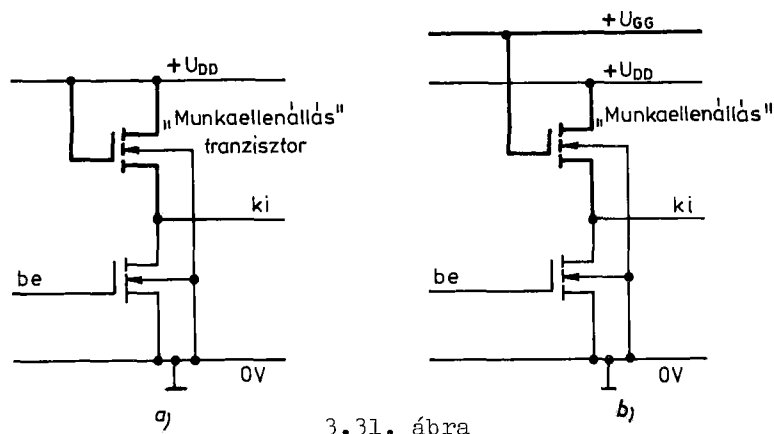


3.30. ábra

Logikai 1 szintű, pozitív, közel $+U_{DD}$ vezérlő feszültség hatására a Gate pozitív feszültsége miatt (korlátozó ellenállás nem szükséges, hiszen a vezérlő elektród szigetelt!) a tranzisztor vezetővé válik, bekapcsol (3.30b ábra) és a kimeneti feszültség elvileg zérus lesz. Valóságban a MOS-tranzisztor bekapcsolt állapotban nem rövidzárként, hanem $k\Omega$ nagyságrendű ellenállásként viselkedik, így a kimeneti feszültség sem pontosan 0 V, hanem a bekapcsolt MOS-csatornaellenállás -

(r_{bek}) és a munkaellenállás, mint feszültségosztó által meghatározott feszültség lesz. Tekintve, hogy a munkaellenállást akár $100\text{ k}\Omega$ nagyságrendűre is választhatjuk (a kis fogyasztás érdekében, és mert valós terheléssel nem kell számolni), a kimeneti feszültség logikai 1 vezérléskor közel lesz a 0 V-hoz.

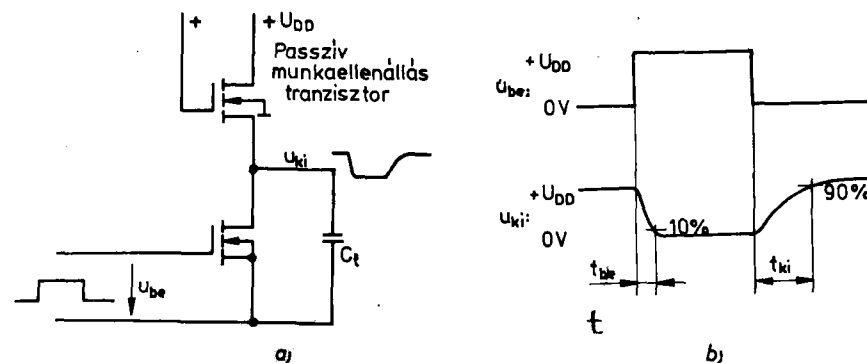
Integrált MOS rendszerekben az inverterek munkaellenállását is MOS tranzisztorokból készítik. A száz $\text{k}\Omega$ nagyságrendű ellenállások integrált formában való előállításai nehézkes és az integrált áramköri lapkán is túlzottan nagy lenne a helyfoglalásuk. A MOS tranzisztor szerkezete egyszerűbb is, mint egy monolitikus ellenállásé (l. a 3.27. ábrát). A munkaellenállás-tranzisztor rendszerint egy "félleg lezárt", nagy ellenállást képviselő szintén növekményes típus (újabb áramkörökben kiürítéssel is használják). A geometriája olyan, amelynek eleve nagyobb a csatorna ellenállása. A kívánt ellenállás beállítása konstans Gate feszültségre, általában az U_{DD} tápfeszültségre vagy az IC-ben előállított külön U_{GG} -re kötéssel megy végbe (3.31a és b ábra).



3.31. ábra

Összehasonlítva a bipoláris és a MOS invertert annyit már most, részletes analízis nélkül megállapíthatunk, hogy a MOS inverterek (és természetesen általában a MOS rendszerek) teljesítmény fogyasztása sokkal kisebb a bipolárisnál. Ennek magyarázatát az előzőekben láthattuk: MOS áramkörökben az adott fokozatot a következő fokozat nem terheli

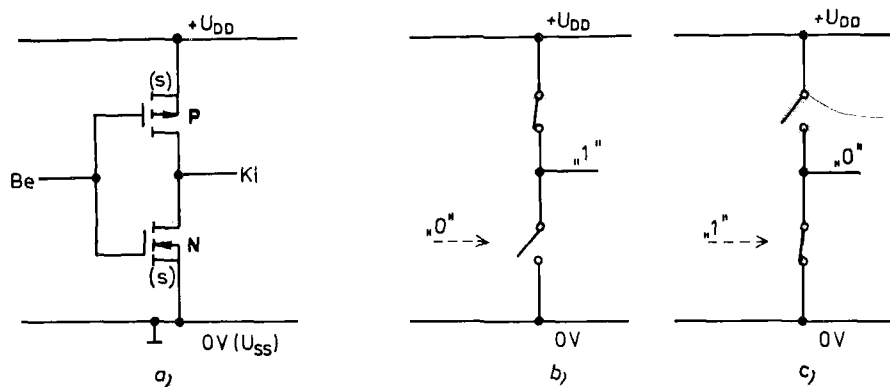
(statikusan) a bemeneti áramával, így a munkaellenállást nagyra lehet választani, ami adott tápfeszültség esetén kisebb kollektoráramot (drain áramot) hoz létre, az $U \cdot I$ szorzat kisebb lesz, kisebb lesz a táp-teljesítmény. A nagyobb munkaellenállás viszont érthető módon a dinamikus működésre hat rossz irányban: az adott áramkört terhelő következő MOS bemenet "ohmos" terhelést nem képvisel ugyan (szakadás), de meglehetősen nagy a bemeneti kapacitás a Gate és a csatorna között.



3.32. ábra

Ezt a terhelő kapacitást (3.32a ábra) a nagyértékű munkaellenállásnak kell feltöltenie, így az időállandó nagy lesz, a működés lelassul! A jelalak a 3.32b ábra alapján követhető: a bemenetet ugrásszerűen 0-tól 1-be vezérelve az aktív, inverter-tranzisztor kis impedanciássá válik és kisüti a közel tápfeszültségre töltődött terhelő kondenzátort. Az időállandó viszonylag kicsi (a kisütő ellenállás a tranzisztor r_{bek} ellenállása), a bekapcsolási idő (t_{be}) viszonylag rövid (természetesen hosszabb, mint a bipoláris tranzisztoros inverternél). Kikapcsoláskor, a bemeneti feszültség 1-ről 0-ra ugrásakor viszont az inverter tranzisztor szakadássá válik és a passzív, nagyértékű munkaellenállás csak lassan képes a terhelő kondenzátort 0 V-ról U_{DD} -re feltölteni (a töltő ellenállás most $100\text{ k}\Omega$ nagyságrendű!), így a kikapcsolási idő (t_{ki}) jóval hosszabb lesz. Végül soron ez korlátozza a MOS rendszer sebességét: a kikapcsolási idő néhány tized μs -ot (néhány száz ns-

ot) is elérhet, szemben a bipoláris áramkörök néhányszor 10 ns-os, esetleg néhány ns-os sőt ns alatti kapcsolási időivel. A MOS áramkörök esetében tehát mindig kompromisszumot kell kötni a sebesség és a teljesítmény között, a munkaellenállások - (ill. a passzív "munkaellenállás tranzisztorok" munkapontjának) megválasztásával. Alapvető javulást csak úgy lehet elérni, hogy a passzív felhuzó MOS tranzisztor helyett aktív, az alsó, inverter tranzisztorral ellenütemben működő tranziszort helyezünk el, ezen az elven működik a CMOS, komplementer MOS (Complementer-MOS) inverter, amelyben egy kapcsoló és egy munkaellenállás helyett két kapcsoló van. E két kapcsoló ellenütemben dolgozik, egyik a 0 V-ot a másik a $+U_{DD}$ tápfeszültséget viszi a kimenetre. Azért, hogy közös vezérlőjellel jöhessen létre az ellenütemű működés, az alsó és felső kapcsolót ellentétes polaritású MOS tranzisztorral valósítják meg (komplementer párral). A CMOS inverter alapkapcsolást a 3.33a ábra mutatja.



3.33. ábra

Amikor a bemenet (gate-ek) feszültsége 0 V körüli, az N-csatornás tranzisztor kikapcsol, gyakorlatilag szakadás lesz, olyan mintha jelen sem lenne. Ugyanakkor a P-csatornás tranzisztor bekapcsol, kinyit, hiszen a 0 V-os bemeneti feszültség a pozitív tápfeszültségen lévő source-hoz képest negatív, nyitóirányú vezérlő feszültséget jelent. A felső, bekapcsolt kapcsoló a kimenetet a pozitív tápfeszültséggel köti össze (3.33b ábra).

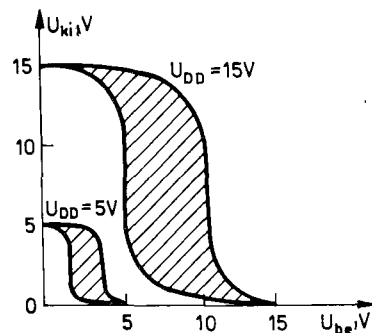
A másik vezérlési állapotban, amikor a bemenet az U_+ -on van, a helyzet fordított; az alsó, N-csatornás tranzisztor nyit ki, a felső, P-csatornás zár le, mert az utóbbinak azonos a GATE és a SOURCE feszültsége, azaz számára $U_{GS} = 0$ V. Az alsó, bekapcsolt kapcsoló a 0 V-ot vezeti a kimenetre (3.33c ábra).

Ezek szerint mindkét vezérlési állapotban közvetlenül egy-egy nyitott tranzisztor (0,5...1 k Ω nagyságu) csatorna ellenállásán keresztül jut a kimenetre a 0 V-os feszültség, ill. a + tápfeszültség. Nincs tehát nagyértékű munkaellenállás, ami a terhelő kapacitás tápfeszültségre töltődését lassítaná. A mindkét állapotban kis kimeneti impedancia miatt a CMOS áramkör sokkal gyorsabb, mint a passzív munkaellenállású NMOS vagy a PMOS áramkörök.

Tekintve, hogy felváltva vagy az alsó, vagy a felső kapcsolót kapcsoljuk be, a másik pedig mindig kikapcsolt állapotban van, a tápfeszültség és a 0 V között mindig szakadás van, ezért az áramfogyasztás statikus üzemben elvileg zérus, gyakorlatilag a lezárt tranzisztor egészen kis (max nA) maradék-áramával egyenlő. A felvett teljesítmény nW (nanowatt!) nagyságrendű egy CMOS inverterre. Dinamikus üzemben 0-ból 1-be, vagy 1-ből 0-ba váltáskor nagyobb a fogyasztás, egyrészt azért, mert az átkapcsoláskor egy nagyon rövid időre mindkét tranzisztor kinyit és rövidre zárja a tápforrást (a csatornaellenállásokon keresztül), másrészt azért, mert a terhelő kapacitást az egyik feszültségről át kell tölteni a másik logikai feszültség szintre, ehhez pedig áram kell, és ez az áram az éppen kinyitott tranzisztor csatornaellenállásán folyik át, amin disszipációt okoz. Ha egy CMOS áramkört MHz-es ütemben kapcsolgatunk, akkor teljesítmény fogyasztása akkora is lehet, mint a bipoláris áramköröké! A kis fogyasztás előnye tehát csak egyenfeszültségen és egészen kis kapcsolási frekvenciákon jelentkezik.

A CMOS integrált áramkörök gyártásakor törekednek a szimmetriára, ennek köszönhetően az inverterek (kapuk) $U_{be} - U_{ki}$ transzfer karakterisztikája közel van az ideálishoz (típusos karakterisztikákat kétféle tápfeszültségre a 3.34. ábrán lát-

hatunk), a "billenési" szint a tápfeszültség felének környezetében van, ezért a CMOS áramkörök kevésbé érzékenyek a logikai jelre szuperponálódott zavarokra.



3.34. ábra

Az előnyök mellett hátrány, hogy a kétféle (N-csatornás, és P-csatornás) MOS tranzisztort egyetlen integrált áramkör lapkán nehezebben lehet előállítani, bonyolultabb technológia szükséges. Ezért a CMOS áramkörök drágábbak, mint a "normál" NMOS, PMOS és bipoláris áramkörök.

3.5. Bipoláris áramkör családok

A továbbiakban a ma leggyakrabban használatos áramkör családok működésével, jellemzőivel, felhasználási területeivel foglalkozunk. Az, hogy többféle áramkör létezik és használatban van a legkülönbözőbb készülékekben, berendezésekben, azt bizonyítja, hogy mindegyiknek megvan a saját felhasználási területe, ahol legjobban megfelel a kívánalmaknak, működési feltételeknek. Nem mondhatjuk tehát, hogy egy adott áramkör család használhatatlan és egy másik a kizárólagosan megfelelő. Csak az áramkörök működésének ismeretében, az elvárható jellemzők tudatában dönthetünk alkalmazásukról, ezért kell tanulmányoznunk a szóbanjehető típuscsaládokat. Először tekintsük át, hogy amikor valamely áramkört minősítünk, milyen paramétereket kell elsősorban megvizsgálnunk! (Részletes adatlapot az áramkör-típusok ismertetésénél közlünk.)

3.5.1. A digitális áramkörök legfontosabb jellemzői

Logikai szintek

Alapvető kérdés, hogy az adott rendszer logikai 0 és logikai 1 jelszintje milyen. Az előzőkben említettük, hogy manapság elsősorban pozitív szintű logikákat használnak, ahol a logikai 1 szint "magasabb", pozitívabb a logikai 0 szintnél (ami nem zárja ki a földhöz képest negatív szint használatát, mint pl. az ECL-nél). Sokszor az igazságtáblázatban, ábrákon - a félreértések elkerülése végett - nem 0 és 1 jelöléseket használnak, hanem

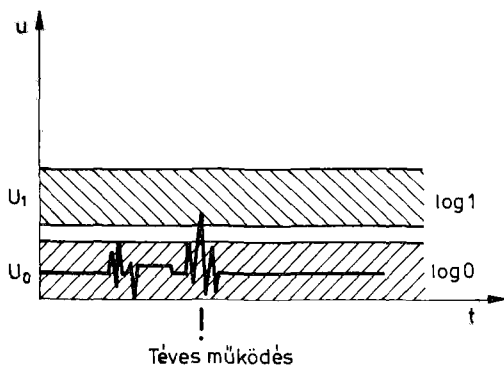
H - High: "magas" - (logikai 1) és
L - Low: "alacsony" - (logikai 0)

jeleket. A jelszintek nagyságát a katalógusok mindig valamilyen türelessel adják meg, feszültség tartományokat közölnek 0-ban és 1-ben, amelyen belül kell, hogy maradjanak a szintek szélsőséges, megengedett terhelés változások, tápfeszültség változás stb. hatására. Szintén alapvetően fontos, hogy a megadott jellemzők mindig a legrosszabb esetre (worst case) vonatkoznak! A digitális áramköröknek ui. a legszélsőségesebb esetekben is illeszkedniük kell egymáshoz, vagyis egy adott típuscsalád bármely áramkörének olyan kimeneti szinteket kell kiadnia, amely a legrosszabb körülmények között (a kimenetekre kapcsolt legnagyobb terhelés mellett, a tápfeszültség megadott szélsőértékek közötti ingadozása mellett, a teljes megadott működési hőmérséklet tartományban stb.) bizonyos tartalékkal alkalmas egy azonos típuscsaládbeli áramkör bemenetének meghajtására. (A digitális berendezések túl bonyolultak ahhoz, hogy részegységeinek illeszkedését utólagos beméréssel és beállítással biztosítsuk!).

Zajtartalék (Noise margin)

Nem elegendő, ha egy áramkör kimenete az őt követő áramkör bemenetével összeköthető, azzal illeszkedik, fontos, hogy ez megfelelő "biztonsági tartalékkal" történjen. A logikai jeleket vivő vezetésekre ui. kapacitív, induktív, elektromág-

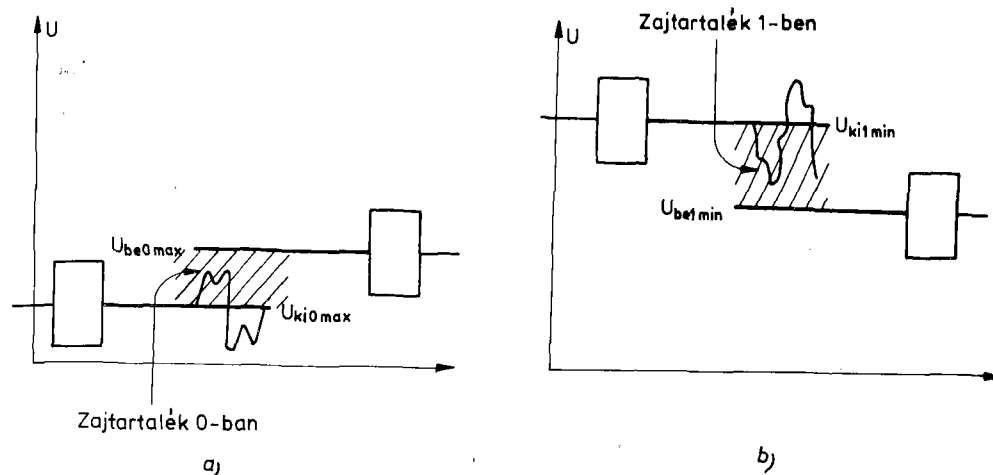
neses stb. uton a szomszédos áramkörökből, a környezetből vil-
 lamos zavarok juthatnak rá, ezek a (az általában zajnak neve-
 zett) feszültségek hozzáadódnak (pozitív-negatív előjellel) a
 logikai szintekhez. Belátható, hogy minél kisebb a logikai
 szintek türése és a 0 és 1 feszültség "távolsága", annál in-
 kább veszélyeztetik a működést a jelre szuperponálódott zajok,
 annál gyakrabban fordulhat elő, hogy a logikai jel és a zaj
 összege pillanatnyilag az ellenkező logikai szint tartományá-
 ba "csap át", téves működést okozva (3.35. ábra). Hiába a jól
 megtervezett, laboratóriumban jól működő áramkör, ha a tény-
 leges üzemi körülmények között mégsem használható, még a gon-
 dos árnyékolás, átgondoltan megtervezett földelés, tápellátás
 stb. ellenére sem, mert esetleg az alkatrészek a mindenképpen
 a rendszerbe kerülő zajt "nem türik el"! Ennek a jelentőségét
 sohasem szabad szem elől tévesztenünk!



3.35. ábra

A zajtartalék - nevéből következően - azt a jeltarto-
 mányt (feszültség-tartományt) jelenti, amelyen belül a logi-
 kai szint a rászuperponálódott zavar jellel együtt "ingadoz-
 hat" anélkül, hogy téves működés jönne létre. A katalógusok
 általában megadják, hogy az adott áramkör legrosszabb esetben
 a kimenetén mekkora logikai szinteket ad ki szavatoltan és mek-
 kora az a szint, amit egy következő ugyanilyen áramkör a be-
 menetén még tévesztés nélkül fogadni tud. Logikai 0-ban (po-
 zitív szintű logika esetén) megadják azt a legnagyobb feszült-

séget, amelyre az adott áramkör kimeneti feszültsége legrossz-
 szabb esetben "felmehet" (3.36a ábra, U_{ki0max}). Ugyanakkor köz-
 lik azt a legnagyobb még megengedett feszültséget, amelyet
 ugyanez a fajta áramkör a bemenetén még logikai 0-nak tekint
 (U_{be0max}). Ez utóbbi természetesen nagyobb feszültség és ameny-
 nyivel nagyobb, éppen annyi a "biztonsági távolság", azaz a
 zajtartalék vagy zajtávolság logikai 0-ban, ekkora lehet a jel-
 re "ülő" zajfeszültség pozitív csúcserőértéke.



3.36. ábra

Logikai 1-ben a meghajtó áramkör szolgáltat feltétlenül na-
 gyobb feszültséget, mint amit egy másik ugyanolyan áramkör a
 bemenetén minimálisan igényel (3.36b ábra, U_{ki1min} és U_{be1min}).
 A kettő különbsége adja a zajtartalékot logikai 1-ben, legfel-
 jebb ekkora lehet a zajfeszültség negatív csúcserőértéke. A rend-
 szerek általában szimmetrikusak, logikai 0-ban és 1-ben azo-
 nos a tartalék, és akkor a két (logikai 0-ban és 1-ben megadott)
zajtartalék összege a megengedett legnagyobb zavarfeszültség
csúcstól-csúcsig vett értékével egyenlő (lásd a 3.36. ábrát).
 A katalógusok általában ezt adják meg "noise margin" címszó
 alatt (pl. TTL: 1 V, ami a 2-szer 0,4 V-ból adódik némi jóin-
 dulattal). Mindez egyenfeszültségre (DC) és kisfrekvenciára
 igaz, vagyis az így megadott jellemző a DC zajtartalék. Na-

gyobb frekvenciás (több MHz-es) zavarokra az áramkörök általában kevésbé érzékenyek (hiszen működési sebességük korlátozottsága miatt "kevésbé veszik észre" a gyors ingadozásokat), ezért legtöbbször megadják az un. AC zajtartalékot is adott frekvencián vagy a frekvencia függvényében.

Teljesítmény disszipáció (felvett tápteljesítmény P_D)

Alapvető és a felhasználási területet nagymértékben meghatározó statikus (egyenáramu) jellemző. Minden áramkör fajta esetében nyilvánvaló cél, hogy az áramfogyasztás minél kisebb legyen, egyrészt azért, mert kisebb teljesítményigény esetén olcsóbb a tápegység, ill. lehetővé válik a telepről, akkumulátorról való táplálás, másrészt azért, mert a kisebb teljesítményigényű áramkörökből több integrálható össze egyetlen lemezken a túlmelegedés veszélye nélkül (ma a nagymértékben integrált LSI áramkörök bonyolultságát sokszor nem annyira a technikai lehetőségek, mint inkább a nagyon sok áramkört magában foglaló IC melegedése korlátozza). Már az inverterek összehasonlításakor láttuk, hogy a bipoláris, a MOS és a CMOS áramkörök fogyasztása nagyságrendekkel eltér. Azt is észre kellett vennünk, hogy a teljesítmény disszipáció szoros összefüggésben van a legfontosabb dinamikus jellemzővel, a működési sebességgel. A működési sebességet jellemző idő adat a

Jelterjedési késleltetési idő (propagation delay)

Definícióját a 3.23. ábra jelöléseivel már ismerjük: a bemeneti vezérléstől a kimeneti, logikai 0-ba ugrásig eltelt késleltetési idő (t_{pd0}) és a bemeneti (ellenkező) vezérléstől a kimeneti logikai 1-be ugrásig eltelt késleltetési idő (t_{pd1}) átlagértéke (t_{pd}). (Vigyázat! A jelterjedési késleltetési idő nem tévesztendő össze a kimeneti jel fel- vagy lefutási meredekségével!) Általában azt mondhatjuk, hogy nagy működési sebességet (kis t_{pd} időt) nagyobb teljesítmény "befektetésével" érhetünk el. A bipoláris áramkörök általában gyorsabbak, viszont fogyasztásuk nagy (sőt annál nagyobb, minél nagyobb a sebességük - gondoljunk arra, hogy a gyorsabb működéshez nagyobb határfrekvenciához az áramkörökben kisebb ellenállások

kelljenek, ami növeli a fogyasztást. Ehhez járul az is, hogy a bipoláris tranzisztorok határfrekvenciája nagyobb kollektoráramnál magasabb). A MOS áramkörök kevesebb teljesítményt fogyasztanak, viszont a nagy ellenállások miatt lassabbak is stb. - mindezeket a tulajdonságokat az áramkör típusok működésének áttekintése után összehasonlítjuk majd. Az látszik az eddigiekből is, hogy egy adott alkalmazásban mindig kompromisszumot kell kötnünk a sebesség és disszipáció tekintetében. Vannak helyek, ahol a sebesség elsőrendű fontossága ("nagy" számítógépek aritmetikai egységei, gyors jeleket digitalizáló, jelfeldolgozó berendezések stb.), van ahol az a legfontosabb, hogy a teljesítmény felvétel a lehető legkisebb legyen, a sebesség igény nem nagy (telepes, hordozható műszerek, digitális órák, kalkulátorok stb.), de a legtöbb helyen ("ipari elektronika": az iparban, vezérlésekben, szabályozásokban levő elektronika, mikroszámítógépek és "környezetük", "hétköznapi" elektronika stb.) az előbb említett kompromisszum értelmében nagyon alaposan meg kell gondolnunk, hogy hol, milyen áramkörcsaládot alkalmazunk, esetleg egy rendszerben vegyesen is. Érthető tehát, hogy nem lehet valamely áramkorról azt mondani, hogy "jobb mint a másik". A műszaki fejlesztés természetesen arra irányul, az integrált áramköröket előállító gyárak arra törekszenek, hogy minden áramkör típusból olyan újabb változatokhoz hozzanak ki, amelyek a lehető legkisebb jelterjedési késleltetés mellett a lehető legkisebb teljesítményigénnyel működnek. Ilyen értelemben az áramkörök "jóságára" jellemző mennyiség a jelterjedési idő és a felvett tápteljesítmény szorzata:

$$t_{pd} \cdot P_D$$

Ennek kell a lehető legkisebbnek lennie. A szorzat energia dimenziója és durva becslésként arra szolgál mértékül, hogy az illető áramkör minimálisan mennyi munka árán hajt végre egy logikai műveletet.

A kimenet "meghajtó képessége"

A fogyasztással összefüggő jellemző, logikai 0-ban és 1-ben is megadják. Ettől függ, hogy egyrészt hány saját családjához tartozó áramkör bemenetét képes meghajtani egy kimenet

(ez a FAN OUT szám), másrészt ebből tudhatjuk meg, hogy egy másfajta áramkörhöz való illesztéskor (ha egyáltalán lehetséges) hány áramkörrel terhelhetjük a kimenetet.

- Egyéb adatok, adottságok, amelyek szintén fontosak, és esetenként döntők lehetnek egy áramkör fajta kiválasztásakor, ilyenek pl.:

a tipusválaszték (van-e egyáltalán olyan IC, amelyet az adott funkcióra keresünk; alkatrészekből történő áramkör építéshez a TTL szolgál leggazdagabb választékkal),

az ár (összefügg a teljesítmény felvétellel és a típusválasztékkal, lehet, hogy adott feladatra olcsóbb IC-t találunk, de drágább hozzá a tápegység, az is lehet, hogy adott feladathoz nincs a típusválasztékban "kész" áramkör, de mégis olcsóbb több, saját családbeli IC-ből összeállítani),

a működési hőmérséklettartomány (legtöbb családban létezik ipari kivitelű sorozat, amelyre általában 0° és $+70^{\circ}\text{C}$ közötti működésre szavatolják valamennyi katalógus-jellemző teljesítését, a katonai változatok legtöbbször -55° és $+125^{\circ}\text{C}$ között működnek a katalógus által megadott határokon belül. (Ilyen széles hőmérséklet-tartományban még a "félvezető elmélet" sok tétele sem igaz!).

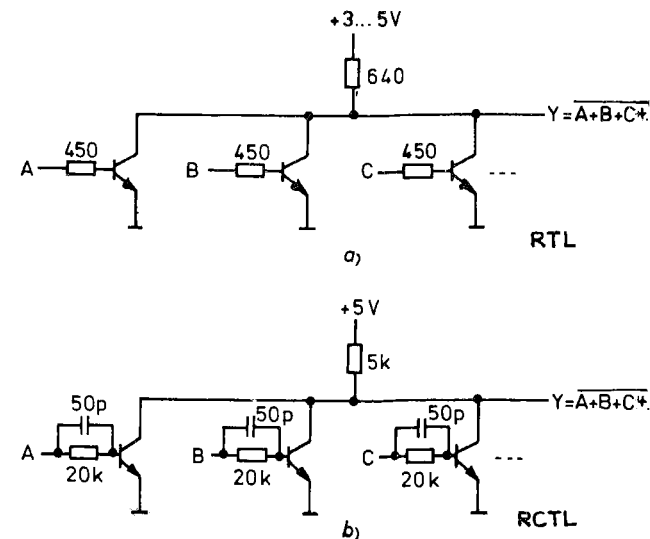
3.5.2. Régebbi bipoláris áramkörök

A teljesség kedvéért és a mai áramkörök működésének jobb megértése érdekében érdemes egészen röviden áttekinteni a "történelmi" előzményeket, nyomon követni a digitális integrált áramkör-típusok kialakulását.

RTL, RCTL (Resistor Transistor [Capacitor] Logic = ellenállás - tranzisztor-[kondenzátor]logika)

Az első IC-k még "diszkrét elemes szemlélettel" készültek: minél kevesebb alkatrész, annál jobb. Nem tettek mást, mint egyszerű invertereket kapcsoltak párhuzamosan (3.37. ábra). Akármelyik inverter bázisra (A, B, C-re) pozitív, logikai 1-et adva, a hozzá tartozó tranzisztor kinyit és a közös

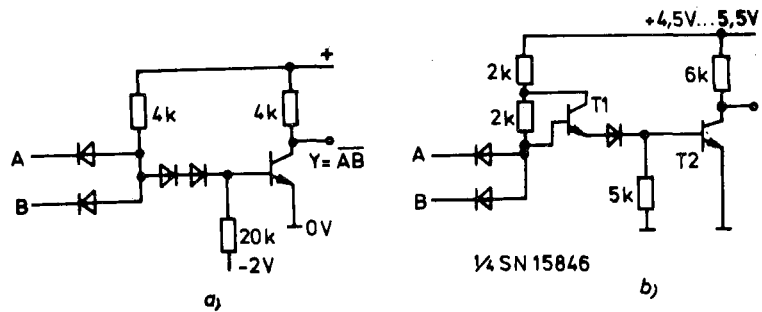
kimenet 0 V közelébe kerül (kimenet 0 lesz, ha bármelyik bemenet 1), így a kapcsolás a NOR univerzális műveletet valósítja meg. A kellő sebesség (mikrosec) eléréséhez kis ellenállásértékek voltak szükségesek, ami a disszipációt növelte meglehetősen nagyra, a zajtartalék viszont kicsi volt. A gyorsító kondenzátorral kiegészített kapcsolás (RCTL) annyiban jelentett javulást, hogy az ellenállások nagyobbak lehettek, a disszipáció csökkenthető volt (3.37b ábra).



3.37. ábra

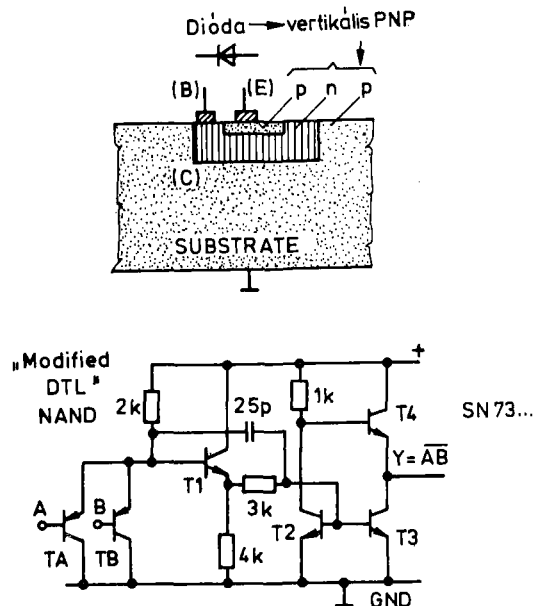
DTL (Diode-Transistor-Logic = dióda-tranzisztor logika)

Eleinte diódás ÉS kaput egészítettek ki egy inverterrel, így a NAND univerzális műveletet valósították meg. Az inverter tranzisztor biztos lezárhatósága érdekében negatív (bázis-lezáró) tápfeszültséget is használtak (3.38a ábra). Ezután fokozatosan szakitottak a diszkrét elemes "takarékos" szemlélettel (nem kizárólag a feladatot teljesítő alkatrészeket integrálták le, hanem a jobb működést elősegítő többletet is), így először a negatív tápfeszültség igényt küszöbölték ki az egyik szinteltoló dióda tranzisztorral (T_1) való helyettesítésével (3.38b ábra), majd olyan módosított áramkört készítet-



3.38. ábra

tek ("modified DTL"), amelyben már kihasználták az integrálás folyamán "magától" létrejövő vertikális PNP tranzisztorokat, amelyekkel a diódákat helyettesítették (3.39. ábra, ez volt az első példa az olyan alkatrészeire, amely csak integrálva képzelhető el), a kimenetre passzív munkaellenállásos inverter-végfokozat helyett aktív, "felhúzó" emitterkövetős ún. "totem-pole" ellenütemű végfokozatot helyeztek. Mindezzel lényegesen megnövelték a sebességet, a zajtartalékot, a "fan-out"-ot.



3.39. ábra

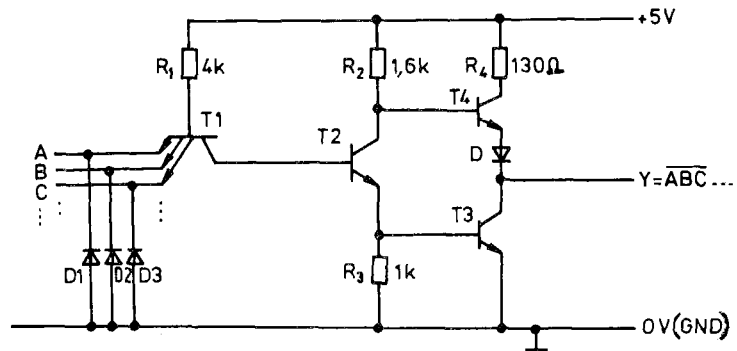
A 3.39. ábra DTL-NAND kapuja volt a ma is használatos TTL "elődjének", a már majdnem hasonló paramétereket (kb. 10 ns késleltetés, $n \cdot 10$ mW teljesítmény felvétel kapunként) nyújtó DTL rendszereknek az alapáramköre (TEXAS SN 53..., SN 73... sorozat). DTL áramkörök még itt-ott megtalálhatók "régii" berendezésekben (számítógépekben).

3.5.3. TTL, T²L (Transistor-Transistor Logic = = tranzisztor-tranzisztor logika)

Ma is ez a legnagyobb típusválasztéku, univerzális célra készülő bipoláris integrált áramkör rendszer, ezért ezzel részletesebben foglalkozunk. Összesen 6 féle TTL változat van, amelyből kiválasztható a sebesség, disszipáció és ár szempontjából a leginkább megfelelő típus. Kezdetben a NORMÁL (STANDARD) TTL sorozat (SN 54/74...) terjedt el legjobban, az évek folyamán ennek bővült legjobban a típusválasztéka (több száz típusra). Ma már a kis teljesítmény fogyasztású, Schottky-védett (Low-power Schottky: 54LS/74LS...) változat a legnépszerűbb, új fejlesztéshez - ahol TTL-re van szükség - ez a leginkább ajánlható (vannak újabb TTL áramkörök, amelyek már eleve csak Schottky változatban jelentek meg). A TTL áramkörök közül a normál változat működését tanulmányozzuk részletesebben és a többi változatot ezzel hasonlítjuk majd össze. A működésben alapvető eltérések nincsenek, a paraméterek viszont erősen eltérnek.

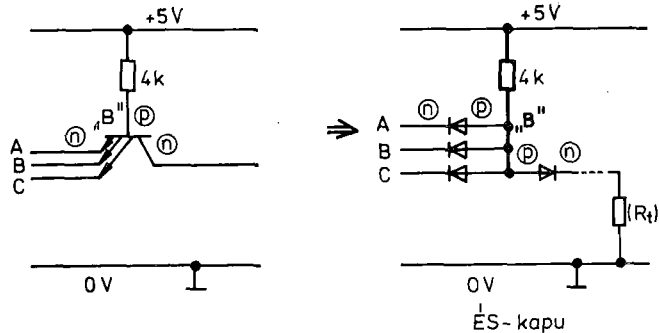
NAND alapáramkör

A normál TTL alapáramkörét, a NAND kaput mutatja a 3.40. ábra. Azonnal szembejön a bemeneten a több emitterű tranzisztor, amelyet a diszkrét elemek között hiába keresnénk, így integrálva viszont egészen természetes (a bázis rétegbe több kis emitter szigetet diffundálnak).



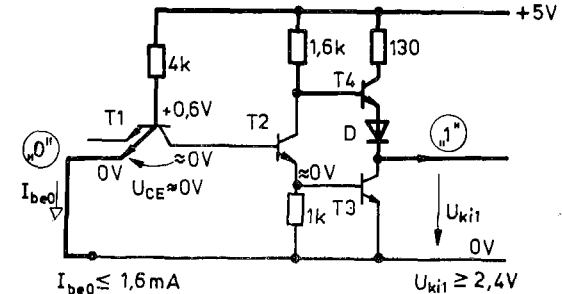
3.40. ábra

Ez a több emitterű tranzisztor tölti be az (aktív) ÉS kapu szerepét. Diódás helyettesítő képe, melyet a rétegsorrend alapján rajzolhatunk, a 3.41. ábrán látható. Ez a helyettesítő kép természetesen csak a logikai működés szempontjából ekvivalens, hiszen a tranzisztor nem két szembe fordított dióda, hanem aktív elem (és éppen ez a nagy előnye). Az ábrán lévő diódák egy pozitív szintű ÉS kaput alkotnak, ahhoz, hogy a diódás kapu kimenete pozitív feszültségű legyen, valamennyi bemenetnek pozitívnak kell lennie. Ha egyetlen bemenet 0 V-on van, akkor a "B" pont kb. +0,6 V-ra kerül, így a jobb oldali dióda katódján az ÉS kapu kimenetén gyakorlatilag 0 V van.



3.41. ábra

A TTL áramkörben a bemeneti ÉS kaput egy "fázishasító" fokozat követi (T2), amely a T3-as inverter tranzisztort és a T4 aktív felhúzó tranzisztort ellenütemben hajtja meg. A T3 és T4 tehát gyors működésű, a DTL-nél is meglévő TOTEM POLE kimeneti fokozatot alkot. Vizsgáljuk meg részletesebben a működést különböző (alapvetően kétféle) vezérlés esetére!



3.42. ábra

a) Ha a bemenetek közül valamelyik (vagy több) logikai 0-n, kb. 0 V-on van (3.42. ábra), akkor a földelt bemenethez tartozó E-B dióda kinyit, hiszen T1 bázisa a 4 kΩ-on keresztül pozitív, nyitó előfeszítést kap, emittere pedig 0 V-on van. A kinyitás meglehetősen nagy bázisárammal történik, így a T1 telítésbe megy, vagyis CE feszültsége gyakorlatilag 0 V lesz, tehát T2 bázisán is 0 V körüli a feszültség. Emiatt T2 és T3 lezár (ahhoz, hogy kinyissanak, T2 bázispontján összesen $2 \cdot 0,6 = 1,2$ V-nak kellene lennie, mert T2 és T3 bázis-emitter diódája sorba kapcsolódik). Mivel T2 szakadásként viselkedik (olyan mintha benne sem lenne az áramkörben), T4 a pozitív tápfeszültségből az 1,6 kΩ-os ellenálláson keresztül bázisáramot kap és mint egy emitterkövető, a kimenet feszültségét a kinyitó D diódán keresztül pozitívba huzza. A kimeneti áramot a 130Ω-os ellenállás korlátozza. A kimeneti logikai 1 feszültséget üresjárásban úgy kapjuk, hogy a tápfeszültségből levonjuk a nyitott T4 B-E feszültségét és a nyitott D dióda feszültségét, így:

$$U_{kil} = 5V - 0,6V - 0,6V = 3,8V \quad (\text{üresjárásban}).$$

Ha a kimenetre passzív (pozitív) terhelést kapcsolunk, akkor ez a feszültség csökken (a terhelő emitteráram hatására folyó bázisáram az $1,6 \text{ k}\Omega$ -on feszültséget hoz létre). A katalógus szerint maximális ($0,4 \text{ mA}$) terhelés esetén a kimeneti feszültség:

$$U_{\text{kil}} \approx 2,4 \text{ V.}$$

A másik legfontosabb statikus adat ebben a vezérlési helyzetben a bemeneti logikai 0-hoz tartozó áram. T1 nyitott BE diódáján keresztül folyik a $4 \text{ k}\Omega$ -os ellenállás teljes árama és mivel a kollektoráram zérus, az összes áram a bemenetet 0 V-ra vivő vezetéken folyik. Mivel T1 bázisa kb. $0,6 \text{ V}$ van, a $4 \text{ k}\Omega$ -os ellenállás árama (mely, ha csak egyetlen bemenet van 0 V-on, megegyezik a bemeneti árammal):

$$I_{\text{be0}} = \frac{5 \text{ V} - 0,6 \text{ V}}{4 \text{ k}} = 1,1 \text{ mA.}$$

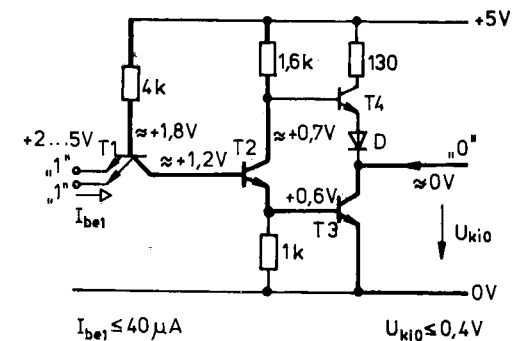
A katalógus szerint ennek értéke legrosszabb esetben:

$$I_{\text{be0}} = 1,6 \text{ mA.}$$

Az I_{be0} a meghajtó áramkör számára nem "szokásos" fogyasztó jellegű terhelést jelentő áram, hanem negatív irányú, a bemenet feszültségét pozitív irányba húzó ún. húzóáram (sink current) - egyezésben azzal, amit az ÉS kapuk bemeneti áramáról tanultunk (lásd a 3.2. fejezetet!). A TTL-t meghajtó áramkörnek, generátornak tehát olyannak kell lennie, amely ezt a pozitív irányba "törekvő" bemenetet 0 V közelében képes tartani (kis belső ellenállással),

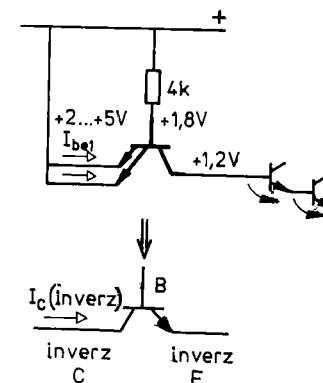
b) Ha mindegyik bemenet logikai 1-es vezérlést kap, a feszültsége $+2,4 \text{ V} \dots +5 \text{ V}$ közötti (3.43. ábra), akkor T1 E-B diódái lezárnak, a $4 \text{ k}\Omega$ -os ellenállás teljes árama most a T1 nyitott C-B átmenetén keresztül T2 bázisába folyik. A nagy bázisáram következtében T2 kinyit, telítésbe megy (kollektoráramát az $1,6 \text{ k}\Omega$ korlátozza) és kinyitja T3-as inverter tranzisztort, amelynek kollektorfeszültsége 0 V körüli lesz. Ha tehát mindegyik bemenet logikai 1-ben van, akkor a kimenet logikai 0-ba kerül, vagyis az áramkör a NAND kapcsolatot való-

sitja meg. A T1 tranzisztor ilyenkor INVERZ üzemmódban van, ami azt jelenti, hogy emitterei és kollektora "szerepet cseréltek." A T1 kollektora kb. $+1,2 \text{ V}$ -on van, a két sorbakapcsolt (T2 és T3) nyitóirányú bázis-emitter feszültségén, a T1 bázisa a pozitívra menő $4 \text{ k}\Omega$ miatt $0,6 \text{ V}$ -tal pozitívabb a kollektornál, feszültsége kb. $+1,8 \text{ V}$. Az emitterek potenciálja a legpozitívabb: $+2 \dots 5 \text{ V}$. Most tehát az emitterek vannak kollektorként, a kollektor van emitterként előfeszítve, ami az inverz üzemmódra jellemző (3.44. ábra).



3.43. ábra

Azt várnánk, hogy ekkor a befolyó "kollektoráram", vagyis a logikai 1-hez tartozó bemenő vezérlőáram (I_{bel}) nagyértékű lesz. A valóságban erre a katalógus max $40 \mu\text{A}$ -t ad meg ($U_{\text{be}} = 2,4 \text{ V}$ -nál).



3.44. ábra

A kis inverz kollektoráramra a T1 geometriai felépítése ad magyarázatot (a bázisba diffundáltatott emitterek viszonylag kis területűek, így a nagy felületű kollektorból emittálódott töltéshordozók a bázisban szétszóródva rekombinálódhatnak, csak keveset tudnak az emitterek - inverz kollektorok - összegyűjteni).

A végfokozatban a D dióda feladata, hogy megakadályozza ebben a vezérlési helyzetben a felső emitterkövető T4 kinyitását. T2 telítésben van, kollektorán a feszültség 0,1...0,2 V-tal pozitívabb, mint emitterén, ahol +0,6 V van, tehát T4 bázisa kb. +0,7...0,8 V feszültségen van. Emittere - ha nem lenne dióda - a kimenet 0,1...0,2 V feszültségén lenne, tehát B-E feszültsége elegendő lenne, ahhoz, hogy kinyisson. Ha T3 és T4 egyszerre nyitva lenne, akkor a tápfeszültség rövidre záródna (csak a 130 Ω korlátozná az áramot). A dióda "megemeli" T4 emitterét, így ahhoz, hogy nyitott T3 mellett T4 is kinyisson, a bázisán legalább 2 dióda-feszültségnek (D nyitófeszültsége + T3 B-E feszültség) kellene lennie, de ott csak +0,7...0,8 V van, ezért T4 biztosan lezárva marad.

A legfontosabb adatok a normál TTL-re, ha mindegyik bemenetet logikai 1-el vezéreljük:

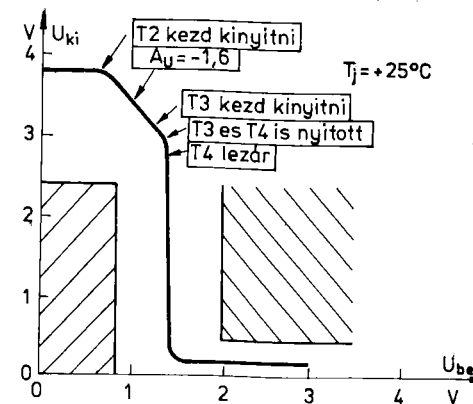
a bemeneti áram:

$$I_{bel} \leq 40 \mu A \quad (U_{bel} = 2,4 \text{ V-nál})$$

és a kimeneti logikai 0 feszültség:

$$U_{ki0} \leq 0,4 \text{ V} \quad (\text{maximális terhelésnél}).$$

c) Végül vizsgáljuk meg a normál NAND kapu működését a bemeneti vezérlés logikai 0-ról 1-re váltása közben! Egy bemenetet kivéve mindegyik bemenet logikai 1-en legyen és ennek az egynek a feszültségét változtassuk 0 V-ról fokozatosan pozitív irányba a logikai 1 szintig! A viszonyokat az $U_{be} - U_{ki}$ transzfer karakterisztika (3.45. ábra) szemlélteti.



3.45. ábra

Ha a bemenet feszültségével elérjük a +0,6...0,7 V-ot, akkor az eddig zárt T2 fokozatosan kinyit, mert bázisán is ugyanannyi a feszültség, hiszen tudjuk, hogy T1 telített (a + tápfeszültségre menő 4 kΩ erősen tulvezérli) és kollektorán gyakorlatilag ugyanannyi a feszültség, mint az emitterén. T2 meginduló kollektorárama feszültséget hoz létre az 1,6 kΩ-on, így az emitterkövető T4 bázisfeszültsége és emitterfeszültsége, vagyis a kimeneti feszültség is csökkenni kezd. Az erősítés ekkor kb. -1,6 (T2 emitterében 1 kΩ, kollektorában 1,6 kΩ és ekkor még lineáris tartományban van). T3 még zárt (1 kΩ miatt). További bemeneti feszültség növelésre +1,2V...+1,3 V-nál - mivel T2 bázisán is ugyanekkora feszültség van - kezd kinyitni a T3 tranzisztor, ezért a kimeneti feszültség rohamosan csökken, az erősítés nagy. A T3 kinyitásának pillanatában még a felső T4 is nyitva van és a két soros tranzisztor a tápfeszültséget szinte rövidre zárja (130 kΩ korlátozza csak az áramot)! Minden átkapcsoláskor a TTL áramkörökben emiatt egy rövid ideig (néhány ns) tartó tápáram-tranziens keletkezik (30-50 mA csucsárammal). Akármilyen lassan változtatjuk a bemeneti feszültséget, a kimeneti feszültség ilyenkor gyorsan (sokszor "gerjedés"-szerű oszcilláció kíséretében) változik, az áramkörben fellépő pozitív visszacsatolt jellegű folyamatok miatt. A kimeneti feszültség végül a T3 maradékfeszültségére áll be.

Ha a bemeneti vezérlés gyors, akkor az átkapcsolási idő, a jelterjedési késleltetés csökkentése szempontjából előnyös, hogy a bemeneten az ÉS diódák helyett aktív tranzisztor van. A bemeneti jel 1-ről 0-ra csökkenésekor a telítésbe menő T1 tranzisztor hirtelen lehuzza T2 bázisát, ezzel sietteti az átkapcsolást. A bemenet 0-1 változásakor viszont a telített (rövidzárként viselkedő) T1 sietteti T2 kinyitását (t_{pd} 10 ns körüli). A TTL LS változatban a bemeneten nem aktív tranzisztor van, hanem gyors Schottky dióda sor, itt másképp oldják meg a gyors átkapcsolást (lásd ott!).

A 3.40. ábrán látható alapkapcsolásban a bemeneten levő ("normális" üzemben lezárt D1, D2, D3...) diódák az áramkör védelmére és a zavarvédelemre alkalmasak; logikai 0-ban "levágják" (rövidrezárják) a negatív, -0,6 V-nál nagyobb amplitudóú zavarjeleket, tranzienseket.

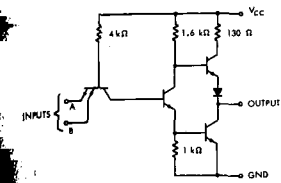
Egy digitális, TTL áramkör katalógus lapja

Az SN 5400, ill. 7400 típusu "legegyszerűbb" kapuáramkör részletes adatlapját a 3.46. ábrán láthatjuk (egy "régii" Texas Instruments, CC 201 katalógusból, mert ez meglehetősen részletes). Értelmezzük a közölt jellemzőket (amelyek természetesen nemcsak az 5400-7400 típusra érvényesek, hanem az egész "normál", "standard" TTL családra).

- A típusmegjelölés (SN 5400, 7400) és az áramkör megnevezése (QUADRUPLE 2-INPUT POSITIVE NAND GATES = négy db 2-bemenetű NAND kapu) alatt a kapcsolási rajz (schematic) és a bekötési rajz látható a kétféle (S FLAT PACKAGE = "lapos" tok, vagy miniatűr tok és a J vagy N DUAL-IN-LINE PACKAGE = "kétoldalt egyvonalban" kivezetett) tokozásra. A rajzok mindig felülnézetre (TOP VIEW) érvényesek! A kisméretű "S" és a "J" tokban levő áramkörök meglehetősen drágák, legolcsóbb a műanyag "N" tokozás, a legtöbb esetben ezt használjuk. A helyes pozicionálás érdekében minden tok fajtán van valamilyen jelölés, amelynek segítségével az 1-es láb helye meghatározható (az S token egy kis pont, a J és N token az 1-es láb felőli oldalon egy "bemarás").

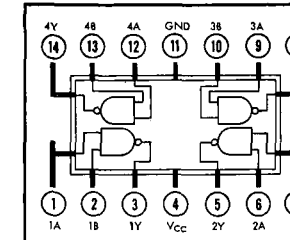
CIRCUIT TYPES SN5400, SN7400 QUADRUPLE 2-INPUT POSITIVE NAND GATES

schematic (each gate)

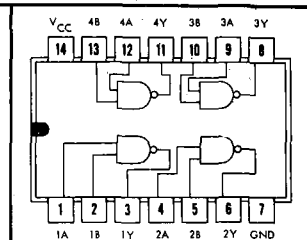


NOTE: Component values shown are nominal.

F FLAT PACKAGE
(TOP VIEW)



J OR N DUAL-IN-LINE PACKAGE
(TOP VIEW)



positive logic: $Y = \overline{AB}$

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply Voltage V_{CC} : SN5400 Circuits	4.5	5	5.5	V
SN7400 Circuits	4.75	5	5.25	V
Normalized Fan-Out From Each Output, N			10	
Operating Free-Air Temperature Range, T_A : SN5400 Circuits	-55	25	125	°C
SN7400 Circuits	0	25	70	°C

electrical characteristics (over recommended operating free-air temperature range unless otherwise noted)

PARAMETER	TEST FIGURE	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
$V_{in(1)}$ Logical 1 input voltage required at both input terminals to ensure logical 0 level at output	1	$V_{CC} = \text{MIN}$	2			V
$V_{in(0)}$ Logical 0 input voltage required at either input terminal to ensure logical 1 level at output	2	$V_{CC} = \text{MIN}$			0.8	V
$V_{out(1)}$ Logical 1 output voltage	2	$V_{CC} = \text{MIN}$, $V_{in} = 0.8 \text{ V}$, $I_{load} = -400 \mu\text{A}$	2.4	3.3		V
$V_{out(0)}$ Logical 0 output voltage	1	$V_{CC} = \text{MIN}$, $I_{sink} = 16 \text{ mA}$, $V_{in} = 2 \text{ V}$	0.22	0.4		V
$I_{in(0)}$ Logical 0 level input current (each input)	3	$V_{CC} = \text{MAX}$, $V_{in} = 0.4 \text{ V}$			-1.6	mA
$I_{in(1)}$ Logical 1 level input current (each input)	4	$V_{CC} = \text{MAX}$, $V_{in} = 2.4 \text{ V}$			40	μA
		$V_{CC} = \text{MAX}$, $V_{in} = 5.5 \text{ V}$			1	mA
I_{OS} Short-circuit output current§	5	$V_{CC} = \text{MAX}$	SN5400	-20	-55	mA
			SN7400	-18	-55	mA
$I_{CC(0)}$ Logical 0 level supply current	6	$V_{CC} = \text{MAX}$, $V_{in} = 5 \text{ V}$		12	22	mA
$I_{CC(1)}$ Logical 1 level supply current	6	$V_{CC} = \text{MAX}$, $V_{in} = 0$		4	8	mA

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$, $N = 10$

PARAMETER	TEST FIGURE	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{pd0} Propagation delay time to logical 0 level	65	$C_L = 15 \text{ pF}$, $R_L = 400 \Omega$		7	15	ns
t_{pd1} Propagation delay time to logical 1 level	65	$C_L = 15 \text{ pF}$, $R_L = 400 \Omega$		11	22	ns

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$.

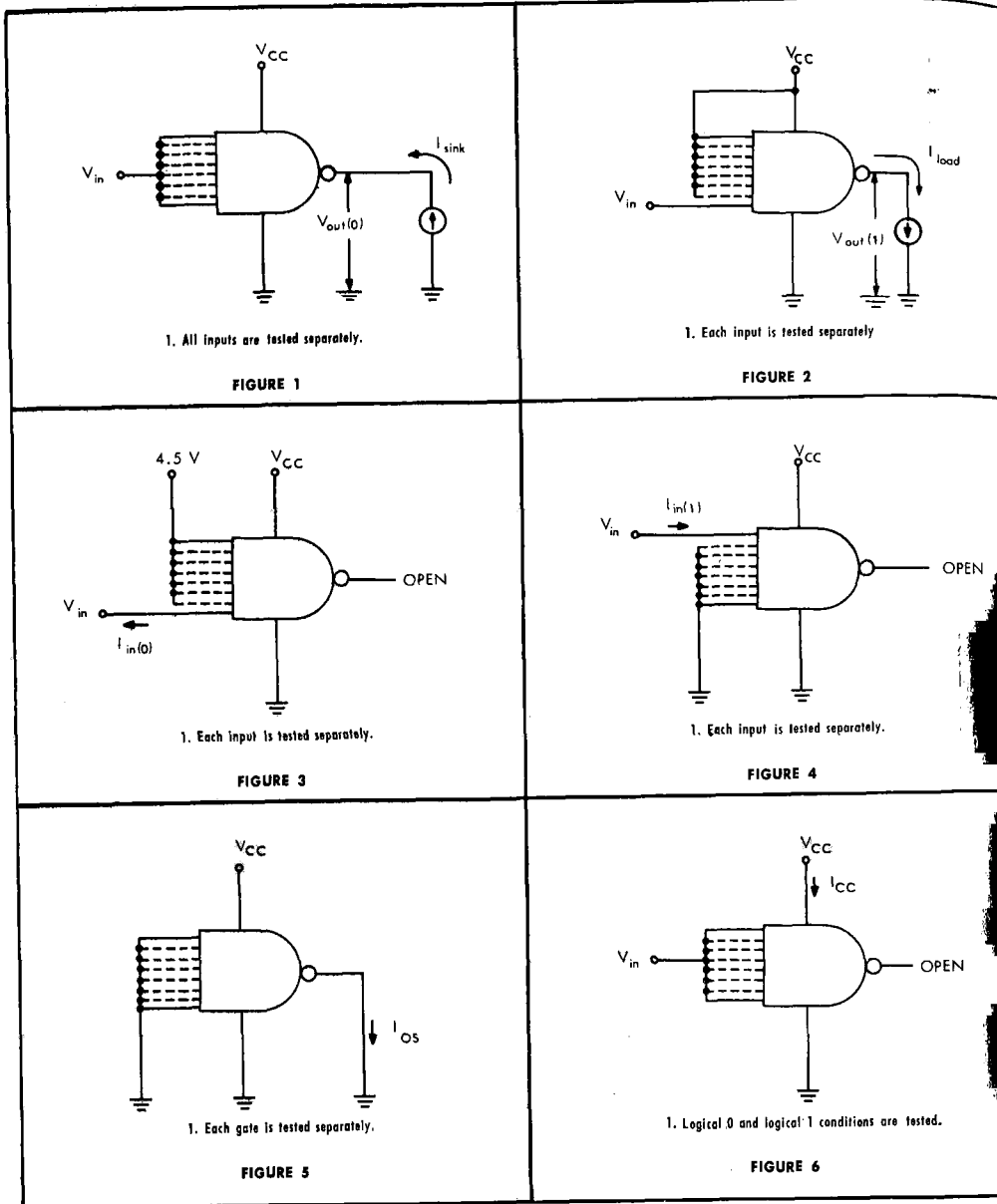
§ Not more than one output should be shorted at a time.

—SEE ORDERING INSTRUCTIONS PAGE 1-1—

3.46. ábra

PARAMETER MEASUREMENT INFORMATION

d-c test circuits§



§ Arrows indicate actual direction of current flow.

3.47. ábra

- A következő rész az ajánlott üzemi viszonyokat tartalmazza (recommended operating conditions), úgy, hogy megadja a megengedett minimális, a névleges és a maximális értékeket. A tápfeszültség (Supply Voltage), V_{CC} névlegesen 5 V minden TTL típusra, tűrése a "katonai" típusokra $(5400) \pm 10\%$, az "ipari" típusokra $(7400) \pm 5\%$ (emiat TTL-hez általában stabilizált tápegységet kell használnunk). A kimenet terhelhetőségét adja meg a következő sor (Normalized Fan-Out From Each Output), $N = 10$, ami azt jelenti, hogy egy kimenetre maximálisan 10 másik TTL áramkör bemenete kapcsolható, vagyis egy kimenet 10 újabb bemenetet hajthat meg. Az $N = 10$ -es FAN-OUT szám jellemző a legtöbb TTL kapu és MSI áramkörre. Arra azonban vigyáznunk kell, hogy eltérő sorozatu TTL áramkörök (lásd később) összekapcsolásakor nem érvényes az $N = 10$, a terhelhetőséget a katalógusban levő táblázatból kell megállapítanunk vagy az áramadatokból kell kiszámítanunk. Vannak nagyobb terhelhetőségű, ún. BUFFER elemek is ($N = 30$, pl.: 7440). A működési környezeti hőmérséklettartomány (Operating Free-Air Temperature Range):

az ipari típusokra (7400): $0 \dots +70 \text{ }^\circ\text{C}$,
a katonai típusokra (5400): $-55 \dots +125 \text{ }^\circ\text{C}$ (!).

- Az ezután következő táblázat a villamos jellemzőket tartalmazza, amelyek a teljes hőmérséklettartományra érvényesek (electrical characteristic over recommended operating free-air temperature, unless otherwise noted), ha csak nincs másképp megadva. A táblázat első oszlopában van a paraméter, amelyet megadnak, majd a paraméter mérési elrendezését mutató ábra száma következik (TEST FIGURE) - lásd a 3.47. ábrán lévő fényképet! A következő rovat közli, hogy az illető paraméter mérést milyen üzemi viszonyok között kell végezni, az ajánlott viszonyok közül melyik szélsőértéket kell választani. Ezeket mindig úgy adják meg, hogy a legrosszabb esetet (worst-case) valósítsák meg, hiszen - amint arról már szó volt - a digitális áramköröknek a legszélsőségesebb esetekben is egymáshoz illeszthetőeknek kell lenniük, meg kell felelniük a specifikációknak. A negyedik oszlop tartalmazza a paraméter mi-

nimális, tipikus és maximális értékét. Ahol csak egy adat van, az mindig a legrosszabb esetet tükrözi.

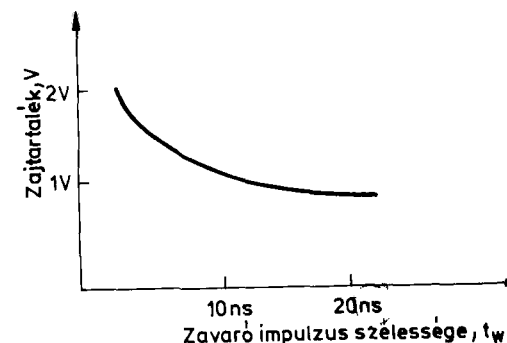
Az első sorban az a logikai 1 feszültség van megadva, amely valamennyi bemeneten szükséges ahhoz, hogy a kimenet logikai 0 feszültségen legyen ($V_{in(1)} = \min 2 \text{ V}$).

A második sor azt a logikai 0 bemeneti feszültséget adja meg, amely akármelyik bemeneten szükséges a kimeneti logikai 1 feszültség eléréséhez ($V_{in(0)} = \max 0,8 \text{ V}$). Mindkét adat a legrosszabb esetre, a tápfeszültség minimális értékére érvényes. A következő két sor az előző bemeneti szintekhez tartozó kimeneti logikai szinteket adja meg maximális terheléssel, vagyis 10 ujjabb bemenet terheléssel. Ha ezeket az adatokat ($V_{out(1)} = \min 2,4 \text{ V}$, $V_{out(0)} = \max 0,4 \text{ V}$) összevetjük a bemeneti feszültség adatokkal, akkor látszik, hogy a biztonságos összekapcsolhatóság érdekében mindkét logikai szinten $0,4 \text{ V}$ zajtartalék van, ami nem túlzottan nagy érték, a TTL a meglehetősen nagy zajérzékenységről nevezetes.

A továbbiakban az adatlap megadja a logikai 0-hoz tartozó bemeneti áramot, amely - mivel huzóáram - negatív előjelű: $-1,6 \text{ mA max}$. A logikai 1-hez tartozó bemeneti áram pozitív értelmű: $40 \mu\text{A max}$. Az áramok szempontjából legrosszabb eset, ha a tápfeszültség maximális, ilyenkor az áramok is a legnagyobbak. Megadják azonkívül a kimeneti rövidzárási áram maximális és minimális értékét, ha a kimenet logikai 1-ben van, amint ez a 3.47. ábrából is kiderül. Ez a kimenet és a föld közötti rövidzárási áram. A táblázat alatt megjegyzésként szerepel, hogy egyszerre egy kimenetnél többet nem szabad rövidre zárni! A logikai 0-ba vezérelt kimenet és a + tápfeszültség közötti rövidzárási áramot nem adják meg, ilyenkor csak a T3 végtranzisztor árama korlátoz, ezért a TTL áramkör tönkremegy, ha a kimenetét a + tápfeszültséghez zárjuk! Az utolsó két sorban a tápáram fogyasztást adják meg, ha a kimenet (terheletlenül) logikai 0-ban (typ: 12 mA), ill. logikai 1-ben van (typ: 4 mA). Ebből következik, hogy az átlagos teljesítmény felvétel 50% logikai 0 és 50% logikai 1 vezérlésnél $8 \text{ mA} \cdot 5 \text{ V} = 40 \text{ mW}$, ez egyetlen kapura 10 mW átlagos fogyasztást jelent.

- Az adatlap végül közli a dinamikus, kapcsolási jellemzőket (switching characteristic) $V_{cc} = 5 \text{ V}$ -nál $25 \text{ }^\circ\text{C}$ környezeti hőmérsékleten és $N = 10$ terhelésnél. Kapukra a jelter-

jedési késleltetési időket adják meg (propagation delay time), ha a kimenet 0-ba (t_{pd0}), ill. ha 1-be (t_{pd1}) megy. Értelmességüket a bipoláris inverter tárgyalásakor már láttuk (a 3.3 fejezetben) A TTL NAND kapu t_{pd0} ideje tipikusan 7 ns, t_{pd1} ideje tipikusan 11 ns, az átlag $t_{pd} = 9 \text{ ns}$, biztonsággal 10 ns. A sorrendi áramköröknek többféle késleltetési és vezérlési időadata van, ezekről ott lesz szó.



3.48. ábra

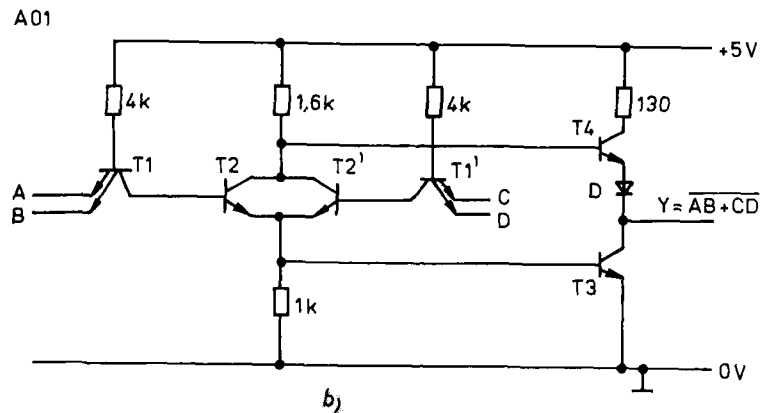
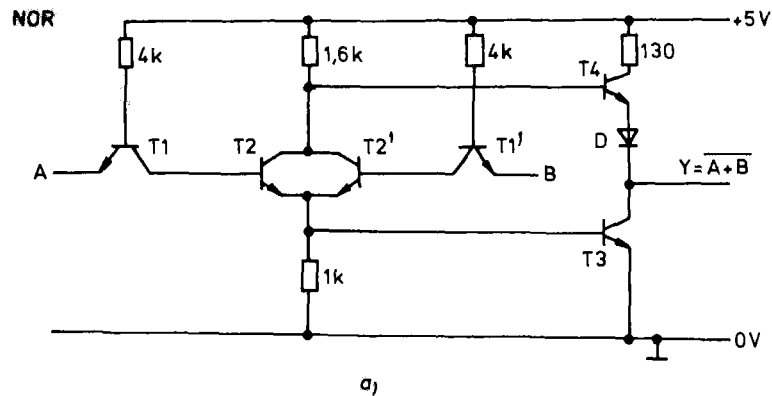
Az adatokat áttekintve megállapíthatjuk, hogy a "normál" TTL áramkör meglehetősen gyors működésű (10 ns), de teljesítmény fogyasztása meglehetősen nagy (10 mW kapu-áramkörönként), DC zajtartaléka ($\pm 0,4 \text{ V}$) kicsinek mondható (a dinamikus, AC zajtartalék nagyobb, a zavaró impulzus rövidülésével egyre növekvő, jellegét a 3.48. ábra diagramja mutatja). A zajok hatásának mérséklése szempontjából kedvező, hogy a TTL "totem-pole" kimenetnek kicsi a kimeneti impedanciája (logikai 1-ben az emitterkövetőnek legfeljebb 70Ω , logikai 0-ban a teljesítményes tranzisztorok legfeljebb 12Ω , így a logikai jelvezetékeket gyakorlatilag feszültséggenerátor hajtja meg, rövidrezárva az idegen jeleket.

A TTL Nem-Vagy, NOR és az ÉS-Nem-Vagy, AND-OR-INVERT (AOI) alapáramkör

A NOR kapu-változatban két külön TTL bemeneti fokozat van egy-egy emitterrel (3.49a ábra: T1 és T1' emittere az A és B jelbemenet), a hozzájuk tartozó fázishasító tranzisztorok (T2

és T2') párhuzamosan vannak kötve. Ehhez a szokásos módon csatlakozik a "totem-pole" végfokozat (T3, D, T4). Amikor vagy az A bemenet, vagy a B bemenet, esetleg mindkettő logikai 1-es (pozitív feszültségű) vezérlést kap, a hozzá kapcsolt fázishasító tranzisztor kinyit, aminek következtében kinyit az alsó végfokozat tranzisztor, a kimenet logikai 0 lesz:

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



3.49. ábra

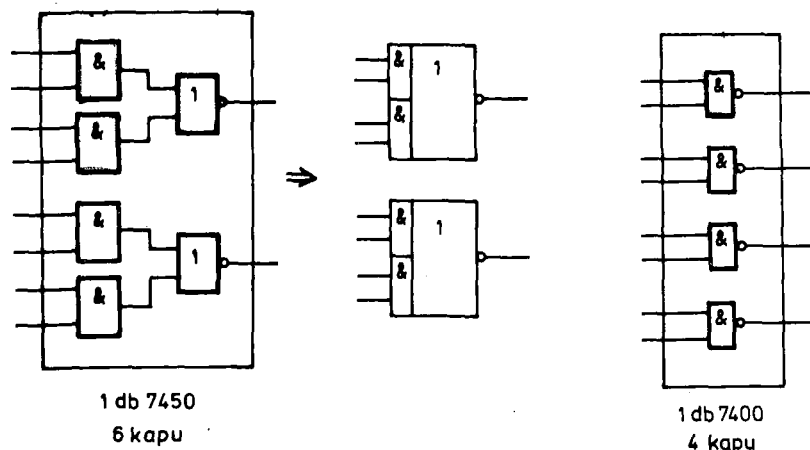
A TTL bemeneteken szokás szerint amúgy is több emitteres tranzisztor van, kézenfekvő tehát, hogy a NOR kapu bemeneteit is több emitterrel készítsék, így jön létre, az AND-OR-INVERT ka-

pu (a 3.49b ábrán 2-szer 2 bemenetű változat látható). Amikor az A és B bemenet 1-es, vagy a C és D bemenet 1-es (esetleg valamennyi bemenet 1-es), akkor T2 vagy T2' nyitása következtében a kimenet $Y = 0$ lesz (vagyis $Y = 0$ lesz, ha teljesül $A \cdot B$ vagy a $C \cdot D$ feltétel):

$$Y = \overline{AB + CD}$$

vagyis csupán "többlet emitterek" segítségével újabb logikai kapcsolat jött létre.

Az AND-OR-INVERT kapu nagy előnye (ahogy ezt már a 2.5. fejezetben taglaltuk), hogy kétfokozatu kapu hálózatot egyesít magában, de jelkésleltetési ideje csak egyszeres. Az AOI kapuval valamennyi BOOLE-függvény megvalósítható - ugyanugy mint az ÉS-VAGY és inverteres hálózatokkal - de kisebb késleltetéssel és kevesebb alkatrészigénnyel, a KARNAUGH-táblás tervezési módszert már a 2.5. fejezetből megtanulhattuk. Ahogy a kapcsolási rajzból is látszik, az AOI kaput egyetlen kapu gyanánt kell kezelni, az ÉS kimenetek nincsenek kivezetve, jelüket nem lehet külön felhasználni. Ez csak részben hátrány, az előny nagyobb: éppen a kevesebb kivezetéseknek köszönhetően egy "két szintes" kapu hálózatot egyszerűbb huzalozással, kevesebb forrasztással lehet felépíteni. Egy IC tokban több kapu fér el, hiszen pl. 2 bemenetű NAND-ből 4 db-ot tartalmaz egy 7400-as típus, viszont pl. a 7450-es típusban 2 készlet 2-szer kétbemenetű AOI kapu van, vagyis összesen 6 db 2 bemenetű kapu, tehát "nagyobb lehet az integráltság foka", ha kapu-hálózatokat AOI-ből építünk fel (3.50. ábra). A könnyebb felhasználhatóság érdekében bővíthető változatokat is készítenek, amelyekhez megfelelő ÉS bővítő áramkörökkel csatlakozva növelhetjük az ÉS bemenetek számát. Bővíthető pl. az említett 7450-es típus is, ilyenkor kivezetik a T2 és T2' közös kollektor pontját (\bar{X}) és emitterpontját (X), ezekhez csatlakoztatható a bővítő kapu, ugyanilyen jelölésű (\bar{X} és X) pontja. A bővítő (pl. 7460) nem önálló kapuáramkör, mert csak a négy emitterű T1, ezenkívül csak a T2 van beépítve, az \bar{X} és X jel nem TTL szintű, hanem kimondottan ilyenfajta "belső csatlakozásra" alkalmas, ezeket a lehető legrövidebb huzalozással

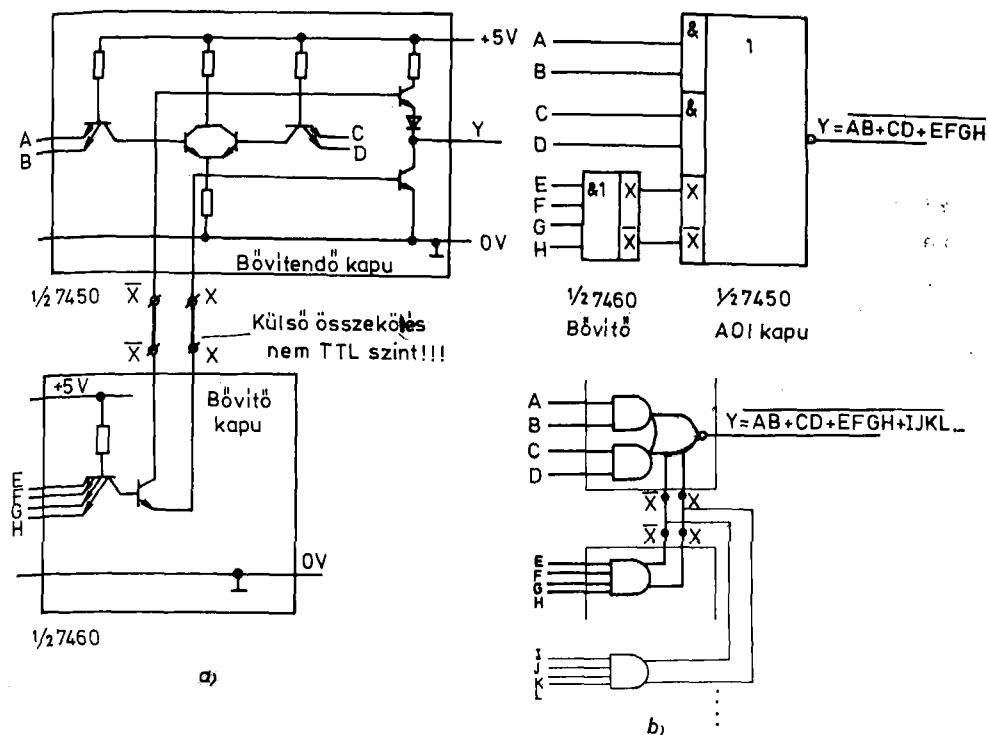


3.50. ábra

kell a bővítendő kapu megfelelő \bar{X} és X kivezetésével összekötni (a szórt kapacitások csökkentése és a zavarok kiküszöbölése érdekében - ne felejtsük el, hogy ilyenkor a TTL kapu "belső jelét" vezetjük el!) Az áramkört bővítővel együtt a 3.51a ábra, a kapcsolási rajzjeleket a 3.51b ábra mutatja (ilyet már rajzoltunk egyébként a 2.5. fejezetben a 2.79. ábra AOI megvalósításához!). Egy bővithető kapuhoz több bővítő áramkör is csatlakoztatható (legfeljebb négy), bonyolultabb függvények előállítására.

Módosított bemenetű és kimenetű áramkörök

A következőkben azokat az áramköri változatokat tanulmányozzuk, amelyek vagy bemenetükön, vagy kimenetükön eltérnek a "normál" TTL-től. Az eltérés vonatkozhat valamelyik villamos ki/bemeneti jellemzőre, de lehet, hogy az áramköri felépítés és annak fizikai működése más. Nyilvánvaló, hogy ezeknek a "speciális" áramköröknek rendeltetése is más, mint a normál típusoknak, ezért a felhasználási területeikkel is foglalkoznunk kell.

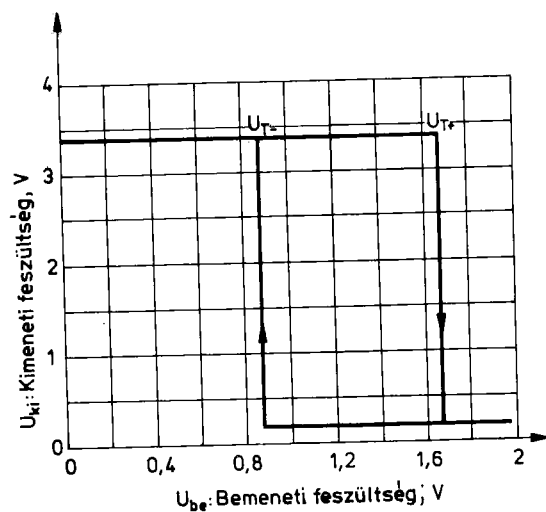


3.51. ábra

a) Schmitt-trigger bemenetű TTL áramkörök

Jellegzetessége, hogy $U_{be} - U_{ki}$ transzfer karakterisztikájában histerézis van. A histerézis ebben az esetben azt jelenti, hogy a bemeneti feszültséget 0 V-ról növelve a billegési szint (a logikai 0-t és 1-et elválasztó határfeszültség, küszöbszint) másutt, magasabban van, mint "visszatéréskor", - amikor a bemeneti feszültséget pozitív, logikai 1-ből csökkentjük 0 V felé. A transzfer karakterisztika így nem egyetlen vonalból áll, hanem hurok alakú (histerézis hurok). Ennek vázlatát láthatjuk a 7414 Schmitt trigger bemenetű inverter (vagy akár a 74132 kétbemenetű vagy a 7413 négybemenetű NAND Schmitt-trigger) típusokra a 3.52. ábrán: ha az $U_{be} = 0$, akkor a kimenet pozitív, logikai 1 szintű (az invertálás miatt), amikor az U_{be} feszültséget fokozatosan növeljük, akkor

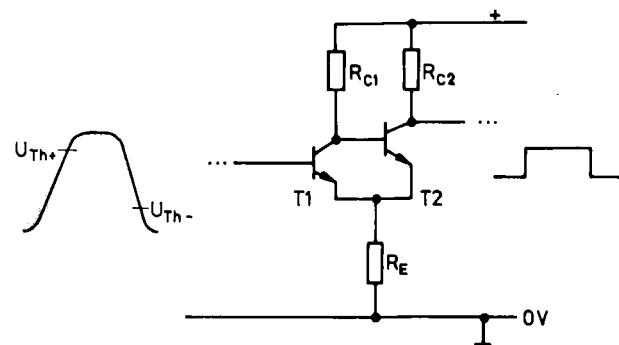
nem a szokásos bemeneti 1,2...1,3 V-nál vált át a kimenet 0 V közelébe, hanem az U_{Th+} (pozitív küszöb: pozitív küszöb) feszültségen +1,7 V körül. Ha ezután a bemeneti feszültséget újból csökkentjük, akkor +1,7 V-nál nem történik semmi, csak U_{Th-} (negatív küszöb, 0,9 V körüli feszültség) elérésekor "ugrik" vissza a kimenet feszültsége 1-es szintre (a billenéseknek "iránya" van, ezt mutatják a karakterisztika vonalakra rajzolt nyilak).



3.52. ábra

A Schmitt-trigger áramkör "magja" a 3.53. ábrán látható két tranzisztorból álló kapcsolás. Amikor a bemenet 0 V közelében van, T1 zárt (olyan, mintha ott sem lenne), a T2 nyitott, mivel bázisa R_{C1} -en keresztül áramot kap és nagyjából R_{C2} és a közös R_E által meghatározott "osztás potenciálon" tartja a két összekötött emittert. A bemeneti feszültség növelésével ezt, az emitterekre beállított +potenciált kell elérnünk, ill. ennél kb. 0,6 V-tal nagyobbat (U_{Th+}) ekkor kezd kinyitni T1. Az ezután lejátszódó folyamat igen gyors, billenészerű: amikor T1 kezd kinyitni, kollektorfeszültsége csökken, emiatt T2 lezárás felé halad, ami miatt viszont a közös emitter potenciál csökken. Ettől T1 jobban kinyit, hiszen (a

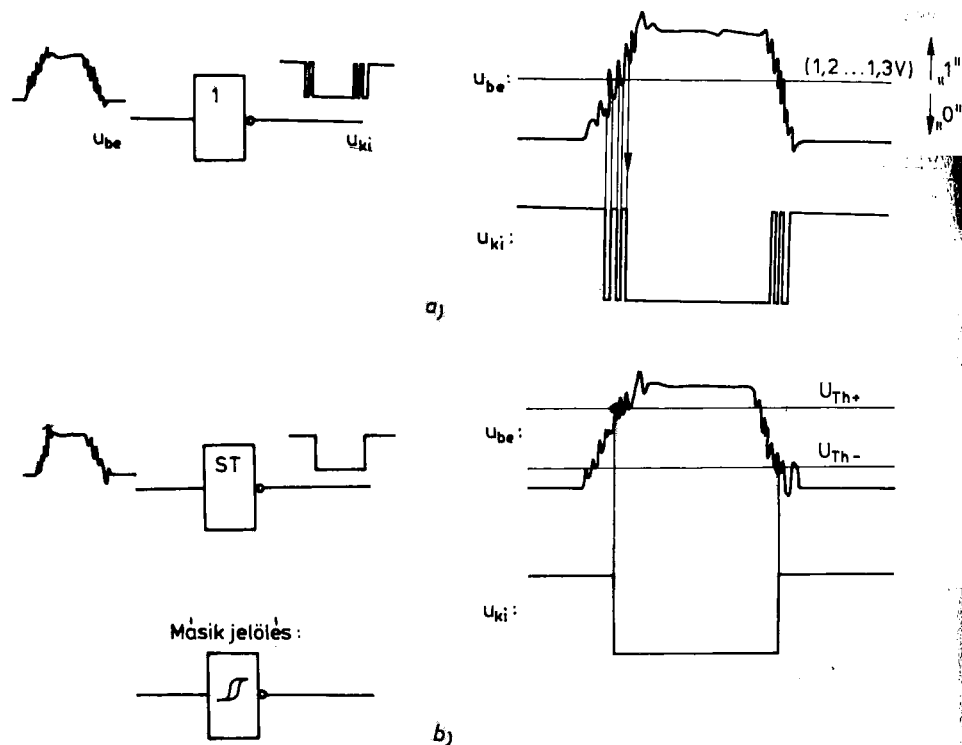
bemeneti feszültség nem változott) a nyitó bázis-emitter feszültsége növekedett. Emiatt T1 méginkább kinyit, T2 méginkább lezárás felé halad stb. (erre a rövid időre az "erősítő" aktív, pozitív visszacsatolt üzemben van). A folyamat végeredménye T1 teljes kinyitása, T2 lezárása, a kimeneti feszültség magasabb szintre ugrott. A bemeneti feszültség további növelése nem változtat a helyzeten, sőt ha ismét a zérus felé csökkentjük, akkor változás nagy tartományban nem következik be, T1 továbbra is nyitott. Csak ha az előző szintnél jóval alacsonyabb bemeneti feszültséget érünk el (U_{Th-}), akkor csökken le T1 kollektorárama olyan mértékben, hogy kollektor-emitter feszültsége meghaladja a 0,6 V-ot, T2 bázis-emitter nyitófeszültségét, így T2 ismét vezetni kezd, megemeli a közös emitter feszültséget, ezzel T1-et jobban lezárja, stb. - így a trigger ismét billenészerűen visszaáll alaphelyzetbe. Maga a Schmitt-trigger alapáramkör nem invertál, a TTL "ST" kapokban, inverterekben további fokozatok vannak (bemeneti áramkör, szint-áttevő, végfokozat).



3.53. ábra

A bemeneti hiszterézis és a billenészerű átkapcsolás teszi alkalmassá a Schmitt-trigger - több más mellett - arra az alapvető funkcióra, hogy a "hasznos" jelre szuperponálódott zavarjel hatását bizonyos határok között kiiktassa, valamint hogy lassu jelekből "határozott" négyszögjelet állítson elő. A lényegét a 3.54. ábra alapján érthetjük meg: ha "normál" kapu bemenetére érkezik olyan jel, amelyre nagy amplitudóju za-

varjel szuperponálódott (a ábra), akkor - mivel a normál kapunak egy határozott trigger szintje van, e szint alatt 0-nak, e felett 1-nek "érzi" a jelet - ahányszor csak átlépi a "zavart" jel ezt a trigger szintet, annyiszor vált át a kimenet 1-ből 0-ba, 0-ból 1-be.



3.54. ábra

A kimeneten tehát nem egy "határozott" négyyszögjel áll elő, hanem az élék helyén sokszoros 0-1 átmenet. Belátható, hogy a legtöbb esetben ez igen zavaró, gondoljunk arra, hogy ezt a négyyszögjelet pl. egy számláló számolja (pl. frekvenciamérés, vagy időmérés céljából), akkor egyetlen négyyszögjel helyett nem egyet számol, hanem az is lehet, hogy sok ezret! Különösen nagy a téves triggerelés veszélye akkor, amikor a beérkező jel lassu fel- és lefutású (esetleg szinuszból formált jel), ilyenkor az óhatatlanul rákerülő nagyfrekvenciás zavarnak (rá-

dió, TV stb.) bőséges "ideje van" arra, hogy a log 0-1 tartományt elválasztó trigger szintet igen sokszor átlépje (az igazsághoz tartozik, hogy a TTL kapuk a lassu jelekre amugyis "bizonytalan", nagyfrekvenciás berezgésekkel teli kimeneti jellel "válaszolnak"). A Schmitt-trigger zavar-elhárító hatását a 3.54b ábra szemlélteti, a 0-ból 1-be menő, "zavart" jel mindaddig nem okoz változást a kimeneti jelben, amíg a bemeneti jel (és zavarjel összegének) pillanatértéke először át nem halad a felső, U_{Th+} küszöbfeszültségen, ekkor a kimenet átbillen 0-ba. A bemeneti jel ezután még többször metszi az U_{Th+} szintet, de mivel az átbillenés megtörtént, a kimeneti jel nem változik! A bemeneti jel visszafutásakor, 0-ba való visszatérésekor csak akkor áll be változás, amikor a jel+zavar pillanatértéke először átlépi az alsó, U_{Th-} küszöbfeszültséget, ekkor a kimenet átbillen 1-be, és stabilan ott is marad még akkor is, ha a bemeneti jel többször metszi az U_{Th-} szintet. Az esetleg lassu bemeneti jelből a trigger billenésének köszönhetően gyors fel-lefutású négyyszögjel keletkezett (ez jelformálásakor előnyös), a zavarjelek pedig nem okoznak többszörös kimeneti jel-ugrást. Egy feltétel van csak: a zavarjel csúcstól-csucsig vett amplitúdója a két küszöbszint különbségénél (a "hiszterézis feszültségénél") csak kisebb lehet:

$$U_{zpp} < U_{Th+} - U_{Th-} = \text{hiszterézis feszültség}$$

(az említett TTL típusokra $U_{Th+} = 1,7 \text{ V}$, $U_{Th-} = 0,9 \text{ V}$, tehát a hiszterézis feszültség $0,8 \text{ V}$).

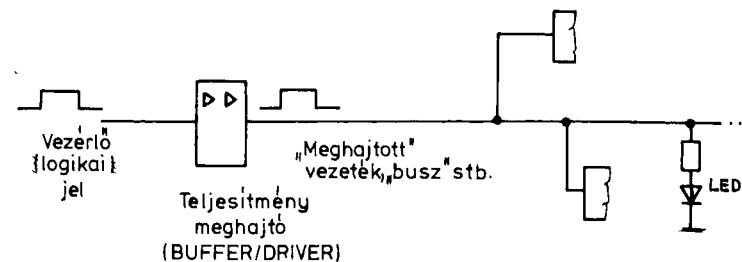
A Schmitt-trigger azonos funkciót valósít meg az analóg áramkörök kategóriájába tartozó hiszterézises komparátorral, a fő különbség az, hogy:

- a szokásos műveleti erősítőkből felépített áramkörök sokkal lassabbak, mint a TTL változatok, amelyek "TTL sebességgel" ($t_{pd} \approx 15 \text{ ns}$) működnek,

- az erősítőkből felépített áramkörök hiszterézisét a külső elemekkel tetszés szerint be lehet állítani, a TTL Schmitt-trigger konstans (de a katalógus szerint hőmérséklet-kompensált) szintekkel dolgozik.

b) TOTEM-POLE kimenetű meghajtók (BUFFER elemek)

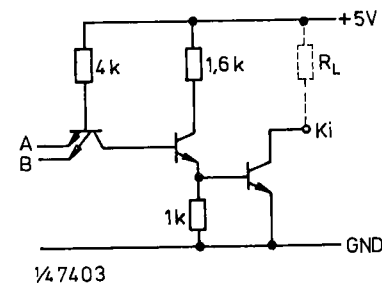
A normál TTL inverterei, kapui és egyéb áramkörei általában az $N = 10$ -es FAN OUT számnak megfelelően 10 ujjabb bemenettel terhelhetők úgy, hogy a logikai szintek még türezen belül maradnak. Ez gyakran nem elegendő vagy azért, mert egy adott kimenetre 10-nél több áramkör bemenete csatlakozik, vagy azért, mert valamely más terhelés (pl. egy másik, nagyobb bemenő áramot fogyasztó áramkör változat, vagy valamilyen "fogyasztó": lámpa, stb.) kapcsolódik rá. Az is lehet, hogy a logikai jelet hosszabb vezetékre kívánjuk adni, erre leggyakoribb példa egy teljes elektronikus rendszeresen végighaladó, több helyre elvezetett, leágasztott "busz" (bus) vezeték. Ilyenkor nagyon fontos, hogy a meghajtó generátor megfelelő teljesítmény leadására legyen képes és kimeneti impedanciája igen kicsi legyen (ilyenkor tulajdonképpen egy teljesítményerősítő beiktatására van szükség) azért, hogy a terhelő impedanciák - amelyeknek hosszú busz-vezeték esetében tetemes kapacitív összetevőjük van - ne lassítsák a rendszert, ne torzítsák el a logikai szinteket, ne "huzzák el" a feszültségeket. A legegyszerűbb TTL buffer, driver (meghajtó) elemek, amelyeknek totem-pole (ellenütemű), nagyáramú végfokozatuk van, 10 helyett $N = 30$ ugyanolyan családbeli bemenetet képesek meghajtani. Ha ennél még nagyobb a terhelés, akkor megengedett és szokásos módszer több azonos típusú áramkört párhuzamosan kapcsolni azzal a szigorú kikötéssel, hogy minden párhuzamos egységnek ugyanazt a vezérlőjelet kell kapnia, vagyis a bemeneteket is párhuzamosan kell kötnünk! (Legismertebb buffer kapuk a 7437: 4-szer 2 bemenetű NAND, a 7428: 4-szer 2 bemenetű NOR). Léteznek ezenkívül "különleges" busz-meghajtók, "adó-vevők", valamint az interface sorozatban (SN 75...) nagyteljesítményű meghajtók, ezekkel még találkozunk. A teljesítményerősítő szabványos jelét a 3.55. ábra mutatja.



3.55. ábra

c) Nyitott kollektoros kimenetű TTL áramkörök (OPEN COLLECTOR OUTPUT)

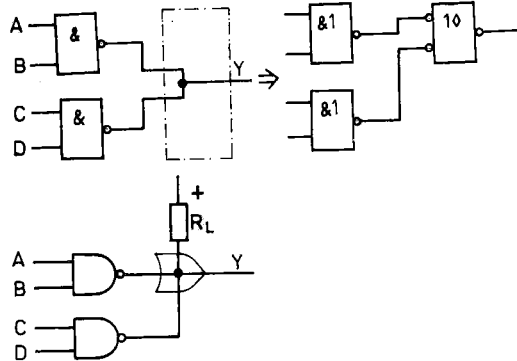
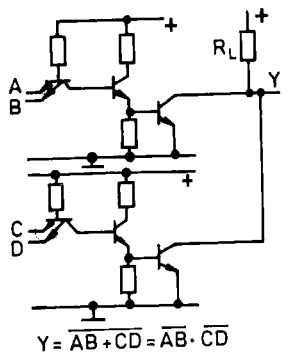
A normál, totem pole végfokozatu kapuk használatának fontos szabálya, hogy különbözőképpen vezérelt áramkörök kimeneteit nem szabad egymással összekötni, "párhuzamosítani". Ilyenkor ui. előfordulhat, hogy az egyik kimenet logikai 1-re, a másik logikai 0-ba menne, ezért a kimenetek között nagy áram folyik, emiatt tönkremehet a logikai 0-ba vezérelt kapu T_3 áramhúzó tranzisztora. Ezenkívül a kimeneti feszültség is határozatlan lesz; valamilyen 0 és 1 közé eső értékre áll be, és nem lesz alkalmas ujjabb áramkörök vezérlésére. Azonos típusú kapuk kimeneteit akkor szabad csak párhuzamosan kapcsolni, ha a bemenetek is párhuzamosak. Ezt sokszor alkalmazzák, amikor a kimeneti terhelhetőséget növelni kell (ilyenkor elvileg tetszőleges számú kaput párhuzamosan szabad kötni), meghajtó (buffer) célra.



3.56. ábra

A különbözőképpen vezérelt áramkörök kimeneteinek összekötése sokszor előnyös lenne, mert így - külön áramkörök nélkül - újabb logikai kapcsolatokat lehetne létrehozni. Ezt a lehetőséget adja meg az OPEN COLLECTOR-os áramkör (a 3.56. ábrán NAND kapu látható). Totem pole végfokozat helyett csak az alsó inverter huzótranszisztor (T3) van beépítve, munkaellenállást kívülről kell rákapcsolni. Az áramkör kis kimeneti impedanciával csak a földpotenciál felé tud "huzni". Logikai 1-ben az open collector-os kimenet szakadásként viselkedik. Több kapu kimenetét párhuzamosan lehet kapcsolni és egyetlen közös munkaellenállással ellátni és így, csupán huzalozással újabb logikai kapcsolatot hozhatunk létre. A 3.57. ábrán látható két NAND kapu közös kimenetén akkor lesz logikai 0 feszültség, ha vagy a felső, vagy az alsó (vagy mindkét) kapu T3-as végtranszisztorja kinyit és a közös kimenetet összeköti a földdel. Ez akkor következik be, a kimenet akkor lesz 0, ha vagy A és B, vagy C és D logikai 1-ben van:

$$Y = \overline{AB + CD}$$



3.57. ábra

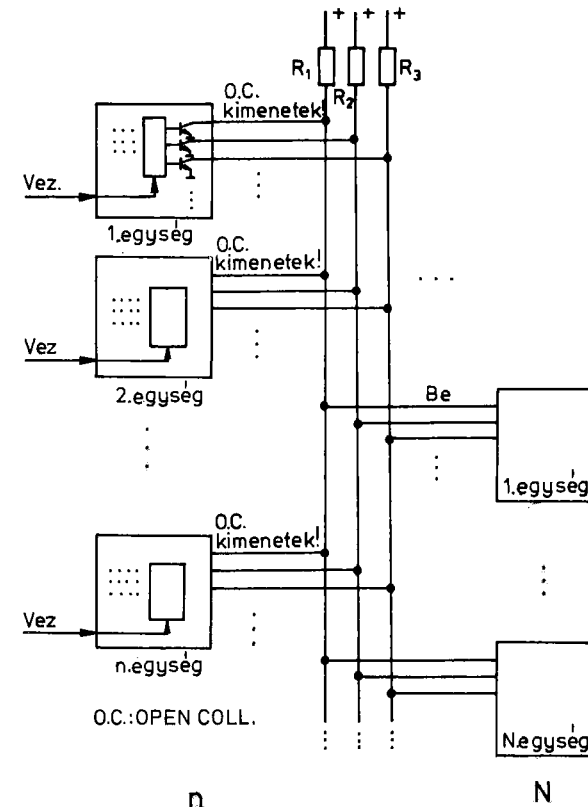
A kimeneti függvényben a negáció jele alatt egy külön logikai VAGY kapcsolat jött létre, ezért az ilyen kapcsolást huzalozott (wired) vagy az összekötés "pont"-ja után "DOT" logikának szokták nevezni, hiszen ez a belső VAGY kapcsolat nem

újabb kapu segítségével, hanem csupán huzalozással, összekötéssel jött létre. Ha a negációt a De-Morgan szabály szerint felbontjuk, akkor két NAND függvény ÉS kapcsolatát kapjuk:

$$Y = \overline{AB + CD} = \overline{AB} \cdot \overline{CD}$$

Ezért ezt a kapcsolatot ÉS-DOT (wired-AND, egyes helyeken wired-OR) műveletnek mondhatjuk. A DOT elemek szabványos rajzjelét és a katalógusokban, szakirodalomban megtalálható szokásos jelét a 3.57b ábra mutatja erre az esetre.

A nyitott kollektoros (open-collector-os) kapuk felhasználása az előbbieket szerint gazdaságos: kapukat huzalozással helyettesíthetünk (példánkban két NAND kapuból ÉS-Nem-Vagy, AND-OR-INVERT kapcsolatot hoztunk létre).



3.58. ábra

Fokozott jelentősége van az open collector-os áramköröknek a mai áramkör technikában, ahol a (sokszor LSI) részegységek összekapcsolását a "szokásos" kapuzás helyett a már említett gyűjtővezetékek, sinek, buszok segítségével oldják meg. Totem pole kimeneteket nem lehet közös gyűjtővezetésekre rákötni, viszont a nyitott kollektoros áramkörök erre kimondottan alkalmasak. Az áramköri egységek kimenetei párhuzamosan, bitenként egy-egy közös sinre csatlakoznak (3.58. ábra), minden sinnek van egy felhúzó munkaellenállása (R_1, R_2, \dots). Az egységeket úgy vezéreljük, hogy kimeneteik 1-ben legyenek (ami annyit jelent, hogy "elengedik" a gyűjtővezetéseket), kivéve azt az egyet, amelynek a jelét fel kívánjuk használni: ezt "engedélyezzük", így ennek a kimeneti bit-kombinációja fog megjelenni a vezetékeken (logikai 0 kimeneti jelnél a hozzá tartozó végtranzisztor 0 V-ra viszi a sint, logikai 1-nél az illető huzótranzisztor lezár, a munkaellenállás felhuzza a sint pozitív feszültségre). Ez a szerkezet azért is előnyös, mert utólag módosítható, bővíthető: egyes egységek cserélhetők, utólag "rátolhatók" a sin-rendszerre, nem kell a teljes áramkört átépíteni.

A gyűjtővezetékekre kapcsolandó munkaellenállás értéke attól függ, hogy hány kimenetet huzalozunk össze (n) és hogy a közös kimenetre hány újabb bemenetet (N) csatlakoztatunk. Az R_L (load) munkaellenállás értékének a maximumát és minimumát határozzuk meg:

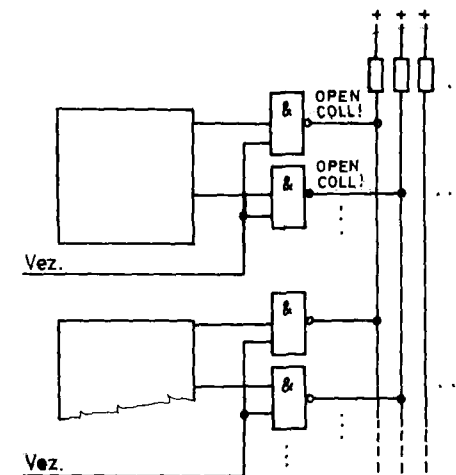
$$R_{Lmax} = \frac{U_{cc} - U_{kil\ min}}{n \cdot I_{kil} + N \cdot I_{bel}}$$

ahol: U_{cc} a tápfeszültség (+5V),
 $U_{kil\ min}$: a minimális logikai 1 feszültség (2,4 V),
 I_{kil} : a kimenő (T3) tranzisztor maradékárama 1-ben (250 μ A),
 I_{bel} : egy áramkör bemenő árama 1-ben (40 μ A)

$$R_{Lmin} = \frac{U_{cc} - U_{ki0\ max}}{I_{ki0max} - N \cdot I_{be0}}$$

ahol: $U_{ki0\ max}$: a maximális logikai 0 feszültség (0,4 V),
 $I_{ki0\ max}$: a legnagyobb megengedett huzóáram (16 mA),
 I_{be0} : a bemenő huzóáram (1,6 mA)

(a zárójeles adatok normál TTL-re vonatkoznak).



3.59. ábra

Amennyiben sinre kapcsolandó áramkörök nem nyitott kollektorosak, a kimenetek és a gyűjtővezetékek közé nyitott kollektoros invertereket, kapukat helyezünk, utóbbiakkal megoldhatjuk a vezérlést is (3.59. ábra). Erre a célra valók az előző pontban említett bufferek, meghajtók open collector-os változatai (pl.: 7406 inverter, ill. 7426, 7438 NAND bufferek stb).

Az open collector-os áramkörök nagy előnye előbb említett két funkció (huzalozott logika, busz-rendszerre csatlakozás) lehetősége. Hátrány viszont a lassabb működés. Nincs aktív felhúzó elem, hanem munkaellenállás, ezért a szórt, és vezeték kapacitások lassítják az átkapcsolást (hasonlóan az RTL-hez). A kimeneti impedancia is nagyobb logikai 1-ben, ezért a zavarérzékenység is megnövekszik. Az open collector-os logikát ott célszerű alkalmazni, ahol az előnyöket jól ki lehet használni és nincsenek szigorú előírások a sebességre! (Van esetek, amikor a sebesség teljesen közömbös, ilyenkor nem is helyeznek az áramkörbe munkaellenállást, ha a terhelés

ujabb TTL bemenet, ennek felhuzó hatása ui. elegendő: a bemeneti tranzisztor bázisában levő $4\text{ k}\Omega$ -os ellenállás szerepel munkaellenállásként.)

Az open collector-os inverterek és kapuk (főleg ezek "buffer" és "interface" típusai) jó szolgálatot tesznek akkor is, amikor egy digitális rendszerrel külső tranzisztoros stb. áramköröket kell meghajtanunk (amelyek már nem 5 V-ról működnek), valamint amikor logikai szint-átvitelre van szükség TTL és más rendszerek között, de megfelelő kiegészítéssel használhatók pl. analóg (JFET-es) kapcsolók meghajtására is (léteznek nemcsak nagy áramot, hanem nagyobb, 15...30 V feszültséget tűrő típusok is).

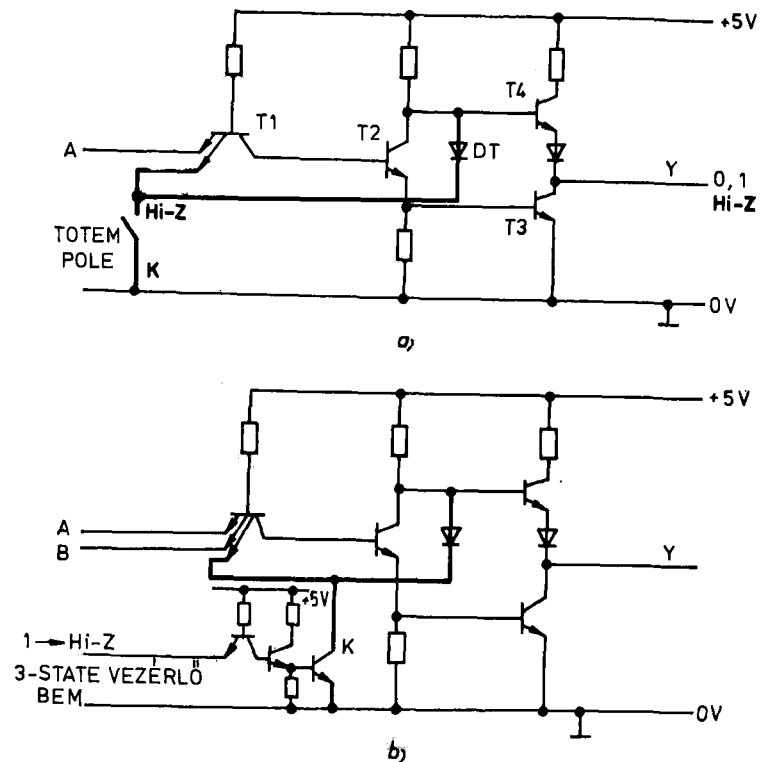
d) 3-STATE (3 állapotú, THREE-STATE "TRI-STATE")

kimenetű áramkörök

A mai áramkör technikában annyira gyakori és fontos a 3-state elemek használata, hogy részleteiben itt tárgyaljuk, bár nem TTL rendszerben létezik csak (de itt fordul elő először). Sok olyan MOS-CMOS (pl. mikroprocesszoros) rendszer van, amelyben TTL áramkört (többnyire LS változatban) csak azért használnak, mert ennek típusválasztékában vannak kellő teljesítményt leadó 3-state bufferek.

Normál, totem-pole kimeneteket - mint tudjuk - nem szabad egymással összekötni, így a ma szokásos és már említett busz-rendszerekhez nem alkalmazhatók. Az open-collector-os kimenetű áramkörök buszra köthetők ugyan, de mint láttuk, a passzív felhuzó munkaellenállás miatt lassu működésűek és nagyobb a zavarérzékenységük is. A 3-state logika megoldja ezt a problémát: lehetővé teszi totem-pole (általában ellenütemű) kimenetek közös gyűjtővezetékre való csatlakoztatását, ami által elkerülhető a sebesség veszteség. Az összekötött totem-pole (ellenütemű) kimenetek természetesen "nem dolgozhatnak egymás ellen", az összehuzalozott sok kimenet közül egyszerre mindig csak egy "dolgozhat" a buszra, adhat rá logikai 0 vagy logikai 1-es jelet, a többi áramkörnek "be kell szüntetnie működését", szakadásként kell viselkednie. Ez a "szakadás"-nak megfelelő állapot a 3-state áramkör harmadik állapota. Így egy 3-state áramkör lehetséges állapotai:

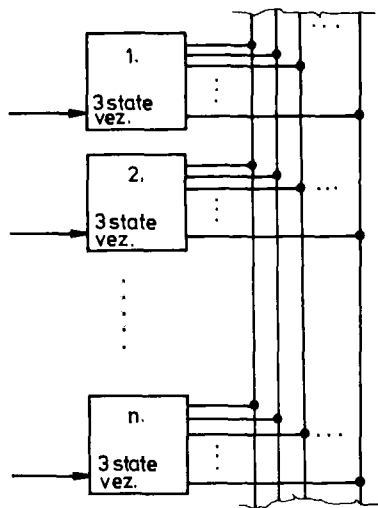
1. Logikai 1
2. Logikai 0
3. Nagy impedanciás (Hi-Z) "szakadt", "leválasztott" kimenet.



3.60. ábra

A logikai 1 és 0 kimeneti jelet a szokásos, "normál" totem-pole kimenetű áramkör állítja elő. A harmadik nagy impedanciás állapot akkor következik be, ha a végfokozat mindkét tranzisztorát, az alsót és a felsőt kikapcsoljuk, lezárjuk. A TTL 3-state inverter alapkapcsolás működését a 3.60a ábrán követhetjük. A K kapcsoló bekapcsolásakor két dolog történik: a bemeneti tranzisztor egyik emittere 0 V-ra kerül, ezáltal lezár T2 és ezzel együtt az alsó végerősítő tranzisztor, T3. Másrészt a beiktatott dióda (DT) - katódja 0 V-on lévén - lehúzza a felső végerősítő tranzisztor, T4 bázisát 0 V közelébe, ez-

által T4 is lezár, a kimenet tehát "lebegő" lesz függetlenül attól, hogy az A jelbemeneten milyen szint van. A K kapcsoló vezérlésére külön 3-state vezérlő bemenet van (pl. a 3.60b ábra szerint: logikai 1 vezérlés nagyimpedanciás, Hi-Z állapotba viszi a kimenetet, logikai 0 pedig engedélyezi a "normál" működést). Természetesen nemcsak inverter, hanem - több jelbemenettel - NAND kapu (3.60b ábra), sőt ugyanezen az elven egész áramkör, funkcionális egység is készülhet 3-state kimenettel. A 3-state vezérlő bemenet elnevezése, jelölése is változatos, pl.: TSC (Tri-State Control: 3. állapot vezérlő), \overline{En} (\overline{Enable} : engedélyező-negált: 0-ra engedélyez, 1-re 3-state), G (Gate: kapujel), \overline{OE} ($\overline{Output Enable}$ kimenet engedélyező-negált), sőt az LSI áramkörök működését engedélyező \overline{CE} ($\overline{Chip Enable}$: IC engedélyező-negált) vagy \overline{CS} ($\overline{Chip-Select}$: IC kiválasztás negált) jelek is vezérlik a kimenet 3-state vagy "normál" működését.



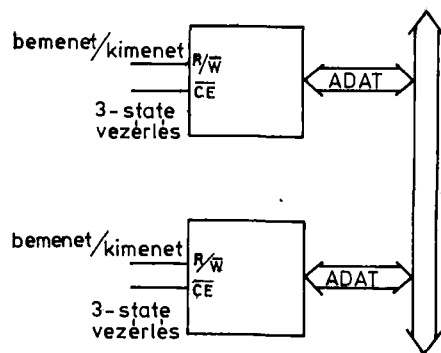
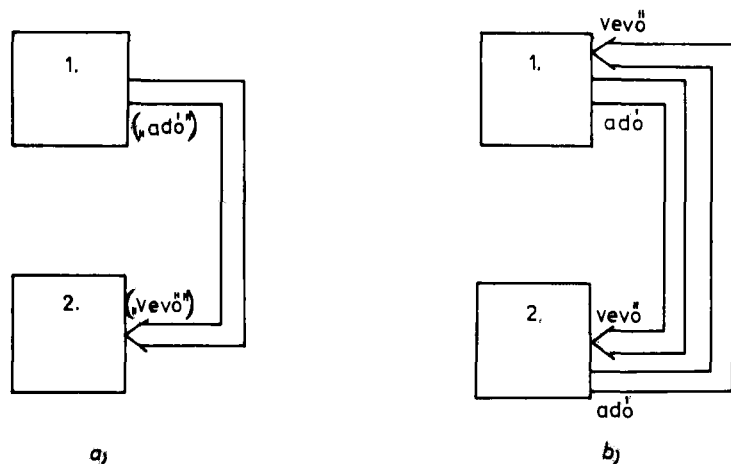
3.61. ábra

A tri-state kimenetű elemek áramköri egységek az open collectoros áramkörökkel ellentétben sebesség veszteség nélkül "fűzhető" közös sinre, busz-vezetékre minden külön alkat-

rész (felhuzó ellenállás stb.) igénye nélkül (3.61. ábra). Arról természetesen jelen esetben is gondoskodni kell, hogy egyszerre csak egy egység adja jelét a közös gyűjtővezetésekre. Az open collector-os szerkezettel ellentétben, most nem kell külön kapuzással tiltani az áramköröket, hanem csak a tri-state vezérlő bemenetekre kell a megfelelő tiltó, ill. engedélyező jelet rávezetni. Említettük, hogy a bonyolultabb (főleg "intelligens"), LSI IC-eket tartalmazó rendszerek huzalozását nagymértékben egyszerűsíteni lehet busz-szerkezettel. Ezekben általában több (8-16-32-) párhuzamos bitből álló jelek "közlekednek". Célszerű ezeket a jeleket "bit-párhuzamosan" vezetni, azaz minden bithez egy-egy vezeték a rendszeren végigvinni és ezekhez kapcsolni párhuzamosan minden egységet. A jelek egy része olyan, hogy egy adott pillanatban valamelyik egység kimenetén áll elő és egy másik egység bemenő jelként fogadja, de egy későbbi működési fázisban a fogadó által előállított jelnek kell eljutnia az előzőleg adóként működő egységbe (3.62a és b ábra). Kézenfekvő és szokásos megoldás, hogy az ilyenfajta "kétirányú" működés biztosításához nem építünk ki egy "adó" vezeték-köteget és egy külön vevő köteget, hanem (mivel az adatforgalom időben elkülönül, adott sorrendben játszódik le) közös, kétirányú busz rendszert készítenek (3.62c ábra). Az ábrában a több párhuzamos vezetékből álló busz-rendszert vastag nyíllal jelöltük, a szokáshoz híven.

Kétirányú busz-rendszer esetén nemcsak arról van szó, hogy az egyes áramköröknek a "párhuzamos" összekötés miatt 3-state kimenetűeknek kell lenniük, hanem arról is, hogy ugyanazok a kivezetések vezérelhetően az áramkör (3-state) kimenetei, ill. ellenkező vezérléskor bemenetei kell hogy legyenek. Gondoljunk pl. memória áramkörökre: kell minden egységhez egy 3-state vezérlő jel, amely azt irányítja, hogy az illető IC éppen kapcsolatban legyen a busszal vagy nem, de kell egy másik vezérlés is, amely megmondja, hogy a buszra kapcsolódás esetén kimenetként vagy bemenetként kezelje kivezetéseit. Ezt természetesen belső áramköri felépítése teszi lehetővé. Több áramkör összekapcsolásánál - mint például egy mikroprocesszor rendszerben - a helyzet bonyolultabbnak látszik ugyan, de éppen ez az egyszerű összekapcsolás és vezérelhetőség teszi vég-

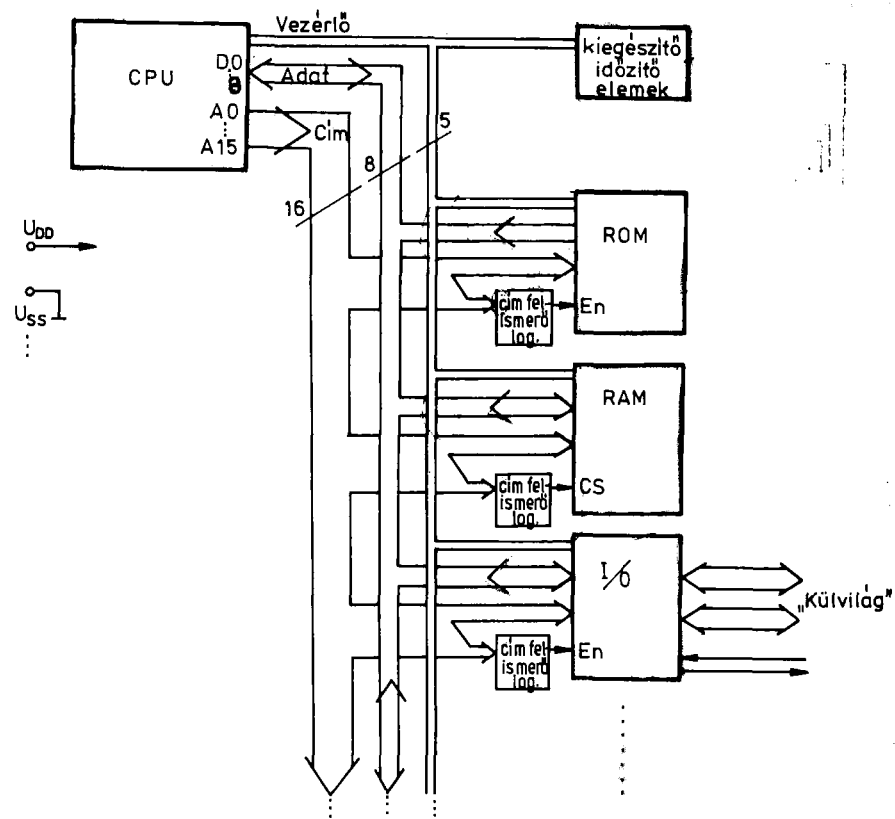
eredményben egyszerűvé, világossá a működést. Ilyenkor természetesen szükséges egy központi egység - maga a processzor - amely sok más feladata mellett olyan vezérléseket küld minden egységhez, amelyekkel biztosítja az előírás szerinti sorrendű működést.



3.62. ábra

A 3.63. ábrán egy mikroprocesszoros rendszer szokásos busz-strukturáját rajzoltuk fel. Nem feladatunk a működés részletes tanulmányozása - ezzel később foglalkozunk -, csupán busz-rendszer "konceptióját" kell megértenünk. A központi egység (CPU: Central Processing Unit) ebben az esetben háromféle

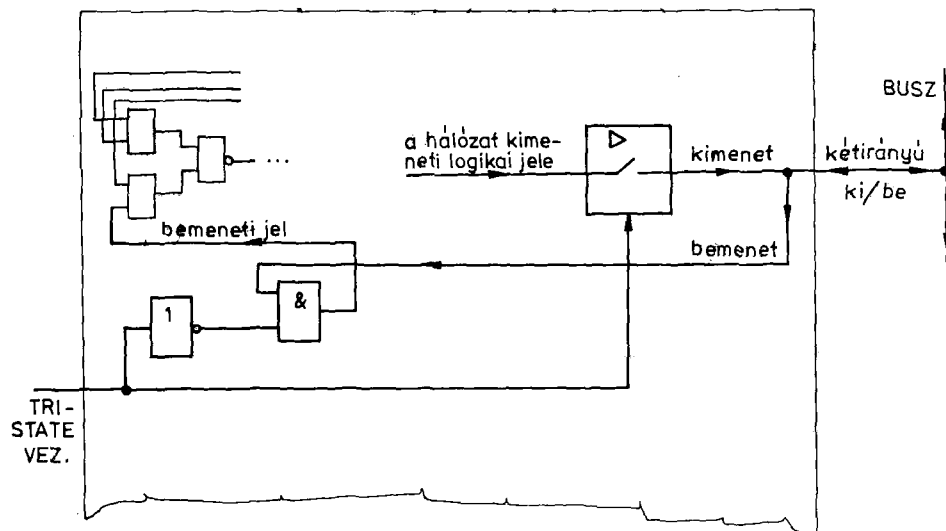
párhuzamos busz-rendszer segítségével kapcsolódik a többi "életfontosságú" egységhez (a földvezetékét, tápvezetékét nem számítva):



3.63. ábra

A CIM-busz (ADDRESS BUS), amely egyirányú, tipikusan 16 vezetékből áll. Ennek segítségével közli a központi egység, hogy melyik egység (ROM: Read Only Memory, állandó tartalmu "csak olvasható" memória, RAM: Random Access Memory, "nem rendszeres" hozzáféréssé, írható-olvasható memória, I/O: Input Output egységek, amelyek a "külvilághoz" csatlakoznak és a további bővítések) melyik részéhez, regiszteréhez, rekeszéhez, "kiván fordulni". Az egységek a cím vezetéseken érkező jelkombinációkból (2^{16}) kell, hogy felismerjék a saját címüket, eh-

hez szükséges a rajzon szereplő cím felismerő áramkör (cím dekódoló). A cím vezeték tehát a CPU felől, mint kimenettől viszik a jelet a többi áramkör bemenetére (rendszerint MOS áramkörök).



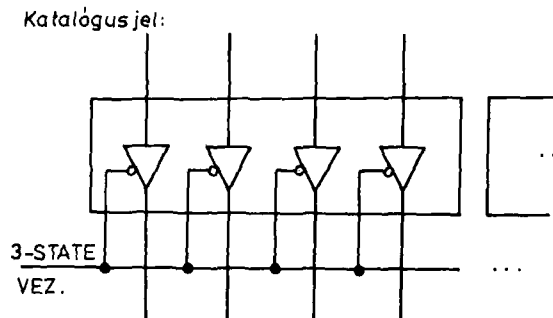
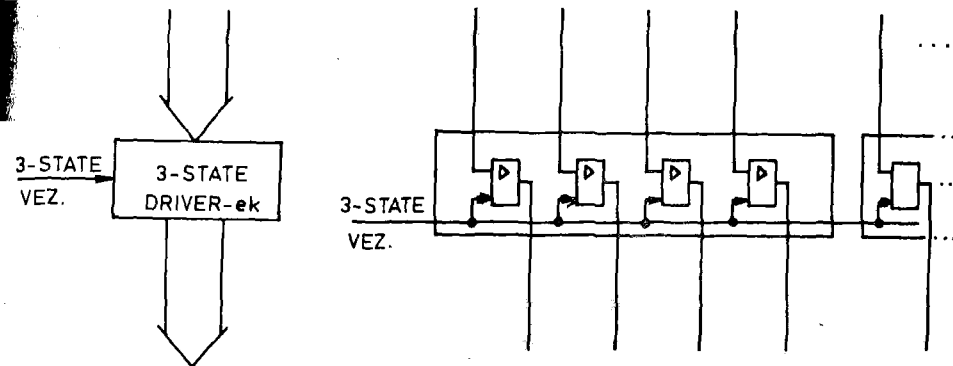
3.64. ábra

Az ADAT-busz (DATA BUS), tipikusan 8 bites, viszont kétirányú. Vannak esetek, amikor a CPU "küld" adatot valamelyik egységbe, akkor a CPU oldalon az adat-vezetékek kimenetként szerepelnek, a többi áramkörnél bemenetként. Más ciklusokban fordított az irány: valamely egység küld jelet a CPU számára, ekkor az illető egység kimenetként kezeli ugyanazokat a vezetékeket, amelyeket az előbb bemenetként kezelt, a CPU viszont "belső áramkörei" bemeneteit köti az adatbusz vezetékekre, miközben az ugyanezekre a pontokra csatlakozó kimeneteit Hi-Z állapotba állítja. A lényeg: kétirányú busz esetén egyetlen áramkörön belül egy pontban a ki/bemeneten találkozik a kimenet és a bemenet. Egyszerre természetesen mindig csak az egyik "érvényesülhet" a vezérléstől függően, a másik tiltott: a bemenet logikai tiltással, a kimenet amikor szükséges Hi-Z állapotba állítással (3.64. ábra)!

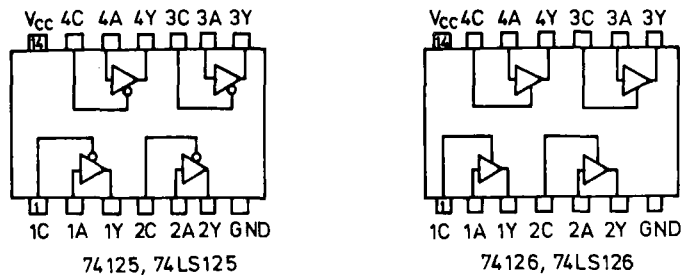
A VEZÉRLŐ-busz (CONTROL BUS) éppen ezt a választást vezérli: megfelelő időzítések kíséretében eldönti, hogy adott pillanatban adott áramkör kimenő jelet ad-e vagy fogadja az adat-jelet. (Ne felejtsük a felvázolt működési séma rendkívül leegyszerűsített!)

e) 3-STATE busz-meghajtók (egyirányú: BUS-DRIVER, kétirányú: "TRANSCIEIVER")

A nagyobb rendszerekben felhasznált, rendszerint MOS, nagymértékben integrált (LSI) áramkörök kimeneti "meghajtó" képessége általában kevés ahhoz, hogy "hosszu", sokféle leágazó busz-vezetékeket megfelelő zajtartalékkal meghajtson (főleg a kapacitív terhelések "veszélyesek": lelassítják a működést).



3.65. ábra



3.66. ábra

Az egyirányú vezeték "megerősítése" általában úgy történik, hogy megfelelő helyen valamennyi párhuzamos vezeték "megszakítják" és közbeiktatnak egy 3-state driver sort (3.65. ábra). Erre általában a TTL (LS) típusok a legalkalmasabbak, ezeket kimondottan erre a célra készítik. Ilyen pl. a 74LS125 ebben 1 tokban 4 db meghajtó van külön-külön 3-state vezérlő (C) kivezetésekkel. A 74LS126 hasonló csak ez akkor "bénítja" a kimeneteket, ha a vezérlő (C) bemenet 0 szinten van. Katalógus rajzukat a 3.66. ábra mutatja. A busz vezeték száma legtöbbször 8, ill. többször 8, ezekhez jól alkalmazhatók a 74LS240...sorozat tagjai (bemenő áramuk kicsi: 0-ban 0,25 mA, kimeneti terhelhetőségük viszont nagy: 24 mA húzó- és 15 mA "forrás" áram, ezenkívül gyorsak is: $t_{pd} = 10...12$ ns, katalógus lapjukat a 3.67. ábrán láthatjuk. Valamennyi típus érdekessége és használhatóságukat növeli, hogy valamennyi bemenetük (a 3-state vezérlő is) Schmitt-triggeres, ami a zajtartalékra kedvező hatású. A 74LS240 típus 8 db invertáló 3-state kimenetű buffert tartalmaz, négyenként külön 3-state vezérlő bemenettel. A 74LS244 hasonló, csak neminvertáló. A 74LS241-es típusban is 8 db neminvertáló buffer van, de a 3-state vezérlések 4-es csoportonként ellenüteműek - szemmel láthatóan abból a célból, hogy

- kétirányú adatforgalmat lebonyolító vezetékbe iktathatók legyenek a 3.68. ábrán látható elven egy-egy buffer "antiparalell" kapcsolásával. Az 1G és 2G vezérlő bemeneteket egymással összekötve és ezt a pontot azonosan vezérelve kijelölhető a jel-irány (0 esetén az ábra példa vezetékén letről felfelé, 1 esetén fentről lefelé közlekedhet a jel,

TYPES SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241 OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

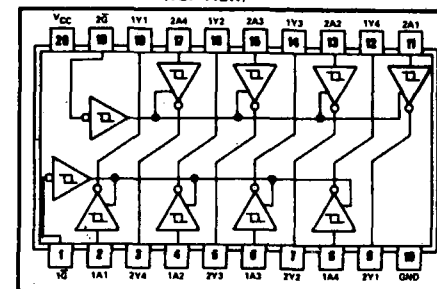
	Typical IOL (Sink Current)	Typical IOH (Source Current)	Typical Propagation Delay Times		Typical Enable/Disable Times	Typical Power Dissipation (Enabled)	
			Inverting	Noninverting		Inverting	Noninverting
SN54LS'	12 mA	-12 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN74LS'	24 mA	-15 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN54S'	48 mA	-12 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW
SN74S'	64 mA	-15 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW

SN54LS240, SN54S240 ... J
SN74LS240, SN74S240 ... J OR N
(TOP VIEW)

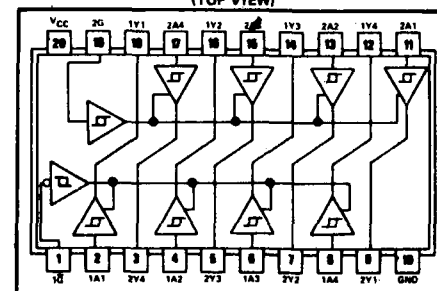
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- P-N-P Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

description

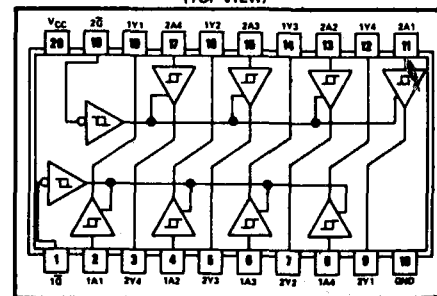
These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical \bar{C} (active-low output control) inputs, and complementary G and \bar{G} inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.



SN54LS241, SN54S241 ... J
SN74LS241, SN74S241 ... J OR N
(TOP VIEW)

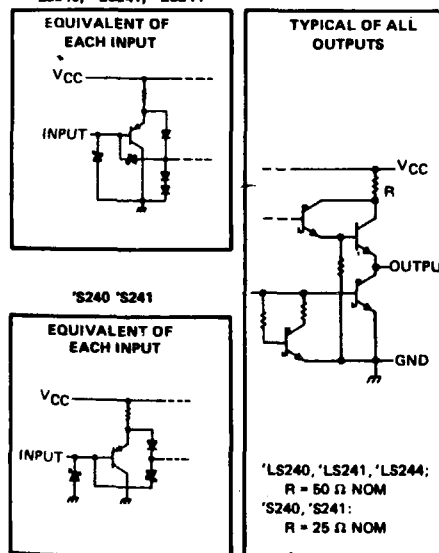


SN54LS244 ... J
SN74LS244 ... J OR N
(TOP VIEW)

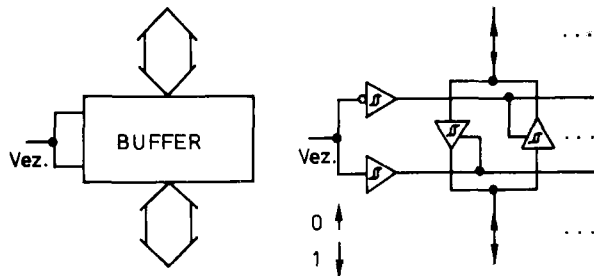


Schematics of inputs and outputs

'LS240, 'LS241, 'LS244



miközben a beiktató teljesítményerősítő "erősíti" a jelet: kis kimeneti impedanciával, nagy terhelhetőségűvé alakítja a meghajtott vezetékét). Egy ilyen áramkör 4 vezetékhez elegendő. Nagyobb mértékben integrált (MSI) típus a 74LS245: 8 vezeték kiszolgálására készült (OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS: 8-as busz "adó-vevők" 3-state kimenettel, katalógus rajza a 3.69. ábrán látható). Külön enable G bemenete van, ezzel tiltható az adat mindkét irányban (1-es szinttel), másik bemenete: DIR (DIRection control), ennek vezérlésével határozhatjuk meg az adatáramlás irányát (0: B-től A felé, 1: A-tól B felé). A 74LS240 sorozat példa arra, hogy "újabb generációs" áramköröket "már csak" LS (kisteljesítményű Schottky), ill. S (Schottky) változatban hozott ki a Texas Instruments (nem létezik "74240"!). Más gyárak típusválasztékában is található ezekhez hasonló busz-buffer változatok (pl. NATIONAL DM 81LS97-98...).



3.68. ábra

Összegezve: a busz-struktúra nagymértékben leegyszerűsíti egy rendszer kapcsolási rajzát, de a "párhuzamosításhoz" - erre alkalmas és megfelelő időrendben vezérelt open collector-os, méginkább 3-state áramkörök szükségesek.

TTL változatok

A bevezetőben már említettük, hogy a TTL-nek összesen 6-féle különböző sorozata, változata van forgalomban, mely változatok sebességben, disszipációban, árban különböznek egymástól. A leggazdagabb típusválasztéku és a "kezdeti időkben" legjobban elterjedt normál TTL (STANDARD TTL) áramkört az előzőekben megismertük, a többi változatot a következőkben tekintjük át.

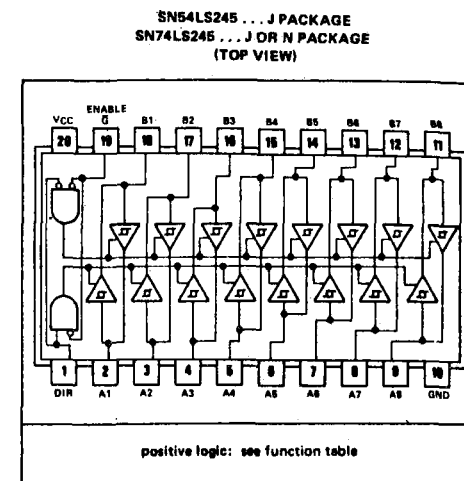
**TTL
MSI**

**TYPES SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS**

BULLETIN NO. DL-S 7612471, OCTOBER 1976

- Bi-directional Bus Transceiver in a High-Density 20-Pin Package
- 3-State Outputs Drive Bus Lines Directly
- P-N-P Inputs Reduce D-C Loading on Bus Lines
- Hysteresis at Bus Inputs Improve Noise Margins
- Typical Propagation Delay Times, Port-to-Port ... 8 ns
- Typical Enable/Disable Times ... 17 ns

TYPE	IOL (SINK CURRENT)	IOH (SOURCE CURRENT)
SN54LS245	12 mA	-12 mA
SN74LS245	24 mA	-15 mA



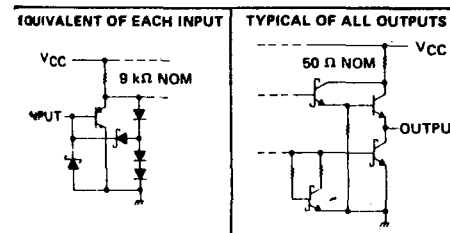
description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

The device allows data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input (\bar{G}) can be used to disable the device so that the buses are effectively isolated.

The SN54LS245 is characterized for operation over the full military temperature range of -55°C to 125°C . The SN74LS245 is characterized for operation from 0°C to 70°C .

schematics of inputs and outputs



FUNCTION TABLE

ENABLE \bar{G}	DIRECTION CONTROL DIR	OPERATION
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = high level, L = low level, X = irrelevant

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, VCC (see Note 1)	7 V
Input voltage	7 V
Operating free-air temperature range: SN54LS245	-55°C to 125°C
SN74LS245	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

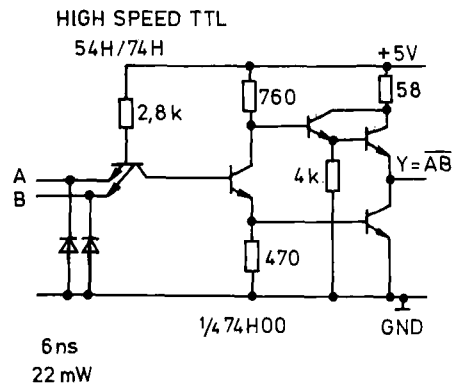
DESIGN GOAL

This page provides tentative information on a product in the developmental stage. Texas Instruments reserves the right to change or discontinue this product without notice.

TEXAS INSTRUMENTS

a) 54H/74H HIGH SPEED, nagysebességű TTL

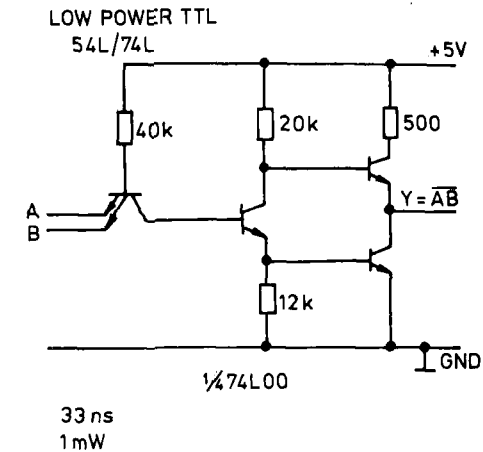
A kapuk jelterjedési késleltetése 6 ns (szemben a standard TTL 10 ns-os idejével). A nagyobb sebesség ára a nagyobb áramfogyasztás; az ellenállásértékek kisebbek, a tranzisztorok munkaponti árama nagyobb a gyorsabb működés érdekében. A teljesítmény fogyasztás 22 mW egy kapu áramkörre. A kapcsolási elrendezés (3.70. ábra) hasonlít a normál TTL-re, eltérés a kisebb ellenállásértékekben van, valamint abban, hogy a végfokozat felső, emitterkövető tranzisztora előtt még egy emitterkövető van, ami a kimenet logikai 1-be ugrását gyorsítja.



3.70. ábra

b) 54L/74L LOW-POWER, kisteljesítményű TTL

Ennek a sorozatnak a kapunkénti teljesítmény igénye mindössze 1 mW. A késleltetési idő viszont 33 ns. Az ellenállások névleges értéke a normál változat ellenállás értékeinek gyakorlatilag tizszerese, ezért a teljesítmény disszipáció tízedrésznyi (3.71. ábra). A LOW-POWER változat felhasználása olyan helyeken célszerű, ahol a nagyobb teljesítményű tápellátás nehézségekbe ütközik vagy azért, mert a tápforrás telep, ill. akkumulátor, vagy azért, mert a tápegységnek kis helyen kell elférnie. A 33 ns késleltetési idő ipari körülmények között általában nem mondható soknak. A LOW-POWER változat áramköreinek bemenetein nincsenek zavarvédő diódák, a negatív tranzieneket a T1 kollektora és a SUBSTRATE által alkotott dióda vágja.



3.71. ábra

Mind a H, mind az L változat ma már kevésbé használatos, legfeljebb "speciális" helyeken (a H változat például gyors aritmetikai egységekben). Tipusválasztékuk is szerény.

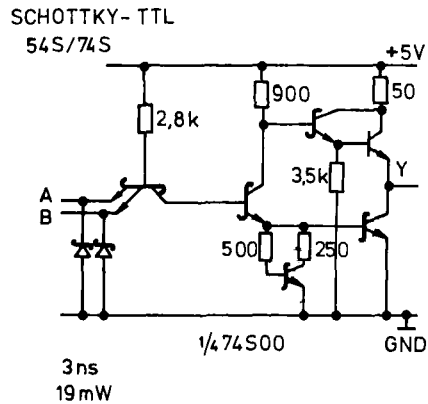
c) 54S/74S SCHOTTKY-TTL (SCHOTTKY-CLAMPED TTL)
Schottky diódával "megfogott" TTL

Modern technológiával készülő igen gyors áramkör, kapui-
nak jelterjedési késleltetése:

$$t_{pd} = 3 \text{ ns (!)}$$

Ezt csak a nemszaturált üzemmódban dolgozó ECL (lásd később) mulja felül 1 ns nagyságrendű késleltetésével. A gyors működés annak köszönhető, hogy valamennyi telítésbe vezérelt tranzisztor kollektor-bázis átmenetével párhuzamosan egy Schottky dióda van, amely megakadályozza a C-B dióda kinyitását, a tranzisztor telítődését. A viszonyokat a bipoláris tranzisztoros inverterek fejezetében már tanulmányoztuk.

A kapcsolási elrendezés (3.72. ábra) a HIGH-SPEED változathoz hasonló. A bázis jelölések a Schottky diódával védett tranzisztorokra utalnak (egyedül a végfokozat felső, emitterkövető tranzisztora nincs védve, ez telítetlen üzemben dolgozik). Egy kapu fogyasztása 19 mW (50% kitöltésű jel esetében).



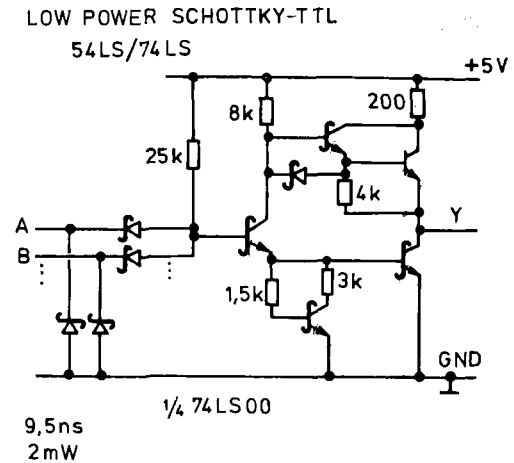
3.72. ábra

A Schottky TTL nagy előnye az extrém nagy sebesség mellett, hogy nem igényel lezárt, előírt hullámimpedanciájú jelvezető vonalakat (ami ilyen gyorsműködésű áramkörök esetében általában követelmény), működése nagymértékben független a hőmérséklettől és a tápfeszültségtől, a normál TTL-hez hasonló zajtartaléka és az ideálishoz közel eső transzfer karakterisztikája van, kimeneti impedanciája kicsi (mindez nem mondható el az egyéb gyorsműködésű áramkörökről).

d) 54LS/74LS LOW-POWER SCHOTTKY TTL, kisfogyasztású Schottky TTL

Az univerzális célra, építőelemként legelőnyösebben felhasználható bipoláris áramkör-fajta. Fogyasztása kapunként mindössze 2 mW, de - a Schottky diódás védelemnek köszönhetően - késleltetése ugyanolyan kicsi, mint a normál TTL-nek: 9,5 ns, tehát 1/5 rész fogyasztással "normál" sebességgel működik.

Az "LS" NAND kapu kapcsolása kissé eltér a többi változattól (3.73. ábra), a bemeneten a DTL-hez hasonló diódás ÉS kapu van (Schottky diódákból), ami nem változtat a többi áramkörre is jellemző áramhuzó jellegén. Az áramkör többi része lényegében egyezik az "S" változattal.



3.73. ábra

Az újabb fejlesztéshez - ha TTL-re van szükség - feltétlenül az LS változat ajánlható (hacsak "speciális célból" nincs szükség 8 mA-nél nagyobb kimeneti áramra logikai 0-ban). A típusválaszték is rendkívül gazdag, az előbbieken említettük, hogy "újabb generációs" áramkörök (74LS100...LS200...LS300...LS400...stb.) már csak LS változatban (vagy, ha nagyobb áramra és sebességre van egyértelműen szükség, akkor S változatban) jelentek meg. Az LS áramkörök ára legtöbbször már alacsonyabb, mint a normál megfelelő típusé! Ne felejtsük el, hogy a körülbelül egyötöd résznyi tápteljesítmény igény nagy előny a tápegység költségei szempontjából is. (Ebből a szempontból természetesen a CMOS áramkörök a legelőnyösebbek - lásd ott!)

e) RSN54...RADIATION-HARDENED TTL, sugárzás ellen védett TTL

A teljesség kedvéért említjük meg. Radióaktív sugárzásnak kitett helyeken való felhasználásra készül az 54/74 változatoktól eltérő technológiával. Normál (RSN54), kisteljesítményű (RSN54L) és nagysebességű (RSN54H) sorozatai vannak. Típusválasztéka sokkal szerényebb, mint a többi változaté; kapuk, flip-flopok szerepelnek csak benne. Ára nagyon magas.

Érdeemes megjegyezni, hogy a hatféle TTL változaton kívül egyéb módosulatok is léteznek. Az európai TEXAS INSTRUMENTS gyártja pl. az SN84...-es sorozatot kiterjesztett ipari hőmérséklettartományra (-20...+80 °C), gyártanak ezenkívül speciális MSI típusokat SN49... jelöléssel stb.

f) A TTL változatok összehasonlítása

Ha a különböző sorozatok legfontosabb specifikációs adatait - a jelterjedési időt és a teljesítmény disszipációt - összehasonlítjuk, akkor láthatjuk a tendenciát: a "régebbi" típusoknál egyértelműen ellentétes összefüggés van a két alapjellemező között: a kisebb disszipációjú áramkör késleltetése nagyobb (54/74L), a kisebb késleltetési idő ára viszont nagyobb a disszipáció (54/74, 54/74H). Lényeges javulást hozott a Schottky-technológia: a nagysebességű áramkörök között egy 54/74S áramkör fele akkora késleltetési idővel és kevesebb teljesítmény igényel dolgozik, mint a régebbi nagysebességű 54/74H változat. A "normál" sebességű 54/74 LS változat viszont a régebbi normál TTL-nél lényegesen kisebb teljesítményt igényel, vagyis a "jóságra" jellemző $t_{pd} \cdot P_D$ mennyiség ezekre sokkal kisebb, kedvezőbb. A táblázatban összefoglaltuk a legfontosabb (átlagos) jellemzőket, így a számszerű összehasonlítást is megtehetjük:

SOROZAT	KÉSL.IDŐ-TELJ. SZORZAT p.Joule	JELTERJ. KÉSL.IDŐ ns	TELJ. DISSZIP. mW	FLIP-FLOPOK MAX.ÓRA FREKV. MHz
54 LS/74 LS	19	9,5	2	45
54 L/74 L	33	33	1	3
54 S/74 S	57	3	19	125
54/74	100	10	10	35
54 H/74 H	132	6	22	50

A táblázat alapján is belátható, hogy - kellő sebesség biztosításával - az LS változat a leginkább gazdaságos ($t_{pd} \cdot P_D = 19$ pJoule).

g) Különböző TTL változatok illesztése

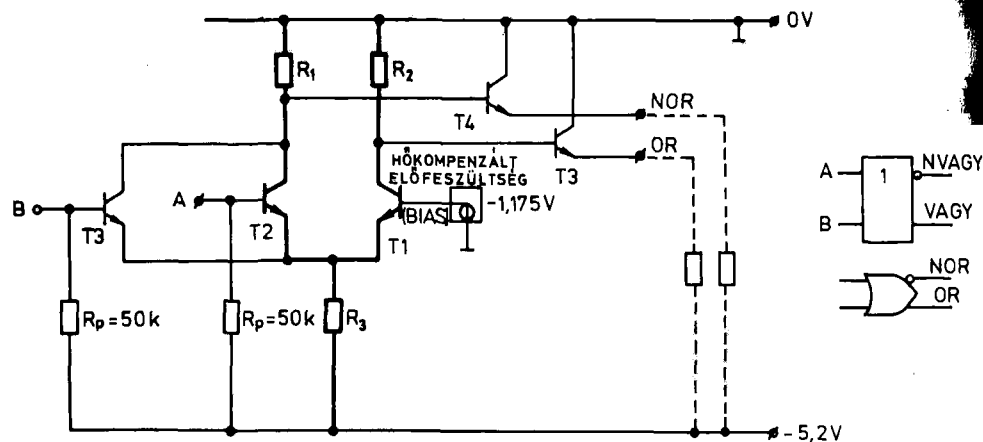
Gyakran előfordul, hogy egyetlen rendszerben többféle TTL áramkör működik. Azokban az egységekben, amelyeknek a sebessége kritikus, "H" vagy "S" változat van, az "igénytelen" helyekre "normál" vagy "LS" áramköröket építenek be. A különböző típusú áramkörök egy-egy ponton összekapcsolódnak egymással, ilyenkor merül fel az illeszthetőség kérdése. A TTL áramkörök egymással kompatibilisek, a tápfeszültség a logikai feszültség szintek, a zajtartalék gyakorlatilag egyenlők (max 0,1 V-os eltérések vannak), a bemeneti áramok és a kimeneti áramterhelhetőség értékek azonban nagyon eltérők. Nyilvánvaló, hogy az "L" vagy "LS" változat bemeneti árama és végerősítő fokozatának áramterhelhetősége lényegesen kisebb, mint az "S" vagy "H" változaté. Egy-egy rendszerre a kimeneti terhelhetőség (FAN-OUT) általában $N = 10$, de különböző rendszerek összeillesztésekor természetesen nem érvényes ez a szám. Az, hogy egy adott áramkör kimenetére hány más típusú áramkör bemenet kapcsolható, végső soron az áramkörök katalógus lapjairól olvasható le.

Alapvetően fontos, hogy a terhelő áramkörök bemeneti áramának összege logikai 1-ben és logikai 0-ban ne haladja meg a meghajtó áramkör megengedett logikai 1, ill. logikai 0 áramterhelhetőségét. A következő táblázatban összefoglaljuk a TTL változatok maximális bemeneti és megengedett kimeneti áramértékeit. A kimeneti áramok az $N = 10$ -re vonatkoznak, de lehetnek ettől eltérő értékek is (ezért minden esetben ajánlott a katalógus lapot tanulmányozni).

Sorozat	MAX BEMENETI ÁRAM		MAX KIMENETI ÁRAM	
	LOG 0-ban mA	LOG 1-ben μ A	LOG 0-ban mA	LOG 1-ben μ A
54/74	-1,6	40	-16	400
54 H/74 H	-2	50	-20	500
54 L/74 L	-0,18	10	-3,6	200
54 LS/74 LS	-0,36	20	-8	400
54 S/74 S	-2	50	-20	1000

3.5.4. ECL, Emitter Coupled Logic, emittercsatolt logika

Szokásos a CML (Current Mode Logic: áram-logika) elnevezés is. Kimondottan nagy sebességű bipoláris integrált áramkör. Felépítése olyan, hogy a benne levő nagy határfrekvenciájú tranzisztorok nem vezérlődnek telítésbe a működés során, ezért nem lép fel töltéstárolási jelenség. Többféle ECL áramkör változat van forgalomban, de a működési alapelv azonos, amelyet célszerű a legjobban elterjedt 10000-es ("10k", pl. MOTOROLA MECL 10000) sorozat alap-kapu áramkörén tanulmányozni. A 10101 négyes OR-NOR kapu egyszerűsített, "belső" kapcsolását mutatja a 3.74. ábra.

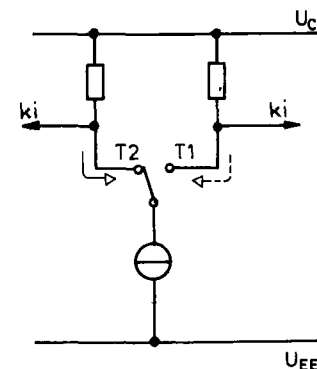


Bemeneti:		Kimeneti:	
Logikai 1	→ -0,75V	Logikai 1	→ -0,60V
Logikai 0	→ -1,60V	Logikai 0	→ -1,75V

3.74. ábra

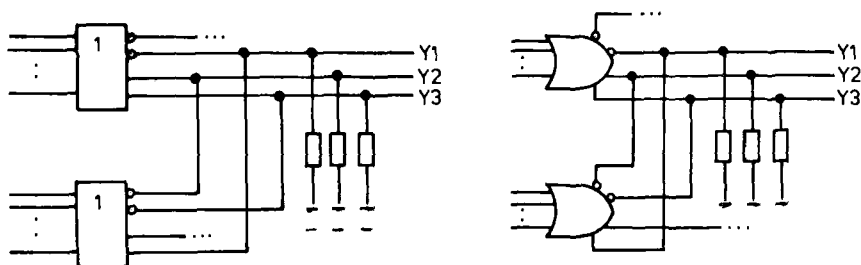
Az elrendezés NPN tranzisztoros differenciál erősítőhöz hasonló (lásd T1-et és T2-t), táplálásának legegyszerűbb, szokásos módja az, hogy a "pozitív pólus" a 0 V, ($U_{CC} = 0$ V, föld) ehhez képest az emitter oldal negatív, -5,2 V-os ($U_{EE} = -5,2$ V) feszültséget kap. A T1 tranzisztor bázisa rögzített előfeszült-

ségen van (BIAS), amelyet az áramkörön belül nagy pontossággal, hőkompenzáltan állítanak elő, értéke $U_{BB} = -1,175$ V. A bemeneti logikai jelszintek ettől néhány tizedes volttal térnek el pozitív-negatív irányban. Ha az A bemenetre, T2 bázisára logikai 1 szintet adunk (körülbelül 0,4 V-tal magasabb feszültséget, mint amit T1 bázisa kap, azaz kb. -0,75 V-ot), akkor az R3 emitterellenállás által szolgáltatott áramot T2 "veszi át", T1 lezár. A munkaellenállásokat és az emitterellenállást úgy méretezték, hogy a nyitott tranzisztor nem mehet telítésbe. Jelen esetben a nyitott T2 kollektor potenciálja körülbelül -0,85...-0,9 V-ra "jön le" (ekkora feszültség jön létre R_1 -en az átfolyó kollektoráram hatására). Erre a pontra csatlakozik a kimeneti szinteltoló és kis kimeneti impedanciát előállító T4 emitterkövető, ennek emitterén, vagyis a kapu (NOR) kimenetén még 0,75 V-tal negatívabb feszültség áll elő (minuszra kötött "lehúzó" terhelőellenállást feltételezve), azaz a logikai 0-nak megfelelő -1,6 V. Ha az A bemenetre logikai 0 szintet -1,6 V-ot adunk, akkor T2 - bázisa negatívabb lévén T1 bázisánál - lezár, T1 emitterén, a NOR kimeneten a logikai 1-nek megfelelő, körülbelül -0,75 V jön létre. Mivel T1 vezeti az áramot, most ennek kollektorpotenciálja csökken és a T3-mal felépített emitterkövető szinteltoló hatására az "ellenütemű" OR kimeneten a logikai 0-nak megfelelő -1,6 V áll elő. Az ECL kapu típusok közül nagyon soknál kivezetik mindkét differenciál erősítő kimenetet (a hozzájuk tartozó emitterkövető közbeiktatásával), így a ponált-negált kimenet is rendelkezésre áll.



3.75. ábra

ellenállást teszünk. A különböző rendszerek típusválasztékában vannak olyan kapuk, amelyeket több emitterkövető kimenettel készítenek, így ugyanaz a jel (ill. a negáltja) több helyre is elvihető wired függvény létrehozására (így jön létre az ún. "emitter-follower logic", EFL). A nagyintegráltságú ECL áramkörökben is így hozzák létre a logikai kapcsolatokat (3.77. ábra). Az LSI áramkörök létrehozása nehezebb a nagy teljesítmény disszipáció miatt: az ECL LSI tokokon sokszor hűtőbordát, "hűtő-kéményt" találunk. Sok IC-t tartalmazó rendszerekben kényszer-hűtésről is gondoskodni kell.



3.77. ábra

A típusválasztékban kapukat, "kész" kombinációs áramköröket és sorrendi alapelemeket, de - amint említettük - nagymértékben integrált LSI elemeket is találunk (mikroproceszorok is léteznek nagyon gyors ECL-ből ún. bit-szeletes (bit-slice) kivitelben). "Újabb generációs" ECL, speciális áramkörök is léteznek ma már, amelyek a nanoszekundumos tartomány alatt dolgoznak ($n \cdot 10$ ps késleltetési idővel, GHz-es órajel frekvenciával!).

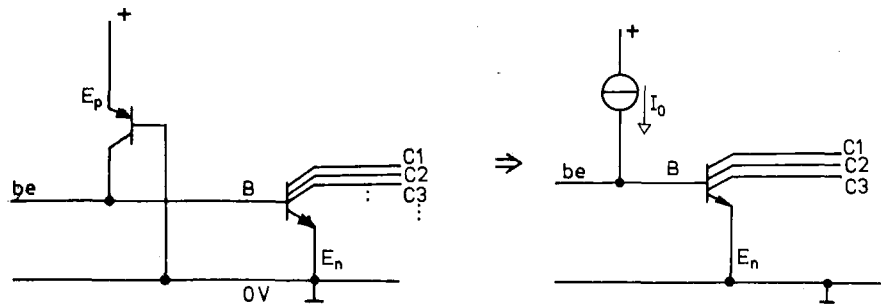
3.5.5. Újabb bipoláris áramkörök: I^2L (Integrated Injection Logic: integrált injekciós logika)

A bipoláris integrált áramkörök fejlesztése nem lezárt folyamat. A mai fő irány a teljesítményfelvétel lehető legnagyobb mértékű csökkentése úgy, hogy a sebesség jellemzők ne romoljanak, inkább javuljanak ($P_D \cdot t_{pd}$ kisebb legyen). Ez fő-

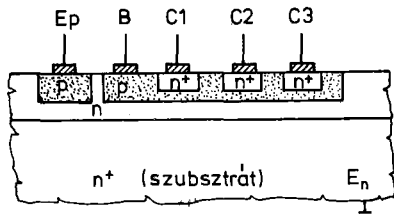
leg az egyetlen IC-lapkán integrálható elem-szám növelése szempontjából fontos (ahogy már említettük, bipoláris áramkörök esetében ennek főleg a sok "összeszűfolt" áramkör nagy disszipációja és az ezáltal létrejövő túlmelegedés szab határt). Jó ideig a következő fejezetben ismertetésre kerülő MOS áramkörök jöhettek csak szóba, ha igen nagy rendszert integráltak le, LSI áramkört készítettek. Néhány éve a MOS-LSI egyeduralkodó helyzete megszűnt, a bipoláris LSI komoly vetélytárs lett (önmagában azzal, hogy LSI, párosulva azzal, hogy sebessége sokkal nagyobb). Újabb technológiákkal (oxid-izoláció: oxid-réteg szigetelés, ion-implantáció: ion "belövéses" félvezető adalékolás) TTL, sőt ECL "óriás" áramköröket is készítenek egészen kis teljesítmény disszipáció - késleltetési idő szorzattal (TTL LSI: 3 pJoule, ECL: 1 pJoule), ns vagy az alatti késleltetési idővel.

Mézőben új és minden hagyománnyal szakító integrált áramkörös logikai rendszer az INTEGRATED INJECTION LOGIC (I^2L , I^2L). Ebben azt a régi törekvést sikerült megvalósítani, hogy egy logikai kapu gyakorlatilag egyetlen tranzisztorból álljon és így az integrálási sűrűség az előző rendszerekhez képest, nagymértékben megnövekedjék (több ezer, akár tízezres nagyságrendű kapu egyetlen lemezkén). Az I^2L kimondottan LSI, nagymértékben integrált áramkör változatban készül, különálló építőelemeket (kapukat stb.) nem gyártanak. A működési elv, a "tervezési filozófia" is erősen eltér a hagyományoktól: az integrált "kapuk" egy-bemenetűek és több-kimenetűek, a logikai kapcsolatot az "open collector"-os kimenetek összekötésével hozzák létre. Az "elemi" I^2L kapu csak integrált áramkörben képzelhető el, diszkrét elemekkel csupán közelítő helyettesítőképet lehet rajzolni, ezt a 3.78. ábrán láthatjuk. A működés megértéséhez a félvezető szerkezet ismerete is szükséges (egy "nagy IC" egy elemi részletének, egy kapujának a metszetét a 3.79. ábra mutatja).

A PNP tranzisztor, amelynek emittora a pozitív (néhány tízed voltos) tápfeszültségre kapcsolódik, mint egy áramgenerátor, bázisáramot injektál (innen a rendszer neve) az NPN több kollektoros tranzisztor bázisába. A metszeti ábrán a bal olda-



3.78. ábra

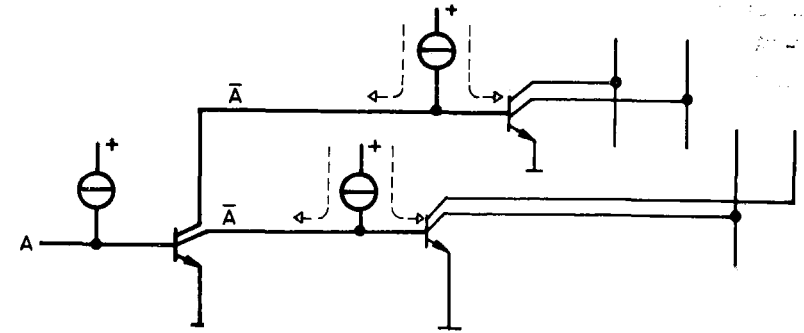


3.79. ábra

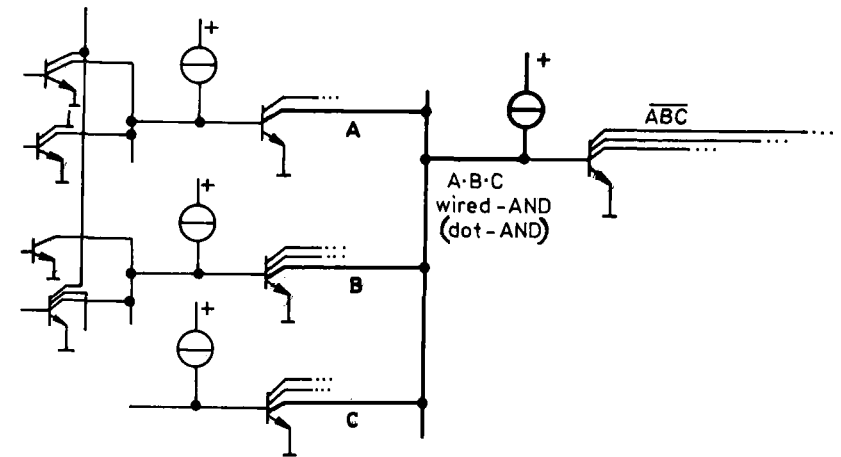
lon rajzolt P rétegből indulnak el a töltéshordozók és a nyitott laterális PNP tranzisztor kollektor P-rétegébe injektálódnak, mely utóbbi P-réteg a "logikai kapu" NPN tranzisztorának bázis rétege. Az injektált bázisáram miatt az NPN tranzisztor kinyit és kollektorai (!) gyakorlatilag földpotenciálra kerülnek - ugyanugy, mint egy inverter esetében. A kollektorok "munkaellenállásai" az ezekhez csatlakozó ugyanilyen szerkezetű I²L kapuk bemeneti PNP áramgenerátorai, amelyek "igyekeznek felhuzni" a kollektor potenciált. Azonban, ha a meghajtó kollektor földpotenciálon van, akkor a rákapcsolódó következő fokozat áramgenerátorának árama ebbe a kollektorba, folyik és nem nyitja ki a hozzá tartozó NPN tranzisztort (3.80. ábra).

A logikai kapcsolatok a megfelelő "open collector"-ok összekötésével jönnek létre, úgy mint a wired-AND logikában. Az összekötött kollektorok munkaellenállása az összekötési pontokra csatlakozó következő I²L kapu áramgenerátora (3.81. áb-

ra). Az I²L LSI áramkörben egy-egy kapunak annyi kimenete kollektora van, ahány helyen fel kell használni a kapu által előállított függvényt.



3.80. ábra



3.81. ábra

Az integrált injekciós logikának nagyon sok olyan előnye van, amely még sohasem teljesült egyszerre más áramkörök esetében. Legfontosabbak:

- Egy I²L kapu helyigénye egy IC lemezkén minden eddiginél kisebb.

- A teljesítmény disszipáció-késleltetési idő szorzat minden eddiginél kisebb (0,01...0,001 pJoule is lehetséges).

- Bipoláris rendszer lévén, sebessége nagyobb lehet a MOS rendszerekénél. Az I²L elrendezést a Schottky technológiával kombinálva 5 ns és az alatti késleltetés érhető el, de MOS-hoz hasonló kis disszipációval.

- Az injektor-áram (tápfeszültség), ill. integráláskor a tranzisztorok alakzatának megválasztásával, módosításával a sebesség és disszipáció jellemzőket több nagyságrenden belül be lehet állítani (ha nem kell gyors működés, akkor a disszipációt lehet csökkenteni, nagy sebesség igényt az áramok növelésével lehet kielégíteni). Akár egyetlen áramkörben is lehetőség van különböző sebességű és disszipációjú zónák kialakítására, csupán a "tranzisztorok" geometriájának változtatásával.

- Az I²L rendszer bipoláris lévén, tetszőlegesen kombinálható egyéb analóg vagy digitális áramkörökkel egyetlen integrált egységben (pl. analóg-digitál, digitál-analóg átalakítók, digitális voltmérők, nagyáramu beavatkozók, interface-ek, híradástechnikai analóg-digitális berendezések stb.).

- Azoknak a gyáraknak, melyeknek bipoláris technológiához meglevő berendezéseik, tapasztalataik vannak, az I²L bevezetése nem igényel gyökeres átalakítást.

3.6. MOS és CMOS áramkörök

3.6.1. MOS áramkörök

A MOS kifejezetten a nagyintegráltságú (LSI) kisműködésű, de nem túl gyors áramkörök készítésére alkalmazható technológia (átlag 0,2...0,5 W teljesítményfelvétel egy teljes LSI áramkörre, 100...500 ns késleltetés, ill. ciklusidő).

A MOSFET-tel felépített áramkörök előnyeit a következőkben foglalhatjuk össze:

- Kis teljesítmény fogyasztás, ami részben lehetővé teszi, hogy teljes rendszert (kalkulátor, számítógép-központi egység, memóriák stb.) egyetlen lemezkére integráljanak.

- Egy MOS tranzisztor kis helyfoglalása az integrált áramkör lemezkén (kb. harmada a bipoláris tranzisztor helyfoglalásának), ami szintén elősegíti a nagy komplexitású áramkörök előállítását. Az ellenállásokat is passzív MOS tranzisztorral valószínűsítjük meg (ahogy ezt a MOS inverterek tárgyalásakor már láttuk), így sokkal kisebb helyen elférnek.

- Villamos szempontból a növekményes MOS tranzisztor különösen alkalmas digitális áramkörök előállítására, mert:

a) alaphelyzetben kikapcsolt állapotban van (ha $U_{GS} = 0$ V, a tranzisztor lezár) - alapállapotban bekapcsolt elemekkel nagyon bonyolult módon lehet csak digitális áramkört építeni - ez az oka annak, hogy pl. réteg-FET-eket szinte sohasem használnak logikai áramkörökhöz;

b) nagy bemeneti ellenállása van: $10^{14} \Omega$ nagyságrendben, ami gyakorlatilag szakadásnak számít, a bemenet kapacitív. Ezért a MOS áramkörök DC fan-out-ja nagyon nagy;

c) "emlékező képessége" van; a GATE kapacitás egyszeri feltöltésével MOSFET bizonyos ideig (technológiától függően ms-ektől évekig) nyitott állapotban tartható. Ez a tulajdonság rendkívül egyszerű áramköri megoldásokat tesz lehetővé;

d) mindkét irányban szimmetrikus: a SOURCE (E) és a DRAIN (C) elvileg felcserélhető, ami lehetővé teszi, hogy a MOSFET-et pl. egy kapacitás töltésére és kisütésére egyaránt felhasználjuk.

- Az integrált áramkör lemezkében a MOS tranzisztorok automatikusan elszigetelődnek egymástól anélkül, hogy a bipoláris rendszerekénél szokásos szigetelés-diffúzióra szükség volna, ami szintén csökkenti a helyigényt.

Főbb hátrányok:

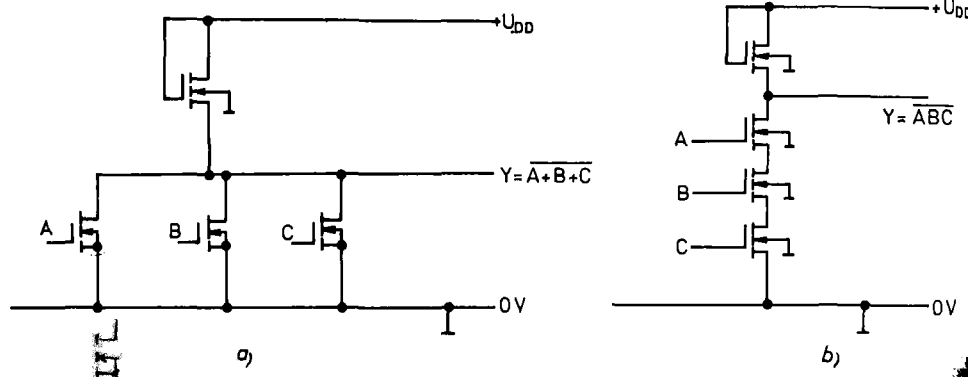
- A MOS rendszerek a nagy impedanciák és a kapacitások miatt lassabbak a mai bipoláris rendszerekénél. A fogyasztás ugyan kicsi, de a teljesítmény-jelterjedési idő szorzat sokkal nagyobb, mint a bipoláris rendszereké (a fejlődés természetesen nem állt meg).

- Más logikai áramkörökkel legtöbbször nem kapcsolhatók össze közvetlenül, csak külön szintáttévek, illesztők (inter-

face-ek) közbeiktatásával. Az "ujabb generációs" típusok már rendszerint közvetlenül illeszkednek az LS TTL-hez.

- "Régebbi" áramkörök többféle tápfeszültséget is igényelnek, ma már egyre több a +5 V-os tápfeszültségről működő típus.

A MOS inverterek működésével az előzőekben már megismerekedhettünk, most a kombinációs hálózatokat alkotó kapu áramköröket vegyük szemügyre. Meg kell jegyezni, hogy a következő kapu áramkörök többnyire csak az LSI áramkörök kis részleteként fordulnak elő, külön, univerzális célra SSI kapu áramköröket általában nem gyártanak, legfeljebb csak egy-két típust - főleg modellezéshez és ún. CUSTOM LSI (a vevő által a gyárnak megadott LSI) áramkörök kikísérletezéséhez. A kapuk a statikus MOS áramkörök csoportjába tartoznak, ami annyit jelent, hogy nem igényelnek egy- vagy többfázisú órajelet és egyenfeszültség szintekkel is működnek (szemben a dinamikus - általában sorrendi - áramkörökkel, amelyeknek van egy alsó határfrekvenciájuk, ez alatt nem működnek).



3.82. ábra

A legegyszerűbb NOR kapu változatot mutatja a 3.82a. ábra. Ha bármelyik bemenet logikai 1 feszültségen van (a tápfeszültséghez közel eső feszültségen), akkor a megfelelő tranzisztor kinyit és a kimenet 0 V körüli, azaz logikai 0 lesz. A kimeneten tehát akkor van logikai 0, ha bármelyik bemenet (vagy egyszerre több) logikai 1-en van, a kimeneti függvény a

tagadott-VAGY, NOR. A felső MOS tranzisztor - mint tudjuk - a munkaellenállás szerepét tölti be (100 kΩ nagyságrendű csatorna ellenállással).

A NAND áramkör a b. ábra szerinti. A kimenet akkor lesz logikai 0 feszültségen, ha a "tranzisztorlánc" mindegyik eleme rövidzár, vagyis ha valamennyi bemeneten logikai 1 feszültség van. Ha akármelyik bemenet logikai 0-án van, akkor a megfelelő tranzisztor lezár és a munkaellenállás (MOS) "felhúzza" a kimenetet logikai 1-re.

A MOS sorrendi áramköröket a későbbiekben a megfelelő témakörökkel kapcsolatban ismerjük majd meg.

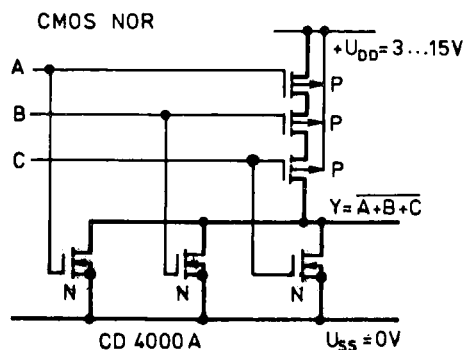
Fontos tudnunk, hogy a MOS áramkörök bemenetei érzékenyek a túlfeszültséggel szemben; a megengedettnél nagyobb GATE feszültség miatt átüthet a GATE alatti vékony oxidréteg. Ez akkor is megtörténhet, ha egy szabadon maradt nagy impedanciájú bemenetet kézzel megérintünk (nagy statikus töltést, vagy 50 Hz-es feszültséget vihetünk rá), amire szereléskor, gyártáskor tekintettel kell lenni. A mai, újabb típusok bemeneteire 100 kΩ körüli értékű "lehúzó" sönt ellenállásokat tesznek, amelyek megvédik az áramköröket a kézzel rávihető "véletlen" töltésektől. Megfelelő elővigyázat azért ezeknél is szükséges, forrasztáskor, szereléskor a páka, szerszámok a rendszer 0 V-jához földelendők!

3.6.2. Komplementer MOS (CMOS, COS-MOS, McMOS...) áramkörök

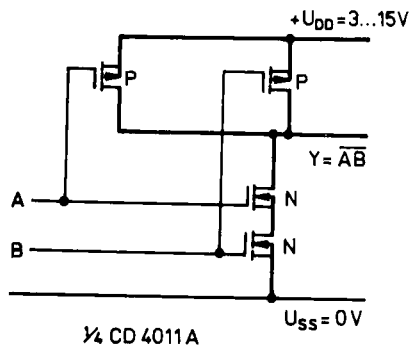
A működési elvet a CMOS inverterek tárgyalásakor a 3.4. fejezetben már megismertük, egyik helyzetben a felső, P-csatornás tranzisztor nyitott, és a kimenetet a pozitív tápfeszültséggel köti össze, a másik helyzetben az alsó, N-csatornás tranzisztor nyit ki és a kimenetet a 0 V-tal köti össze.

A CMOS univerzális áramköre a NOR-kapu, amelynek egyszerűsített rajza a 3.83. ábrán látható (RCA). A működés elve az inverteréhez hasonló; ha akármelyik bemenet logikai 1-ben van, akkor a megfelelő N-csatornás (alsó) tranzisztor kinyit és a kimenetet a 0 V-tal köti össze. Az ugyanazon bemenethez tar-

tozó P-csatornás tranzisztor viszont lezár és ezért a felső "munkaellenállás-lánc" megszakad - a kimeneti feszültség 0 V lesz. Csak ha mindegyik bemenet 0 V-on van, akkor nyit ki valamennyi P-csatornás tranzisztor és mivel valamennyi N-csatornás tranzisztor szakad, a kimeneti feszültség a pozitív tápfeszültséggel lesz gyakorlatilag egyenlő. Figyeljük meg, hogy a NOR kapuban az N-csatornás tranzisztorok az előző rendszerekben is "megszokott módon" párhuzamosan vannak kapcsolva, a komplementer P-csatornások viszont sorosan (a kapcsolásuk és a működésük is "komplementer").



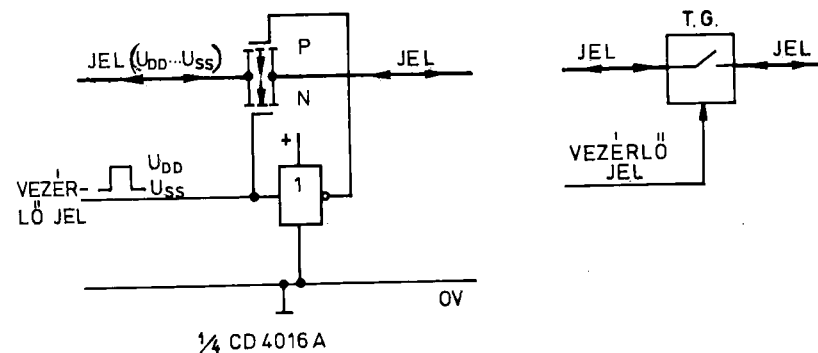
3.83. ábra



3.84. ábra

A NAND áramkörben (3.84. ábra) a helyzet fordított: az alsó, N-csatornás tranzisztorok soros kapcsolásban vannak, a P-csatornások párhuzamosan. Ha valamennyi bemenet egyidejűleg

logikai 1-en van (ÉS), akkor az N-csatornás tranzisztorlánc rövidzár, és valamennyi P-csatornás tranzisztor szakad, a kimenet logikai 0-án lesz (NEM-ÉS). Ha csak egyetlen egy bemenet is 0 V-on van, akkor már megszakad az N-lánc és a vele "párban levő" P-csatornás tranzisztor kinyit, ezért a kimenet logikai 1-en lesz. Egyik áramkörnél sem fordul elő, bármilyen vezérlés esetén sem, hogy az alsó és a felső oldal egyszerre rövidzárú vagy szakadássá válna (ezért ilyen bonyolultak az áramkörök).



3.85. ábra

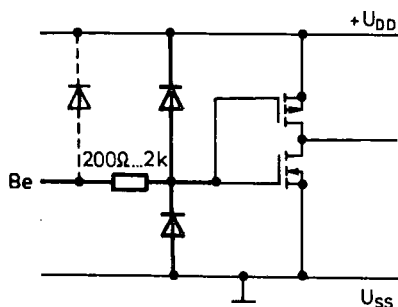
A CMOS rendszerben van még egy alap kapuáramkör: az ún. TRANSMISSION-GATE, "áteresztő-kapu". Ez végeredményben egy soros kapcsoló, melyet logikai szinttel lehet vezérelni (3.85. ábra). Ha a vezérlőjel logikai 1 (+ tápfeszültség), akkor ez közvetlenül kinyitja a kapcsoló N-csatornás tranzisztorát és egy inverteren keresztül (az inverter kimenetén ekkor 0 V van) kinyitja a párhuzamos P-csatornás kapcsoló tranzisztorát is. A jel - akár 0, akár 1 - a kapcsolón "akadálytalanul" átjuthat (a bekapcsolt ellenállás 200Ω , javított változatokban 80Ω körüli). Ha a vezérlőjel 0 V, akkor mindkét kapcsoló tranzisztor lezár (az N-csatornás 0 V-os, a P-csatornás $+U_{DD}$ vezérlőfeszültséget kap) a kapcsoló kikapcsol, ellenállása gyakorlatilag végtelenné válik, a jelet nem engedi át. A kapcsoló "bilaterális", kétirányú, azaz a kapcsolandó jelnek nincs kitüntetett iránya, nincs külön bemenet és kimenet (még ha ezt fel is tüntetik az IC bekötési rajzon).

A TRANSMISSION-GATE a CMOS kombinációs és sorrendi áramkörök egyik fontos alapeleme, de ugyanezt az elrendezést 0 V és $+U_{DD}$ közötti bármely analóg jel kapcsolására is fel lehet használni (ilyen analóg kapcsoló például a CD4016A, mely a 3.85. ábrán látható kapcsolóból 4 darabot tartalmaz egy tokban vagy pl. a CD4051A és a CD4052A, amelyek 8 csatornás kapcsolók).

A komplementer-MOS áramkörök villamos tulajdonságaival az inverter tárgyalásakor már ismerkedtünk, a különféle CMOS családok SSI-MSI típusaira (kapukat, kombinációs, alapvető sorrendi áramköröket tartalmazó IC. típusokra) is hasonlóan előnyök a jellemzők.

A főbb előnyök (amelyek a CMOS katalógusokban általában főcímben szerepelnek):

- Kis teljesítmény felvétel statikus üzemben (konstans vagy ritkán változó vezérléskor, kapunként 10 nW... μ W).
- Nagy zajtartalék (nagyobb tápfeszültségnél több V, "majdnem" a tápfeszültség fele).
- Üzembiztos működés széles tápfeszültség tartományban (egyes típusokra +3...+15 V).
- Kis kimeneti impedancia (típustól függő, de közhatalmál kisebb).
- Viszonylag gyors működés (gyorsabb a MOS-nál, de lassabb a legtöbb bipolárisnál, tápfeszültségtől és típustól függően n.10 ns).

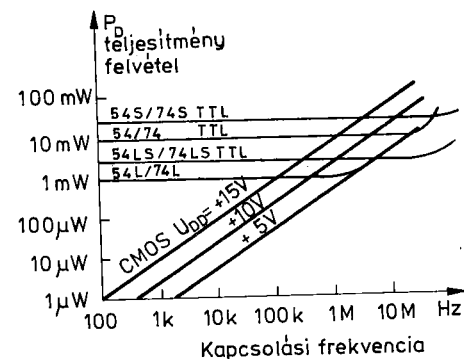


3.86. ábra

- Kezelésüket megkönnyíti, hogy a CMOS áramkörök bemenetein túlfeszültség védő ellenállás-dióda hálózat van (a leggyakoribb a 3.86. ábrán látható, bár vigyáznunk kell, mert ki-csatolt tápfeszültség esetén a gyártók legtöbbször nem szavatolják a védelmet, sőt általában az is tiltott, hogy bármely bemenetre jelfeszültséget adjunk - "feszültség generátorral"- a tápfeszültség bekapcsolása nélkül).

A főbb hátrányok (amelyeket a CMOS katalógusok ritkán emlitenek):

- A CMOS áramköröket nem statikus, hanem gyors ütemben felváltva 0-1 vezérléssel működtetjük, akkor az ismétlődési frekvencia növelésével lineárisan növekszik a teljesítmény felvétel (3.87. ábra). Látszik, hogy a CMOS fogyasztása 100 kHz felett elérheti, sőt meghaladhatja akár a bipoláris TTL áramkörök fogyasztását is.



3.87. ábra

- Egyes típusok kapuinak (pl. a 3.83. és 3.84. ábrán láthatóknak) a kimeneti ellenállása és emiatt a késleltetési ideje különböző vezérlések esetén mindig más és más lehet. Például a rajzon szereplő NOR kapu kimeneti ellenállása, ha a kimenet 0-ban van 1...1/3 FET csatorna ellenállás, logikai 1-ben viszont 3 FET csatorna ellenállás, ahogyan ez a 3.83. ábrából belátható, vagyis a kimeneti ellenállás 1:9 arányban változhat! Ez néha "titokzatos" hibáknak lehet forrása, hiszen a változó kimeneti ellenállás nem túl nagy kapacitív terhelés

esetén is erősen eltérő késleltetéseket eredményez különböző vezérlési állapotokban - különböző idejű hazárdokat okozva. Léteznek külön buffer végfokozattal ellátott típusok (l. később), ezek ebből a szempontból előnyösebbek.

- Sok olyan CMOS típus van, amely nem képes normál TTL bemeneteket meghajtani, csak külön buffer elemek közbeiktatásával. Az L változatokhoz általában elegendő a kimenő áramuk, az LS-hez már nem biztosan. Ezért minden esetben figyelemmel kell kísérni az éppen felhasznált típus adatlapját!

- "Tirisztor hatás" következhet be, ha egy CMOS kimenet feszültségét egy nagy amplitudójú zavar impulzus akár csak egy rövid pillanatra a tápfeszültség fölé viszi (a CMOS bonyolult IC-szerkezetében több PN átmenet van, ebből négyrétegű "tirisztor" alakul ki). Ha a "tirisztor" "begyújt", akkor a zavarfeszültség után is megmarad ez az állapot, ami az áramkör tönkremenetelét okozhatja.

- Bonyolultabb szerkezeti felépítésük miatt a CMOS áramkörök kissé drágábbak a hagyományos TTL-nél, de az árkülönbség csökkenőben van. Kárpótlást nyújt viszont a kisebb fogyasztás.

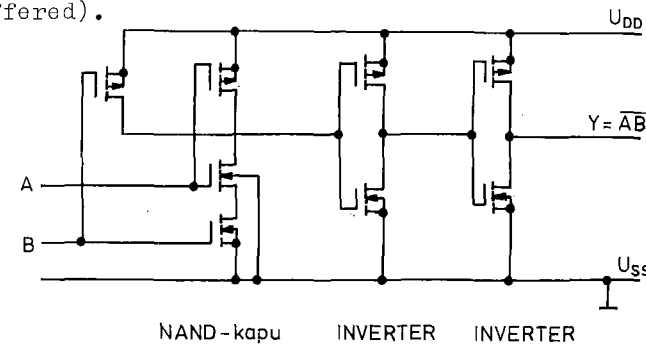
Az előbbiekből belátható, hogy a célnak megfelelő típus (család) kiválasztása nagy körültekintést (katalógus tanulmányozást) igényel. Ma már sokféle tipussorozat létezik, szinte "minden elképzelhető" célra gyártanak CMOS integrált áramköröket, SSI, MSI-, sőt egyre többször LSI típusokat, a CMOS ezidő szerint a legcélszerűbben alkalmazható univerzális digitális építőelem. A tájékozódás megkönnyítése érdekében összefoglaljuk az elterjedtebb tipussorozatokat (a villamos jellemzőket is megemlítjük, bár az összehasonlítás nagyon nehéz: ahány gyár, katalógus, annyiféle viszony között - tápfeszültség, terhelő kapacitás, stb. - adják meg ezeket):

- CD 4000 A sorozat (RCA). Ez volt időrendben a legelső, amely univerzális építőelemként felhasználható SSI, MSI elemeket tartalmazott. A kapuk "belső" kapcsolását bemutató rajzokon (3.83., 3.84., 3.85.) is ennek tipussorozatnak a tagjai szerepelnek (CD 4000 A részlete, a CD 4011 "negyede", valamint a CD 4016 "negyede"). Ma is az egyik legtöbbször alkalmazott áramkör-fajta. Típusválasztéka gazdag (100 fölötti ti-

pus). Nagyon kis fogyasztású (10 nW kapunként), de legtöbb típusának kimeneti terhelhetősége is kicsi, normál TTL-hez általában nem, legfeljebb az L változathoz illeszthető, egy néhány típus az LS-hez is, természetesen, +5 V-os tápfeszültség mellett. A terhelhetőséget minden esetben érdemes a katalógusból ellenőrizni (TTL szint-illesztéshez a CD4049-4050 buffer használható). A megengedett tápfeszültség-tartomány +3 V... ..+15 V (a 15 V-os maximumnál nem szabad nagyobbat ráadni), a statikus és dinamikus jellemzők nagyobb tápfeszültség esetében jobbak (10...15 V). Amit a kimeneti ellenállás vezérlési kombinációktól függő változásáról mondtunk, azt különösen a több bemenetű kapuk felhasználásakor kell figyelembe venni. Egy kapu jelterjedési késleltetési idejét a katalógus erre a típusra 5 V-nál 35 ns-nak, 10 V-nál 25 ns-nak adja meg, 15 pF terhelő kapacitás mellett. A kimeneti jel fel-lefutási ideje ennél hosszabb: 35...65 ns!

A CD 4000 A sorozat változó kimeneti ellenállásának hatását a

- CD 4000 B (Buffered: buffer erősítővel ellátott) változattal küszöbölték ki. A MOTOROLA cég MC 14000 B sorozatában a "buffered" változat a "természetes" (pl. az MC 14011 B megegyezik a CD 4011 A buffer erősítő változatával, kapcsolását a 3.88. ábra mutatja: a kapu belső felépítése azonos maradt, de kimeneti fokozatként egy komplementer végfokozatot invertert helyeztek el, ez elé még egyet azért, hogy a kimeneti függvény ponáltja jelenjen meg). Az említett sorozatban külön jelölik ha "nem bufferelt" valamely áramkör (pl. MC 14011 UB, UB: UnBuffered).



1/4 MC14011B

3.88. ábra

A bufferelt áramkörök előnye a kis és állandó kimeneti impedancián kívül a gyorsabb kimeneti jel fel-lefutás. A kapuk legfőbb jellemzőit a MOTOROLA cég katalógusa érdekesen, képletek segítségével adja meg, pl.:

Tápáram: 5 V-nál: $I_T = 0,3 \mu\text{A/kHz} \cdot f + I_{DD}/N$
10 V-nál: $I_T = 0,6 \mu\text{A/kHz} \cdot f + I_{DD}/N$
15 V-nál: $I_T = 0,9 \mu\text{A/kHz} \cdot f + I_{DD}/N$
ahol: f : a bemeneti jelfrekvencia
 I_{DD} : a statikus tápáram
 N : egy tokban lévő kapuk száma

Kimeneti jel fel-lefutási idő (C_L a terhelő kapacitás):

5 V-nál: $t_r = t_f = 1,35 \text{ ns/pF} \cdot C_L + 33 \text{ ns}$,
tipikusan: 100 ns ($C_L = 50 \text{ p}$)
10 V-nál: $t_r = t_f = 0,6 \text{ ns/pF} \cdot C_L + 20 \text{ ns}$,
tipikusan: 50 ns ($C_L = 50 \text{ p}$)
15 V-nál: $t_r = t_f = 0,4 \text{ ns/pF} \cdot C_L + 20 \text{ ns}$,
tipikusan: 40 ns ($C_L = 50 \text{ p}$).

Jelterjedési késleltetési idő (az MC 14011 B-re):

5 V-nál $t_{pd} = 0,9 \text{ ns/pF} \cdot C_L + 80 \text{ ns}$,
tipikusan 125 ns ($C_L = 50 \text{ p}$)
10 V-nál $t_{pd} = 0,36 \text{ ns/pF} \cdot C_L + 32 \text{ ns}$,
tipikusan 50 ns ($C_L = 50 \text{ p}$)
15 V-nál $t_{pd} = 0,26 \text{ ns/pF} \cdot C_L + 27 \text{ ns}$,
tipikusan 40 ns ($C_L = 50 \text{ p}$).

Hasonló buffer végfokozattal ellátott változatokat más cégek is készítenek (pl. a FAIRCHILD 34000 sorozat, ebben a 3-as szám után lévő számjegyek a 4000-es sorozat "eredeti" típuszámából származnak).

- MC 14500-as sorozat. Ezek is a MOTOROLA-cég gyártmányai, főleg B változatban készülnek. Típusválasztéka rendkívül gazdag, az MC 14000 B (CD 4000 B) sorozatot "egészíti ki" egy sor SSI, MSI, LSI típussal (kapuktól a digitális sebesség

szabályozóig, TTL-CMOS, CMOS-TTL szint-áttevőktől a füst-detektorig tartalmaz CMOS áramköröket). Újabb sorozat indul MC 14 1000-tól: mikroprocesszorok, egyetlen IC-ben lévő mikro-számítógépek, memóriák és egyéb MSI, LSI elemek vannak a típusválasztékban.

- MM 54/74 C sorozat. A számjelzés a TTL 54/74 sorozatra utal. E család villamos paramétereit tekintve nem különbözik lényegesen a CD 4000-es sorozattól (3...15 V tápfeszültség, -5 V-nál 50 ns, 10 V-nál 25 ns körüli késleltetés). Érdekessége, hogy funkciójában és (egy-két kivételtől eltekintve) láb elrendezésében megegyezik hasonló számjelzésű normál TTL "megfelelőjével". Aki már dolgozott TTL-el, megismerte típusválasztékát, annak semmilyen gondot sem okoz a "C" jelzésű CMOS áramkörre való áttérés. (A 7400 CMOS megfelelője a 74C00 és így tovább.) Ma már a TTL típusválaszték nagyrésznének CMOS "párját" gyártja a NATIONAL SEMICONDUCTOR, emiatt nagyon népszerű az 54C/74C sorozat. TTL áramkörökkel "keverve" is lehet alkalmazni, de figyelembe kell vennünk, hogy a 74 C áramkörök kimenő huzóáramának legnagyobb értéke 0,36 mA lehet, ami 2 kisteljesítményű TTL (74 L...) vagy 1 kisteljesítményű Schottky TTL (74 LS...) bemenet meghajtásához elegendő (ezt viszont minden egyes típusra szavatolják).

- HCMOS (High speed CMOS) nagysebességű CMOS áramkörök megjelenése az utóbbi évek fejlesztésének eredménye. A fő cél a bipoláris áramkörök sebességének megközelítése, elérése az egyéb előnyök megtartása mellett. Az "új generációs" CMOS családokra példa a már piacon lévő 54HC/74HC sorozat 10 ns jelterjedési késleltetéssel, amely megegyezik az 54LS/74LS TTL áramkörök késleltetésével (!). Ez a sebesség a CMOS rendszer számos előnyével párosul: nagy zajtartalék, széles tápfeszültség-tartomány, és főleg extrém kis teljesítmény disszipáció. (A tirisztor-hatást is sikerült minimumra csökkenteni.) Az 54HC/74HC család áramkör típusai láb elrendezésben egyeznek LS ekvivalensükkel, ráadásul kimenetükön a CMOS-nál eddig szokatlan, 4 mA nagyságú kimeneti huzó- és forrásáram kiadására képesek. A National gyártja már a népszerű 4000-es sorozat egyes tagjainak nagysebességű változatát is (MM74HC4000). A nagysebességű CMOS-nak TTL-hez közvetlenül illeszkedő változata is van (MM 54 HCT/74 HCT).

A CMOS áramkörök felhasználásakor sohasem szabad megfelelkezni a nem használt "felesleges" bemenetekről (TTL-nél egy bemenet üresen hagyása 1-es vezérléssel egyenértékű, bár nem ajánlott módszer). A MOS-CMOS áramkörök bemenete igen nagy impedanciás, ha üresen hagynánk, potenciálja bizonytalan lenne, külső, "véletlen" ráhatások dönthetnek el, hogy 0 vagy 1 szintre kerülne. Ezért a nem használt bemeneteket minden esetben a 0 V-hoz vagy a pozitív U_{DD} tápfeszültséghez kell kötnünk az áramkör igazságtáblázatától függően úgy, hogy "ne szölgjön bele" a működésbe (pl. NAND szabad bemenetet 1-re, NOR bemenetet 0-ra). A bemenetek párhuzamos kötése DC terhelési problémát nem okoz, csak a bemeneti kapacitást növeli, ami sebesség csökkenéshez vezet. Ha a sebesség kritikus, akkor ehhez ne folyamodjunk.

A kimenet rövidrezárása (0 V-hoz vagy $+U_{DD}$ -hez) a MOS és CMOS áramkörökre a legtöbb esetben nem jelent "közvetlen életveszélyt" (a csatorna ellenállás korlátozza az áramot), baj akkor történhet, ha emiatt az IC tokra megengedett disszipációt tartósan túllépjük. A tápfeszültség fordított rákapcsolása (pl. az IC 180°-kal elforgatva foglalatba helyezése miatt) az áramkör azonnali tönkremenetelét (örök villamos mezőkre szenderülését) okozhatja.

3.7. Az áramköröcsaládok összehasonlítása

Az áramköröcsaládok áttekintése után, célszerű a legfontosabb paraméterek alapján összehasonlítást tenni közöttük azért, hogy az adott feladathoz a leginkább megfelelő típus kiválasztása ne okozzon nagy nehézséget.

A sebesség szempontjából sorrendbe állítva az áramköröket a lassubbaktól a gyorsabbak felé (egy kapunak megfelelő t_{pd} csökkenése irányában) a következő közelítő elrendezést írhatjuk fel:

- "passzív" (erősítőt nem tartalmazó, szokásos terhelések, vezetékhozzak között működő) logikák, a DDL áramkörök (μs),

- ("Régi" bipoláris áramkörök: RCTL, RTL (μs),

- MOS áramkörök (LSI),
"régebbi" PMOS ($< 1 \mu s$),
"mai" NMOS ($1 \mu s \dots 100 ns$),
- bipoláris I^2L áramkörök kisebb fogyasztású változatai ($< 100 ns$),
- bipoláris TTL L változat (33 ns),
- CMOS CD 4000, 74 C sorozat (30 ns),
- bipoláris TTL Normál változat (10 ns),
- CMOS 74 HC nagysebességű változat (10 ns)!,
- bipoláris TTL LS változat (9,5 ns),
- bipoláris TTL H változat (6 ns),
- bipoláris Schottky I^2L (5 ns),
- bipoláris TTL S változat (3 ns),
- bipoláris ECL 10k (2 ns),
- speciális ECL áramkörök ($< 1 ns$).

A teljesítmény disszipáció szerinti közelítő sorrend (nagyobb disszipációtól a kisebb felé haladva, az egy kapura eső disszipáció átlagértékét alapul véve):

- speciális, gyors ECL n·10 mW,
- bipoláris ECL 10k 25...50 mW,
- bipoláris TTL H változat 22 mW,
- bipoláris TTL S változat 19 mW,
- bipoláris TTL Normál változat 10 mW,
- CMOS 1 MHz-en 10 mW!,
- (régi RTL) (5 mW),
- bipoláris TTL LS változat 2 mW,
- bipoláris TTL L változat 1 mW,
- (régi RCTL) 1 mW,
- CMOS 100 kHz-en 1 mW,
- MOS, "régebbi" PMOS LSI egy kapuja 1 mW,
- MOS, "mai" NMOS LSI egy kapuja $< 1 mW$,
- CMOS 1 kHz-en 100 μW ,
- bipoláris I^2L és Schottky I^2L n·100 nW,
- CMOS DC üzemben 10 nW.

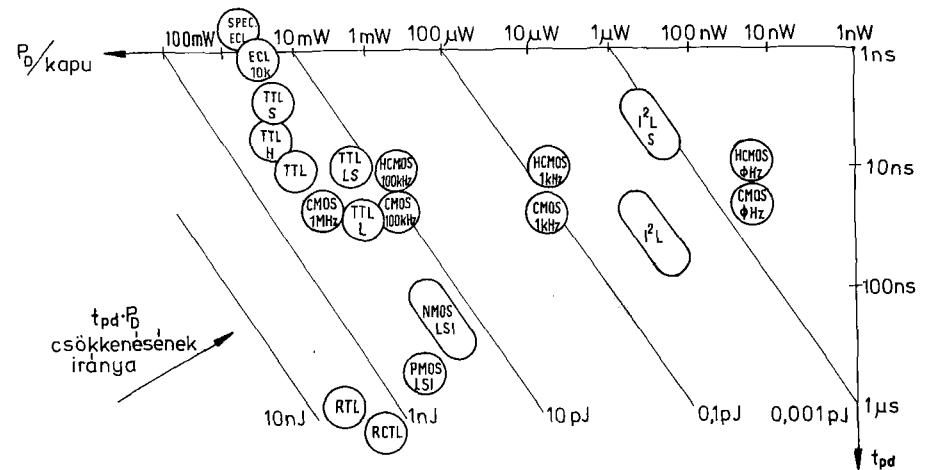
A jelterjedési késleltetési idő–teljesítmény disszipáció szorzatban a következő a hozzávetőleges sorrend:

- "Régi" RTL	> 1 nJ,
RCTL	
- MOS: "régi" PMOS	100...150 pJ,
- bipoláris TTL H	132 pJ,
- bipoláris TTL Normál	100 pJ,
- bipoláris TTL S	57 pJ,
- bipoláris ECL lok	50 pJ,
- MOS: "mai" NMOS	n·10 pJ,
- bipoláris TTL L	33 pJ,
- bipoláris TTL LS	19 pJ,
- CMOS 100 kHz-en	> 10 pJ,
- CMOS 1 kHz-en	0,1 pJ,
- bipoláris I ² L	0,01 pJ,
- bipoláris Schottky I ² L	0,001 pJ,
- CMOS DC üzemben	< 0,001 pJ,

Az előbbi jellemzőkből látszik a már említett tendencia, a nagyobb sebesség ára általában a nagyobb teljesítmény disszipáció: a késleltetési idő sorban a kis t_{pd} irányában levő áramkörök a teljesítmény disszipáció felsorolásában az elsők között vannak. Ez a "fordított" sorrend csak az igen jó, modern áramkörök esetében "borul fel". A három jellemző alapján az áramköri rendszerek közelítő "besorolását" a 3.89. ábra vázlatos diagramján tekinthetjük át. A diagram a teljesítmény disszipáció és jelterjedési késleltetési idő logaritmikus léptékű koordináta-rendszerében jelöli ki a rendszerek közelítő helyét. A berajzolt ferde vonalak a konstans, adott értékű $P_D \cdot t_{pd}$ szorzat értékekhez tartoznak. Az áramkör-rendszerek közül az a legelőnyösebb a "jósági szám" szempontjából, amely a tengelyek találkozási pontjához a lehető legközelebb jutott a "versenyfutásban".

Egyéb, gyakorlati szempontok szerint is tehetünk összehasonlítást az áramkör-családok között, ezek közül többet a családok ismertetésekor meg is említettünk. Fontos jellemző és sokszor döntő a választásban az áramkör zajtartaléka. Ebből a

szempontból a CMOS a legjobb, különösen ha 10...15 V-os tápfeszültségről működtetjük. Ilyenkor több voltra számíthatunk, ami pl. a TTL max 1 V-jával összehasonlítva sokkal jobb érték "Zajos" ipari környezetben tehát a CMOS használata igen célszerű (léteznek speciális "egyedi" bipoláris HT: High Threshold, nagy küszöbszintű logikák, de ezek típusválasztéka a CMOS-énál sokkal szűkösebb, ára magasabb). A MOS áramkörök zajtartaléka sem túl nagy, "zajos" környezetben célszerű a MOS LSI-vel felépülő áramkör rendszert a "külvilágtól" védetten (árnyékoltan) elhelyezni és kellő zajtartaléku, kisimpedanciás áramkörökkel "körülvenni".



3.89. ábra

Szintén gyakorlati szempont a típusválaszték. LSI-ben jelenleg az NMOS vezet (mikroprocesszor, memória, kalkulátor-IC, DVM logika, stb.), de egyre jobban felzárkózik a CMOS LSI és a bipoláris I²L. A technológiai fejlesztés egyik típusnál sem szünetel, egyre nagyobb rendszereket képesek egyetlen lemezen előállítani ($10^5 \dots 10^6$ alkatelemmel): pl. egyetlen IC-ből felépülő mikroszámítógépeket, CCD (Charge-Coupled-Device: töltéscsatolt elemeket tartalmazó) képfellevő rendszereket, adatátvitelben használatos kódoló rendszereket vagy akár a szórazottató elektronikában felhasználható digitális hang- és képfeldolgozó "processzorokat". Az univerzális építőelemként fel-

használt áramkörök közül típusválasztékban még mindig a TTL és a CMOS vezet, esetenként versengve a piacért, más esetekben kiegészítve egymást. Hasonló a helyzet a később ismertetésre kerülő, a megrendelő által előírható tartalom vagy programozható LSI áramkörök terén (un. CUSTOM LSI, PLA: Programable Logic Array: programozható logikai elrendezés, BOÁK: Berendezés Orientált Áramkörök), itt is nagy a verseny a megrendelők "kegyeiért" a MOS, CMOS, TTL, I²L áramkörök gyártói között. Az ár és a gazdaságos felhasználhatóság kérdése sem kevésbé fontos, itt is nagyon sokszor "üzletpolitikai" elvek érvényesülnek (pl. az újabb, modernebb TTL LS típusokat olcsóbban adják a normál TTL-nél azért, hogy "rászoktassák" a felhasználókat, vagy pl. bizonyos CMOS IC-eket még ennél is olcsóbban adnak, elősegítve CMOS alkalmazását, elterjedését).

3.8. Különböző áramkörcsaládok illesztése

A digitális berendezésekben előfordulhat, hogy a különböző egységek, rész-áramkörök eltérő áramkörcsaládok felhasználásával épülnek meg, és egyes helyeken össze kell kötnünk őket. Ilyenkor felmerül az illesztés kérdése: az adott áramkör képes-e meghajtani (kellő zajtartalékkal) a másikat, ill. a másik biztonságosan fogadni tudja-e ezt a jelet. A lehetséges összekapcsolásoknak igen sok variánsa van, ebből néhány jellemző példát mutatunk be a következőkben.

3.8.1. TTL-CMOS illesztés

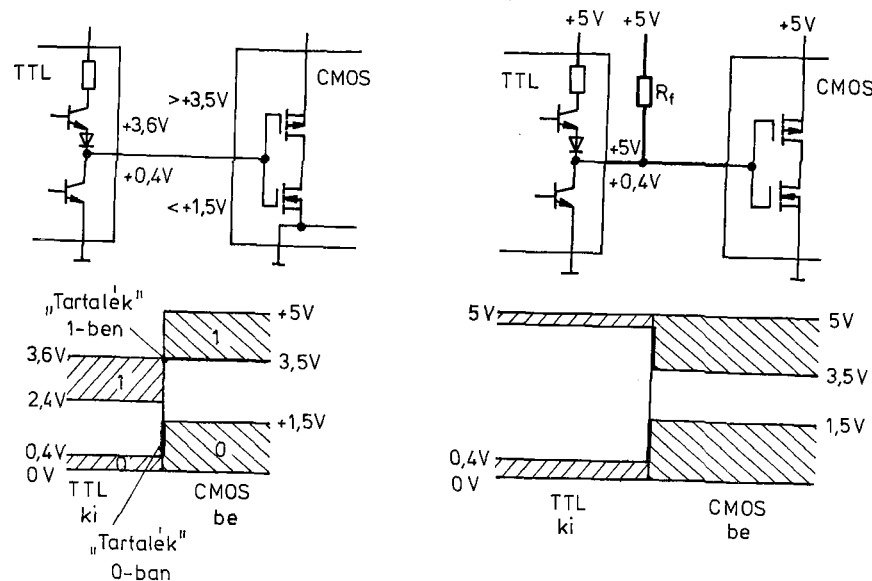
A TTL totem-pole kimenetek logikai 0-ban 0,4 V (Schottky változatok 0,5 V) alatti, logikai 1-ben 2,4 V feletti feszültséget szolgáltatnak a katalógus legrosszabb esetet tükröző adatai szerint.

A CMOS bemenetek - ha a CMOS áramkör 5 V-ról működik - +1,5 V-nál kisebb, ill. +3,5 V-nál nagyobb feszültséget igényelnek. A bemenő áram gyakorlatilag zérus. Ez utóbbi tény figyelembe vehetjük a TTL-hez való illesztéskor: a TTL kime-

net terheletlenül logikai 1-ben kb. $5\text{ V} - 1,4\text{ V} = 3,6\text{ V}$ -os HIGH feszültséget ad. Ebből következik, hogy a TTL közvetlen összekötéssel is illeszkedik a CMOS-hoz:

	TTL kimenet	CMOS bemenet
logikai 1:	+3,6 V	> +3,5 V
logikai 0:	+0,4 V	< +1,5 V

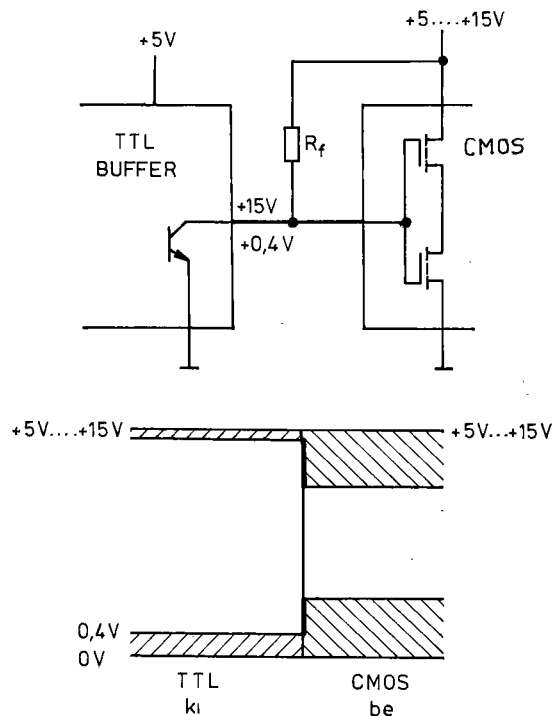
Egyedüli hátrány, hogy logikai 1-ben gyakorlatilag "elfogy" a zajtartalék (0,1 V marad).



3.90. ábra

A kellő zajtartalék biztosítása érdekében célszerű a totem-pole kimenetre egy felhúzó ellenállást helyezni (3.90. ábrán R_f). Ez "segít" a logikai 1-hez tartozó HIGH kimeneti feszültség növelésében: a kimenet logikai 1-es állapotában a végfokozat alsó áramhúzó T3 tranzisztora kikapcsol, ezért a felhúzó ellenállás akadálytalanul felviheti a kimeneti potenciált akár +5 V-ig is (terheletlenül), ui. a felső T4 emitterkövető és a soros D dióda záróirányba kerül és szintén szakadást képvisel. A CMOS bemenet így R_f -en keresztül gyakorlati-

lag +5 V-ot kap. R_f -nek alsó határt az adott TTL család áramhuzó képessége (normál TTL: 16 mA), felső határt a TTL végfokozat alsó T3 tranzisztorának, valamint T4 és D-nek a maradék-árama szab. Természetesen túl nagy ellenállás esetén a sebesség lecsökken, ezért, ha nem is a minimális, de elegendően kis (kohm nagyságrendű) ellenállást célszerű választani.



3.91. ábra

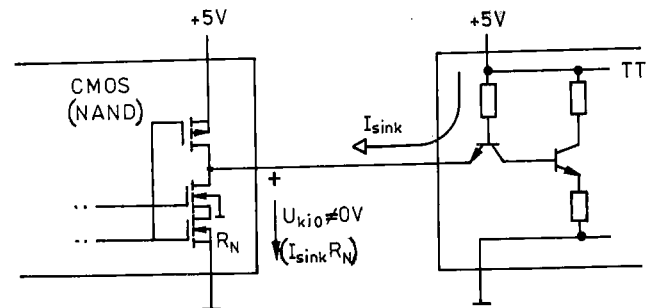
Amennyiben a CMOS áramkör 5 V-nál nagyobb (5...15 V) tápfeszültségről működik, az előbbi út járhatatlan. Legegyszerűbb ilyenkor munkaellenállással ellátott, nagyobb feszültséget tűrő OPEN-COLLECTOR-os buffer elemet használni (3.91. ábra). Ilyen típus például a 7406 (hat inverter, 30 V, 40 mA) vagy a 7407 (hat neminvertáló buffer, 30 V, 40 mA).

3.8.2. CMOS illesztése TTL-hez

Az 5 V-ról működtetett CMOS kimenet feszültségszintek tekintetében elvileg alkalmas lenne TTL bemenet meghajtására. Logikai 1-ben a TTL bemenet csak μA nagyságrendű bemeneti áramot fogyaszt, ezt a CMOS üzembiztosan képes szolgáltatni. Probléma csak logikai 0-ban az ehhez tartozó TTL "sink", huzóáramával lehet. A CMOS típusok kimeneti áramterhelhetőség szempontjából nem olyan egységesek, mint a bipoláris áramkörök. Típusról-típusra minden esetben meg kell győződnünk arról, hogy az áramkör N-csatornás "alsó" tranzisztor(a) a katalóguslap szerint képes(ek)-e a kívánt áramot huzni anélkül, hogy a logikai 0 szint túlzottan megemelkedne (3.92. ábra)! A CD4000A sorozat tagjai között pl. sok olyan van, amely csupán 0,1 mA nagyságú sink áramot tud kiadni +0,4 V-os logikai 0 feszültségnél (pl. CD4011A, 4012A NAND kapuk). Közvetlen összeköttetéssel való illesztésre csak azok a típusok alkalmasak, amelyekre garantálják a kb. 0,4 mA kimenő áramot. Ezek a BUFFERED áramkörök, pl. a:

- CD 4000 B
- MC 14000 B
- MC 14500 B
- F 34000
- 74 C.... sorozat.

Kimenetükre 2 L TTL (TTL Low power) bemenet, ill. 1 LS TTL (TTL Low Power Schottky) bemenet köthető.

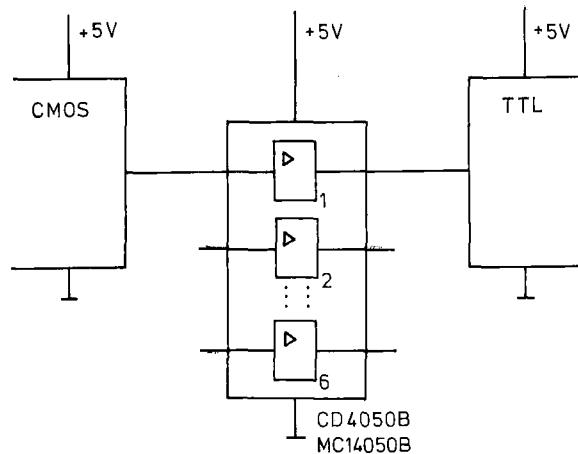


3.92. ábra

Normál TTL meghajtására, valamint kis kimeneti áramú típusok illesztésére célszerű a kimondottan erre a célra készült CMOS buffereket alkalmazni. Ma használatos, elterjedt típusok pl. a:

- CD 4049 B (6 invertáló buffer), ill. a
- CD 4050 B (6 neminvertáló buffer).

Egy-egy CMOS kimenetre az egyetlen tokban levő hat buffer közül egyet rákötve, biztonságosan nagyra, tipikusan 6 mA-re, a legrosszabb esetben 3,2 mA-re növelhetjük a huzóáramot (2,5 mA-re, de legrosszabb esetben 1,25 mA-re növelve a forrásáramot is) 5 V tápfeszültség mellett (3.93. ábra).

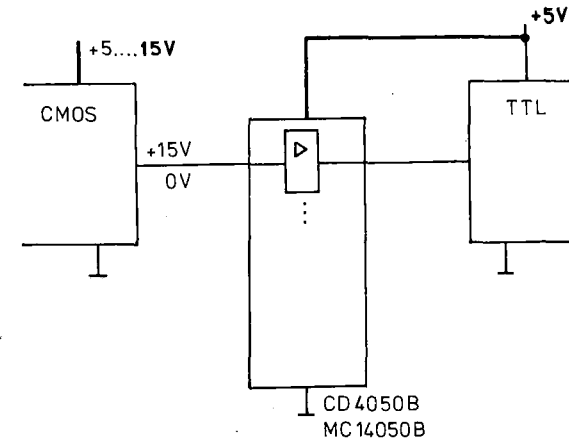


3.93. ábra

Előfordulhat, hogy +5 V feletti tápfeszültségről működő CMOS kimenetet kell TTL bemenethez illesztenünk. Az előbb említett

- CD 4049 B és
- CD 4050 B

bufferek erre a célra is alkalmasak (3.94. ábra). A buffereket ekkor TTL +5 V-os tápfeszültségéről működtetjük, bemenetüket a "nagyfeszültségű" CMOS kimenetre köthetjük. Ezekre a buffer típusokra ugyanis megengedett, hogy bemeneti feszültségük nagyobb legyen a tápfeszültségnél!



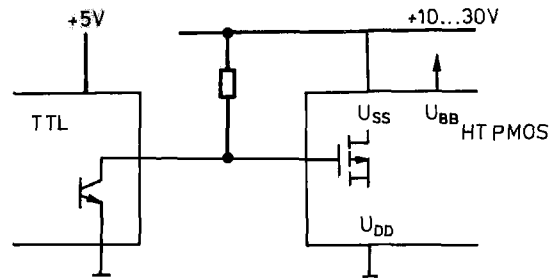
3.94. ábra

3.8.3. TTL-MOS illesztés

Kis küszöbszintű (Low-Threshold) MOS (ilyen a legtöbb "mai", +5 V-ról működő MOS rendszer) TTL-ről történő meghajtása általában nem okoz problémát, mivel igényelt bemeneti szintjei általában a 0 V-nál körülbelül 1,5 V-tal nagyobb ($U_{SS} + 1,5 V$), ill. a tápfeszültségnél körülbelül 1,5...2 V-tal kisebb ($U_{DD} - 1,5...U_{DD} - 2 V$) feszültségek. Ebből következően legtöbbször a közvetlen illesztés is megfelelő. A TTL viszonylag alacsony 1-szintű feszültsége miatt esetenként kevés a zajtartalék, ilyenkor a CMOS illesztésnél ismertetett felhúzó ellenállásos kiegészítés javasolható (3.90. ábra).

A nagy küszöbszintű (high-Threshold) MOS-hoz csatlakozás igénye ma már egyre kevesebb esetben merül fel, mert ez a család a legrégebb, főleg a P-csatornás, nagy tápfeszültségű (egészen -30 V-ig) generáció. A közvetlen meghajtás nem járható ut, mindenképpen szintáttevő áramkörre van szükség. Nem kritikus sebesség igény esetén legegyszerűbb az open-collector-os meghajtás a MOS tápfeszültségére kapcsolt munkaellenállással a 3.91. ábrához hasonlóan. Ez a megoldás PMOS-hoz akkor megfelelő, ha lehetőség van a negatív tápfeszültség olyan "átrendezésére", hogy a negatív U_{DD} -t földeljük és ehhez képest

adunk az áramkörre pozitív U_{SS} -t (3.95. ábra). Léteznek olyan INTERFACE áramkörök, amelyek kimondottan MOS meghajtásra készülnek a sebesség megtartásával (pl.: TEXAS: SN 75...sorozat, pl. SN 75361A, SN 75450B ezekről a későbbiekben, a következő kötetben lesz szó).



3.95. ábra

3.8.4. MOS-TTL illesztés

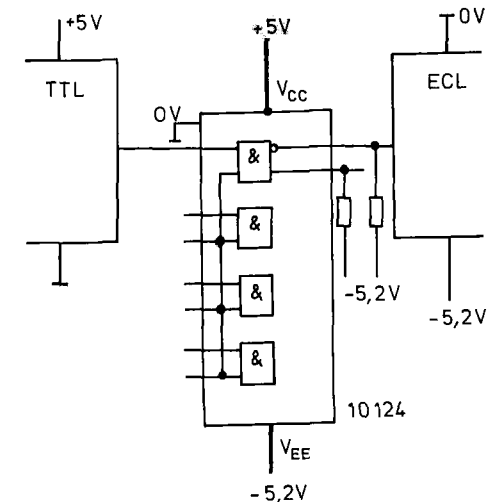
A "mai" kis küszöbszintű NMOS, 5 V-ról működő áramkörök általában képesek közvetlenül 2 L TTL vagy legalább 1 LS TTL áramkört meghajtani. Legtöbb esetben (pl. mikroprocesszoros rendszerekben) a vezetékeket, buszokat egy, ill. kétirányú bus buffer/driver-ekkel amúgyis "megerősítik" (maguk a bufferek is rendszerint TTL áramkörök), így a TTL kompatibilitás biztosított (lásd a 3.5.3. fejezetet a 3-state rendszerről!). Az áramterhelhetőségi határookra (fan-out) természetesen minden esetben figyelemmel kell lenni.

A nagy küszöbszintű MOS áramkörök illesztéséhez legalkalmasabbak az erre a célra készülő interface típusok (SN 75..., pl. SN 75370 - lásd később).

3.8.5. TTL-ECL illesztők

A két bipoláris logika egészen eltérő szintekkel és tápfeszültségekkel dolgozik, ezért közvetlen összekötés szóba sem jöhet. Illesztéskor lényeges szempont a működési sebesség meg-

őrzése, ezért csak "professzionális", erre a célra kifejlesztett szintáttevő IC ("level-translator") javasolható. A 10 000 sorozatu áramkörök típusválasztékában szerepel TTL LOK translator, ilyen pl. a 10124-es típus: ez a TTL jelet szimmetrikus ECL jelre "fordítja le". Az IC-re a +5 V-os TTL és a -5,2 V-os ECL tápfeszültséget is rá kell adnunk. Egy tokban 4 db AND-NAND szint áttevő van egy közös tiltó-vezetékkel (3.96. ábra). A TTL oldalon 3,2 mA bemeneti húzóáramot fogyaszt (normál, ill. Schottky TTL-hez készült). Saját késleltetési ideje tipikusan 5 ns. Egyéb (régebbi) ECL típus-sorozatok mind-egyikéhez kapható "kész" szintáttevő IC.

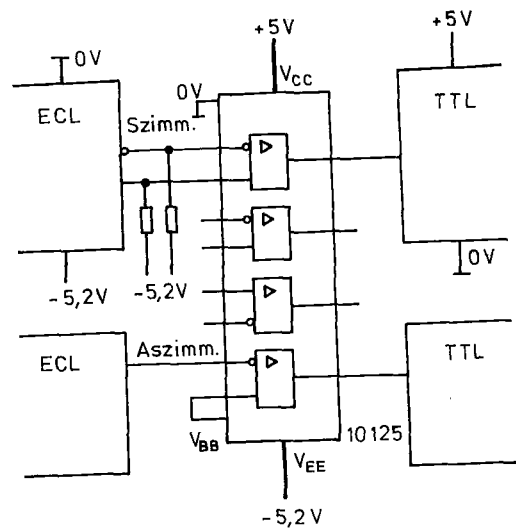


3.96. ábra

3.8.6. ECL-TTL illesztők

A helyzet az előző esethez hasonló: az illesztést kimondottan erre a célra készült - és az ECL rendszerek típusválasztékában mindig megtalálható - IC-vel érdemes megvalósítani. A LOK sorozatban pl. a 10125 a megfelelő típus. A tokban levő 4 szintáttevőnek szimmetrikus (differenciál)bemenete van, amelyet a kívánt polaritással szimmetrikus (ponált-negált)ki-

menetű ECL áramkörhöz csatlakoztathatunk (aszerint, hogy invertálni kívánjuk a jelet vagy nem). Aszimmetrikus csatlakoztatáskor a "fennmaradó" bemenetet a tokból kivezetett V_{BB} ponthoz (hőkompenzált előfeszültséghez) kell kötnünk (3.97. ábra). A TTL kimenet 10 Schottky áramkörrel terhelhető ($t_{pd} = 5 \dots 7$ ns).



3.97. ábra

4. TÍPIKUS KOMBINÁCIÓS HÁLÓZATOK ÉS MEGVALÓSÍTÁSI LEHETŐSÉGEIK SSI - MSI - LSI ELEMEKKEL

4.1. SSI áramkörök

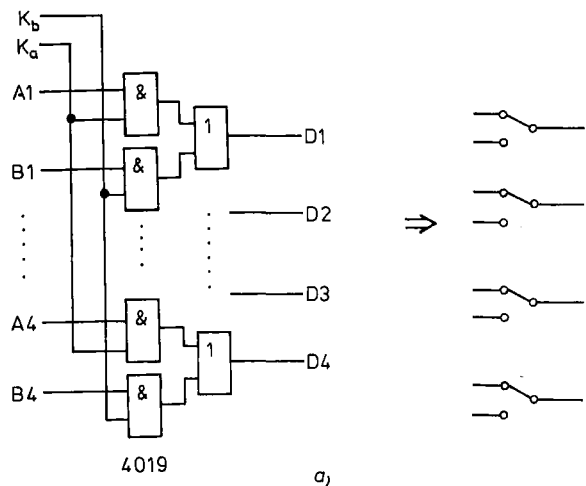
Az SSI (Small Scale Integration, kismértékben integrált) áramkörök csoportjába tartoznak a legegyszerűbb alapáramkörök: az inverterek, a különféle kapuk, a speciális rendeltetésű inverter-kapú változatok (bufferek, 3-state elemek), valamint a sorrendi áramkörök alapelemei (amelyekről a következő fejezetben lesz szó). Az inverterek, kapuk logikai művelet végzésre, "jel-kapuzásra" történő felhasználásáról a 2.2. fejezetben már részletesen szó volt. Arra is felhívtuk a figyelmet, hogy egy áramkör megtervezésekor az alkalmazni kívánt család áramkör-készletét, típusválasztékát mindig szem előtt kell tartanunk, máskülönben a "minimális" hálózat csak papíron marad minimális. A katalógus használata tehát rendkívül fontos. Arra is törekednünk kell, hogy SSI áramköröket csak akkor alkalmazzunk, ha "feltétlenül szükséges" és nem kapható az adott feladatra "kész", MSI áramkör (vagy több adott feladatra LSI!). Az SSI áramkörökből való építkezés ui. - éppen a kismértékű integrálás miatt - sokkal munkaigényesebb (sok kivezetést, pontot kell összekötni, bekötni), így sokkal drágább, mint az MSI-LSI áramkörök használata.

Amint arról az egyes áramkör-rendszerek tárgyalásakor szó volt, mindegyiknek van egy "alap áramköre" (pl. invertere 54/7404 stb.) és egy ehhez hasonló alap-kapú áramköre.

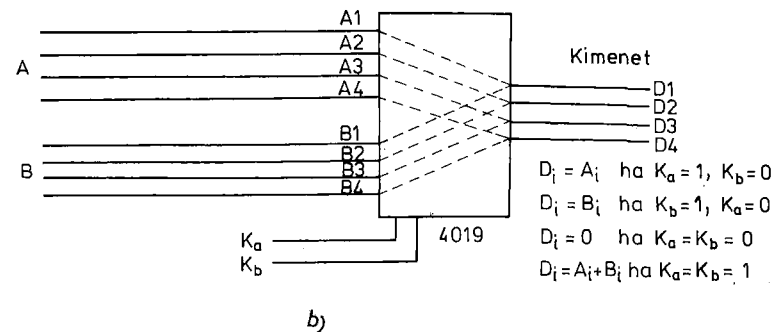
A bipoláris TTL 54/74 sorozat alap-kapuja a NAND, ebből legnagyobb a típusválaszték, ezek kaphatók a legtöbb változatban (Normál, LS, L, H, S) és a legtöbb "konfigurációban" (2 bemenettel: 54/7400, - a továbbiakban elhagyjuk a típuszám-ból a sorozat jelet - 3 bemenettel: 10, 4 bemenettel: 20, 8 bemenettel: 30, open collectorosak, 2 bemenettel: 01, 03, 3 bemenettel: 12, 4 bemenettel: 22, Schmitt-triggeres bemenet-

tel, inverter: a 14, 2 bemenetű: a 132, 4 bemenetű: a 13). A többi kapu inkább "kiegészítő" szerepű, teljesebbé teszi a típusválasztékot, de kevesebb változatban kapható: jelentősebbek a NOR kapuk (2 bemenettel: 02, 3 bemenettel: 27, 4 bemenettel: 25) és az AND-OR-INVERT kapuk (51, 52, 54), de főleg ezek bővíthető változata (50, 53, 55) a hozzájuk csatlakoztatható ÉS bővítővel (60, 61). Az ÉS kapu (AND: 08, 11, 21), valamint az egyetlen 2 bemenetű VAGY kapu (OR: 32) "speciálisnak" számít - hosszabb is a késleltetési idejük, mint a "normális" kapuknak. Létezik kizáró-VAGY (EOR: 7486) kapu is, de ezt aritmetikai áramkörként kezelik.

A CMOS rendszerben egészen kismértékben túlsúlyban van a NOR (2 bemenetű: CD 4001, 3 bemenetű: 4000 és 4025, 4 bemenetű: 4002, 8 bemenetű: 4078), de a NAND is sokféle típusban létezik (2 bemenetű: 4011, 3 bemenetű: 4023, 4 bemenetű: 4012, 8 bemenetű: 4068), gyakorlatilag egyformán elterjedt mindkét kapu típus. (A 74C sorozatra a TTL típusválaszték jellemzőbb.) Ebben a rendszerben is van AOI kapu (4085), bővíthető változatban is (4086, ennél a bővítő bemenet is logikai szintű), valamint van EOR (4030, 4070) és ENOR (4077) kapu. Létezik ezenkívül néhány "CMOS specialitás" is, pl. négyes ÉS-VAGY választó kapu (Quad AND/OR Select Gate: 4019, 4.1a ábra).



4.1a ábra



4.1b ábra

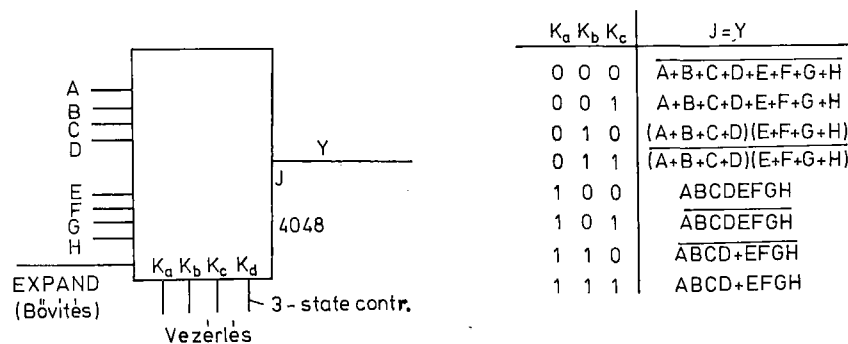
$$D_i = A_i \text{ ha } K_a = 1, K_b = 0$$

$$D_i = B_i \text{ ha } K_b = 1, K_a = 0$$

$$D_i = 0 \text{ ha } K_a = K_b = 0$$

$$D_i = A_i + B_i \text{ ha } K_a = K_b = 1$$

A VAGY kapuk előtt levő két-két ÉS kapu egy-egy bemenetét két vonalra kihozták (K_a és K_b). Ezek állapotával lehet tiltani, vagy engedélyezni a másik (A_i , ill. B_i) ÉS bemenetre érkező jel "továbbhaladását". Például 2 különálló 4 bites adatforrás (adatvonal) jelét lehet kiválasztani és a "közös" 4 vonalra rákapcsolni K_a és K_b ellentétes vezérlésével, ill. mindkét jelet tiltani lehet, ha $K_a = K_b = 0$ (vagy mindkét jelet "egyszerre" a kimenetre engedni, VAGY kapcsolatát létrehozni, ha $K_a = K_b = 1$). Hasonlóan érdekes típus a 4048-as bővíthető 8 bemenetű sok célú (multifunctional) 3-state kimenetű kapu: 3 bináris bittel lehet vezérelni (8 lehetséges kombinációban) azt, hogy az ABCD és az EFGH bemenet csoporttal (és a bővíthető bemenettel) milyen logikai függvényt állítson elő (fejlesztési munkához célszerű, tömbvázlatát a 4.2. ábra mutatja).



4.2. ábra

"Ujabb generációs" bufferelt változatban kaphatók ÉS kapuk és VAGY kapuk is, de ezek késleltetése nagyobb mint az univerzális változatoké (a többszörös invertálás miatt). Szintén újabb változatban, az MC 14500 sorozatban létezik ezenkívül "vegyes" kapu (2 NAND + 1 OR/NOR: MC 14501 UB, vagy 4 inverter + 1 NOR + 1 NAND : MC 14572 UB), 5 bemenetű "majoritás" kapu (MC 14530 B) stb. A típusválasztékhoz tartoznak a TRANSMISSION-GATE változatok is (4016, 4066).

Az ECL-ben az SSI kombinációs típusválaszték nagy része NOR kapu, ez az alapáramkör. Például a 10k sorozatban a 10100: négyes 3 bemenetű NOR, 10101: négyes 2 bemenetű OR/NOR kapu szimmetrikus kimenettel, lásd a 3.74. ábrát! Van ezenkívül OR-AND és OR-AND-INVERT kapu (10119 ill. 10121, ezekben az ÉS kapcsolatot belső huzalozással, a NAND kapcsolatot "open emitteres" fokozatok összekötésével hozzák létre jelkésleltetési idő növekedés nélkül!). Külön kategóriába tartoznak a kapunak, differenciál vonal-meghajtónak és differenciál vonalvevőnek is használható interface típusok. Interface kapu-módosulatok az előzőekben említett translator-ok (szint áttevők) is.

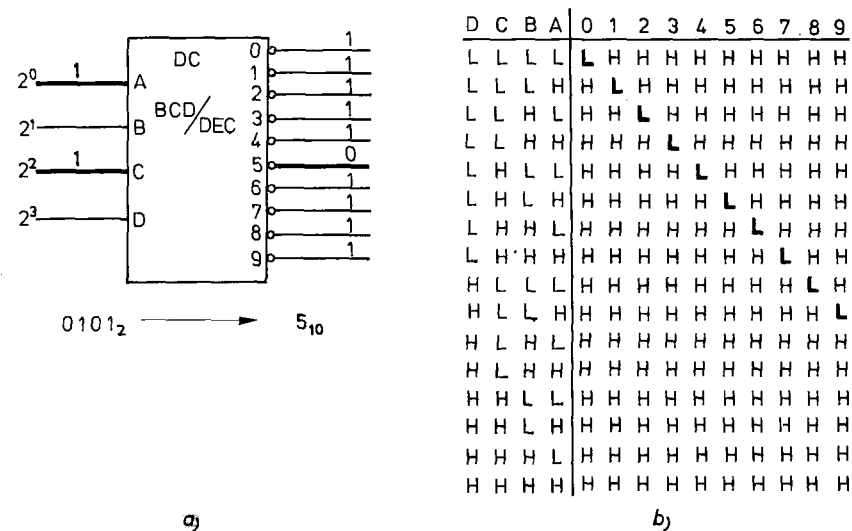
4.2. MSI kombinációs áramkörök

Az integrált áramköröket előállító gyárak már a kezdeti időkben felfigyeltek arra, hogy az akkoriban forgalomban lévő kapukból általában ismétlődően bizonyos típusú funkcionális áramkör típusokat építettek a felhasználók (pl. kódolókat, dekódolókat, aritmetikai elemeket, stb.). Bebizonyosodott, hogy gazdaságos ezeket a tipikus áramköröket integrálni és "készen", egy áramköri egységként árusítani. Ezekből a tipikus, több kaput tartalmazó funkcionális integrált egységekből jöttek létre a ma kapható és használt MSI kombinációs IC-k. Ha áttekintjük a legfontosabb MSI integrált áramkör fajtákat, akkor egyben képet kapunk arról is, hogy milyen kombinációs hálózatokra, egységekre van szükség a gyakorlatban. A továbbiakban, néhány jellemző példát mutatunk be.

4.2.1. Dekódoló, kódoló (kódváltó)

- BCD-decimális dekódoló

Feladata a 4 bites BCD kódot decimálisra átalakítani; "ahányas" BCD (binárisan kódolt decimális) számjegy érkezik a bemenetre, "annyiadik" kimenetnek kell aktívnak lennie (a 4.3a ábrán pl. az 5-ös BCD bemeneti kombináció hatására az 5-ös számú kimeneten jelenik meg jel). Figyeljük meg: a bemenetre 4 bites, bináris, 0-ákból és 1-ekből álló jelkombináció érkezik, a kimeneten viszont egyszerre mindig csak egyetlen egyen jelenik meg jel. A 10 kimenet közül mindig csak 1 "él": az ilyen dekódolót 1 a 10 közül, általában "1 az N közül" ("1 out of N", "1 of N") dekódolónak nevezzük.



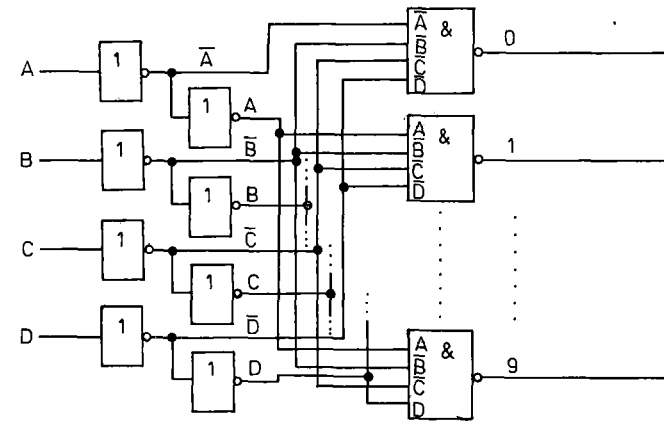
4.3. ábra

"Takarékos" áramköri megvalósítását a don't care minterek felhasználásával már a 2.4. fejezetben megterveztük. A cél ott az volt, hogy az áramkör a lehető legegyszerűbb legyen, ennek érdekében felhasználtuk azt a tényt, hogy a BCD csak a bináris 0-tól a bináris 9-ig értelmezhető, az egyéb eseteket don't care ("mindegy") állapotoknak tekintettük. Ennek természetesen az lett az eredménye, hogy az így létrehozott "taka-

rékos" áramkör a 10...15-ös bináris bemeneti kombinációkra az "előre el nem határozott" kimeneti jelekkel válaszolt (pl. egyszerre több kimeneten is megjelenik jel, ami nem felel meg az "1 of N" elvnek). Logikusabb és célszerűbb a tiltott bemeneti kombinációk esetére (10...15) a kimeneteket is tiltani úgy, hogy ilyenkor egyik kimeneten se keletkezzen jel (más előírást is lehetne teljesíteni, a lényeg az, hogy a tiltott bemeneti állapotokra is definiálni kell a kimenetek állapotát, azaz megszűnik a don't care állapotokkal való egyszerűsítés lehetősége). Az áramkör ebben az esetben bonyolultabb, de könnyen belátható, hogy ennek az MSI áramkörök esetében nincs túlzott jelentősége, legfontosabb szempont az optimális felhasználhatóság.

TTL-ben a legismertebb típus az 54/7442 (LS42), amelynek tömbvázlata megegyezik a 3.50a ábrával. Fontos rögzíteni az általános jelölési szabályt: a bemenet legkisebb helyiértékű bit-jét, LSB-jét, a 2^0 -nak megfelelő A betűvel, a 2^1 -nek megfelelő B betűvel, a 2^2 -nek megfelelő C-vel, a 2^3 -nak megfelelő D-vel.... jelöljük. Ez a dekódoló aktív 0 ("active low") kimenetű, ami azt jelenti, hogy az aktivált kimenet, amelyen jel van, 0-szintű, L, az összes többi "nyugalmi" 1-en, H-szinten van (l. a 4.3b ábra igazságtáblázatát! pl. a BCD 5-ös bemeneti jelre az 5-ös számú kimenet lesz L szintű, a többi H-szinten marad). Az aktív 0-ra hívja fel a figyelmet a kimeneti vezetékhez rajzolt null-kör. Felhasználáskor nagyon fontos minden esetben tájékozódni arról, hogy aktív 0 vagy aktív 1 kimenetű-e az áramkör, mert később egy "tévesztés" nagyon kellemetlen következményekkel járhat (újra tervezés, új nyomtatott huzalozás készítése stb.). A 7442-es "belseje" (4.4. ábra) annyi NAND kapuból áll, ahány kimenet van, ezek "kapuzzák ki" az A, B, C, D ponált és negált értékéből a bináris számértéket ($Y_0 = \overline{A}BCD$, $Y_1 = A\overline{B}CD$,... $Y_9 = A\overline{B}C\overline{D}$, egyszerűsítésre most nincs lehetőség). A kapcsolási rajzon feltűnik, hogy a bemeneti ponált értékek nem azonnal mennek a kapuk bemenetéhez, hanem két inverteren keresztül: először a negált áll elő, és utabb invertálás után a ponált. Ezzel a módszerrel a bemeneti vezeték terhelését csökkentik, mindegyik csak 1 inverterhez megy (több kapu bemenet helyett) így csak 1 egységnyi

terhelést képvisel egy bemenet! Ezzel a "bemeneti bufferes" megoldással találkozunk minden egyéb olyan áramkörben, amelyekben a bemeneti változók jelét többször is felhasználják (kivéve a "nagyon régi", első generációs típusokat), ezért a további ismertetésekben ehhez nem fűzünk külön magyarázatot.



4.4. ábra

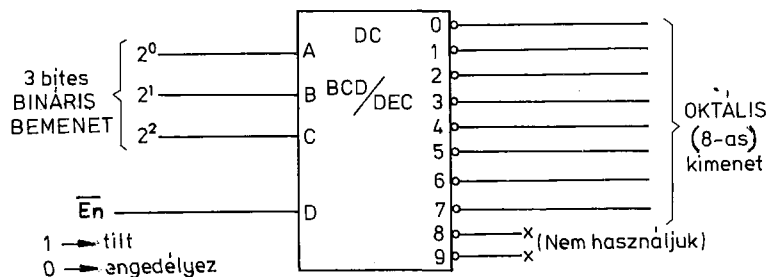
BCD decimális dekódoló és meghajtó a 7445-ös (30 V, 80 mA), a 74145-ös (15 V, 80 mA), ill. a 74LS145-ös (15 V, 12 mA) áramkör. Számjelijző meghajtáskor kimondottan előnyös, hogy az áramkör "érvénytelen" BCD kódra nem ad ki aktív L jelet (kialszik a számjegy). Régebbi, 10 számjegy-katódos gáztöltésű számjelijző (NIXIE) csövek meghajtásához készült a "nagyfeszültséget" is tűrő 74141 (60 V, 7 mA, aktív L, nem "TTL kimenet!").

A CMOS családban a CD4028 (MC14028) az univerzális BCD decimális vagy bináris-oktális dekódoló (aktív 1 kimenettel!).

- Bináris oktális dekódolók

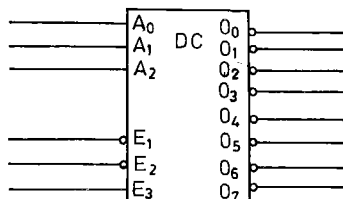
Az oktális rendszerben 3 bites bináris számcsoportot kell 8 vezetékre dekódolni. Ez minden további nélkül megoldható BCD decimális dekódolóval is úgy, hogy csak az első 3 bit-jét A, B, C-t "vesszük figyelembe", a D bemenetet 0-ra kötjük. A kimenetek a 0...7-ig számozott kimenetek lesznek. A D bemenet is felhasználható Engedélyező (Enable) negált bemenetként:

ha $D = 0$, akkor az oktális dekódoló az előbbiek értelmében "1 a 8 közül" dekódolóként működik, ha $D = 1$, akkor bármit is adunk a többi bemenetre a dekódolóra kerülő BCD jel 8 vagy annál nagyobb, ezért a 0...7 kimeneten nem jelenik meg jel, a dekódolót "letiltottuk" (l. a 4.3. ábra igazságtáblázatát és a 4.5. ábrát).



4.5. ábra

Az INTEL - mikroprocesszor kiegészítő elemként - gyárt sokoldaluan felhasználható "1 a 8 közül" dekódolót 8205 típus megjelöléssel. Ennek 3 jelbemenete van (A_0, A_1, A_2) és ezenkívül 3 Enable bemenete (E_1, E_2, E_3 ; 4.6. ábra). Ahhoz, hogy a dekódoló engedélyezve legyen az E_1 -re és E_2 -re L szintet, az E_3 -ra H szintet kell adnunk. Hasonló típus a TEXAS 74138 (LS138) dekódolója (az engedélyező bemeneteket ennél G_1 és G_{2A} ill. G_{2B} -vel jelölik). A három engedélyező ponált-negált bemenet előnyösen használható, ha kár 0, akár 1 szinttel szeretnénk vezérelni az áramkör működését, de főleg akkor előnyös, ha a dekódolót több bitre kívánjuk kiterjeszteni (lásd később).

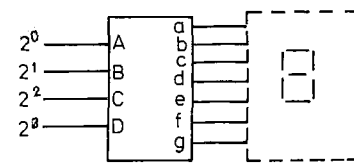
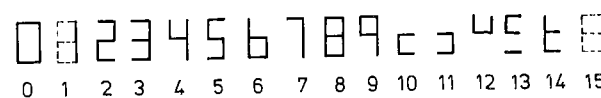
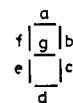


8205
Engedélyezés = $\bar{E}_1 \cdot \bar{E}_2 \cdot E_3$

4.6. ábra

- BCD-7 vonalas dekódolók (BCD-7 segment decoder)

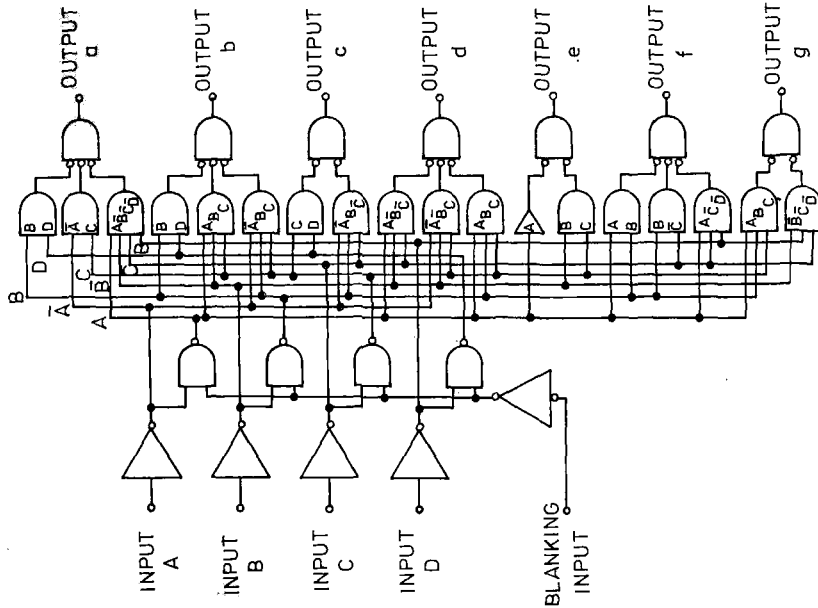
A számkijelzés és az alfanumerikus megjelenítés témája a későbbiekben részletesen szerepel, itt csak a dekódoló szerkezetével, működési elvével foglalkozunk. Ezek a dekódolók egy "BCD számjegyet" (A, B, C, D bitet) alakítanak át úgy, hogy a kimenetek jeleivel a 4.7a ábrán látható 7 vonalas kijelzőt meghajtva kirajzolódjon a megfelelő számjegy.



Kijelző és áramköre
a)

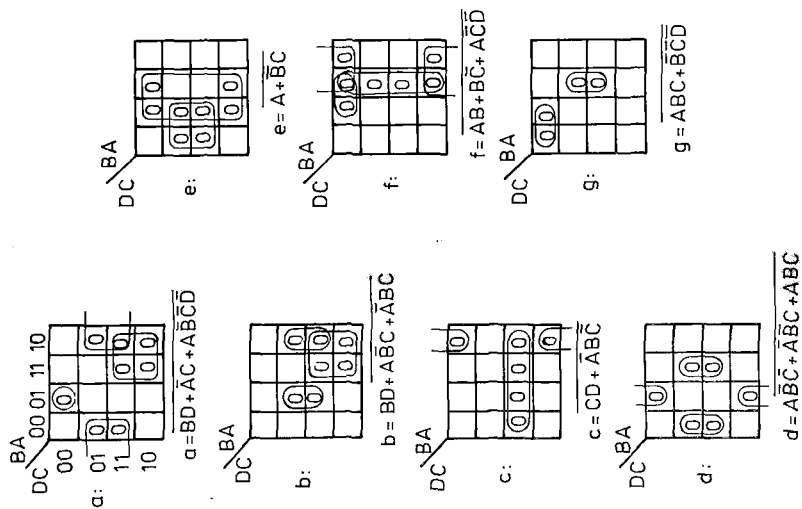
4.7a ábra

Lényeges észrevennünk, hogy ez a dekódoló típus nem "1 az N közül" működésű, mert egyszerre több kimenetnek is aktívnak kell lennie egy-egy szám kirajzolásakor. A 4.7b ábrán a legelterjedtebb 7446-47 (aktív 0 kimenetű), 7448-49 (aktív 1 kimenetű) TTL dekódolók kijelzési képeit látjuk az összes lehetséges bemeneti kombinációra. Látható, hogy pl. a 0 szám kijelzésekor egyszerre aktívnak kell lennie az a, b, c, d, e, f kimeneteknek, az 1-es kijelzésekor a "b" és "c" kimenetnek stb. A dekódolóban levő hálózatnak szükségképpen bonyolultabbnak kell lennie az "1 az N közül" hálózatnál. A bonyolultabb hálózatot TTL-ben rendszerint AND-OR-INVERT-tel valósítják meg (Emlékezzünk: AOI hálózatához úgy jutunk, hogy a minterm táblában a 0-kat vonjuk össze.) Ebben az esetben az áramkör megtervezéséhez mind a 7 kimeneti függvényhez egy-egy minterm táblát kell rajzolnunk a 0 helyek kitöltésével. Ezt közvetlenül a kijelzési képekre ránézve elvégezhetjük; az a...f minterm táblákba azokhoz a bemeneti kombinációkhoz kell 0-át ren-



c)

4.7b, c ábra



b)

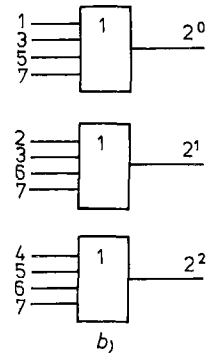
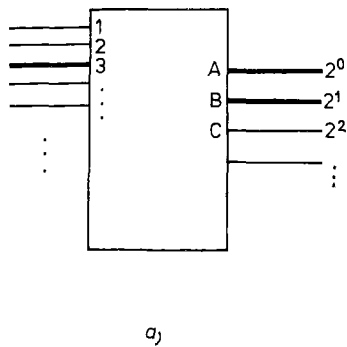
delnünk, ahol nem aktív az illető kijelző szegmens. A 4.7b ábrán összefoglalóan felrajzoltuk a KARNAUGH-táblákat és felírtuk az egyszerűsítés után kapott függvényeket. A 4.7c ábrán az SN 7449 IC típus belső kapcsolási rajzának másolatát mutatjuk be: a rajz pontosan az általunk egyszerűsített függvények megvalósítása! (A BLANKING INPUT-tal kiegészítve: ezzel a bemenettel kioltható valamennyi szegmens). A többi említett típus "belseje" is hasonló, csak a 7446-47 minden kimenetén még egy-egy inverter van (aktív 0 kimenet), ill. kiegészítő bemenetek és kimenetek is találhatóak rajtuk (LT: Lamp Test: ennek aktiválásával valamennyi szegmens kigyujtható a kijelző ellenőrzése céljából, RBI: Ripple Blanking Input: kioltó bemenet, amely több digitális kijelzéskor az előző helyiértékről származó jellel kiolthatja a kijelzőt a "felesleges nullák" kioltása érdekében, pl. "0036 helyett 36" kijelzést mutatva, BI/RBO: Blanking Input/Ripple Blanking Output: ugyanennek az említett jelnek a továbbadására szolgáló kimenet, ill. feltétel nélküli kioltó bemenet, open collector-os kivitelben).

A CMOS típusválasztékában is bőven van BCD-7 vonalas dekódoló. A "második generációs" típusok közvetlenül LED kijelző meghajtására is alkalmasak. (CMOS "létükre" 6-10 mA kimeneti áramot képesek szolgáltatni, pl. MC 14511B, 14513B, 14544B stb. ez utóbbi folyadékkristályos kijelző meghajtására is alkalmas - lásd később).

A kijelzők leggyakrabban valamely számláló vagy tároló tartalmát kell, hogy jelezzék, ezért a "tipikusan együtt előforduló áramköröket" egyetlen MSI-ben (LSI-ben?) is integrálják (pl.: 74143: BCD számláló + 4 bites tároló + BCD-7 vonalas dekódoló meghajtó, áramgenerátoros - 15 mA-es - kimenettel LED meghajtására vagy CMOS-ban: CD 4026: dekádszámláló 7 vonalas dekódolóval stb.). Kaphatók a kijelzővel "egybe integrált" számláló-tároló-dekódolók (TIL 306...) is.

- Oktális-bináris, decimális BCD kódoló:
prioritás kódoló (priority encoder-ek)

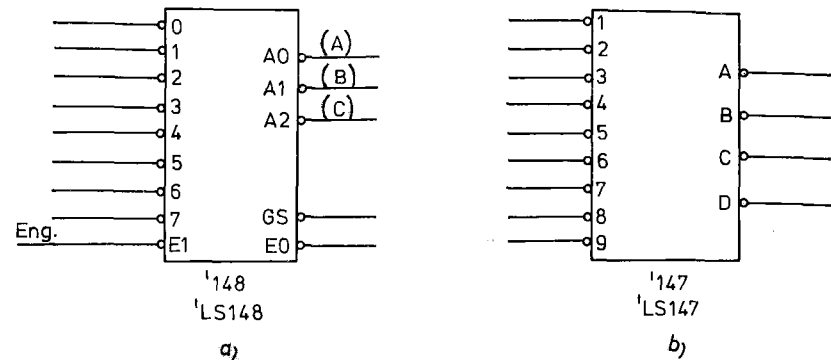
Az előbbieken említett bináris-oktális (BCD dedimális) dekódolás "ellenkezője" is előfordulhat megoldandó feladatként: azt a bináris (BCD) számot kell az áramkörnek kiadnia,



4.8. ábra

amelyik sorszámú bemeneti vezetéken jel van (4.8a ábrán pl. a bementi 3-as vezeték aktív, akkor a kimeneten a bináris 3-as számnak kell megjelennie). Ilyet nagyon egyszerűen készíthetünk VAGY kapukból (egy oktális-bináris aktív 1 bemenetű, aktív 1 kimenetű változatot mutat a 4.8b ábra). A MSI változatok többet nyújtanak: prioritási elven kódolnak, azaz ha egyszerre több bemeneten jelenik meg aktív jel, akkor a legmagasabb sorszámút "veszik figyelembe", ennek bináris (BCD) megfelelőjét állítják elő a kimeneten. TTL oktális-bináris priority encoder, a 74148 (LS148)-as típus, amely aktív 0 bemenettel és aktív 0 kimenettel működik. Kiegészítették aktív 0-ra működő engedélyező bemenettel (EI) és a több IC egymás után kötését lehetővé tevő EO kimenettel (valamint egy, a bemeneti jel jelenlétét jelző GS kimenettel, 4.9a ábra). "Ujabb generációs" változata a 74LS348 3-state kimenettel. Decimális-BCD (aktív 0) változat a 74147 (LS 147) típus (4.9b ábra). A CMOS-ban a 148-ashoz hasonló, de aktív 1-gyel működő típus a 4532B (MC 14532B).

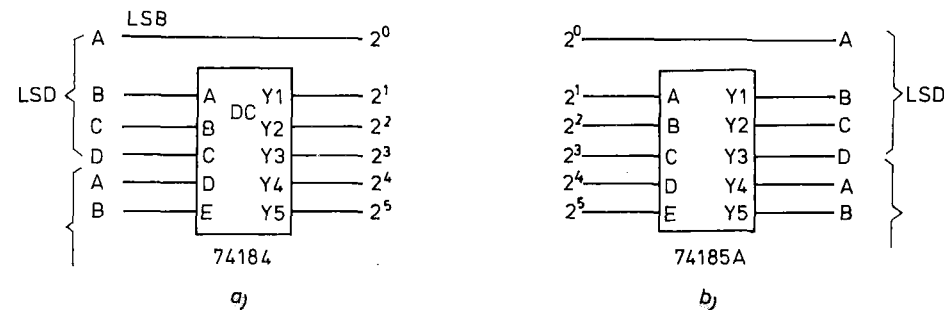
A priority encoder-ek fő felhasználási területe több jel "sorrendbe állítása". A "magasabb prioritású" jel megjelenése letiltja az alacsonyabb prioritásút, annak jelzése érvényesül, de sokszor alkalmazzuk "fordított" kódolóként. Gyakori a mérés-technikai felhasználás is.



4.9. ábra

- BCD-bináris, bináris-BCD kódátalakítók

Ritkán előforduló feladat 4-es csoportokban megjelenő, és 0...9 számokat kódolva tartalmazó BCD adatot "teljes bináris", 2 hatványai szerint rendezett formába átalakítani vagy ugyanezt fordítva - de ha ezt áramkörökkel (nem programmal) kell megoldanunk, akkor meglehetősen nehéz a megvalósítás. A sok változó miatt Karnaugh-táblás minimalizálást nem célszerű végezni, a "logikai uton" való megoldás túlzottan nehéz.



4.10. ábra

A TTL MSI áramkörök között található BCD-bináris (74184, 4.10a ábra) és bináris-BCD (74185A, 4.10b ábra) "kész" kódátalakítók, amelyek "egy lépésben" 6 bit átalakítására képesek (az LSB mindkét kódban megegyezik, ezért ezen a helyen pusztán összekötés szükséges). Ezeknek az integrált áramköröknek a "belsejét" fel sem rajzolják, mert ezekben nem minimalizált

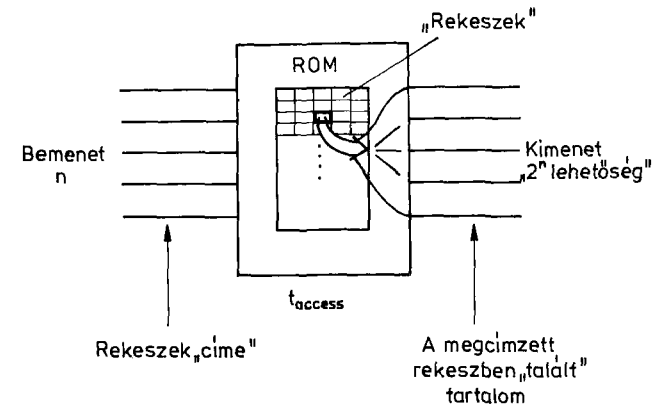
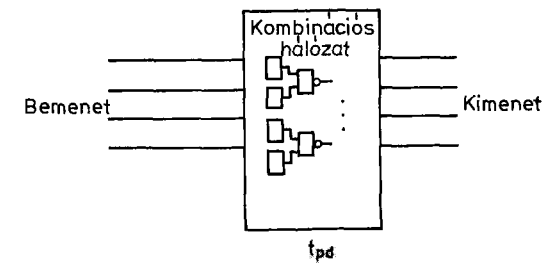
kapu hálózat van, hanem egy állandó tartalmu memória (ROM), amely "tudja", hogy egy-egy bemeneti kombinációra milyen kimeneti jelkombinációval kell "válaszolni" ("ezt programozták bele"). A konverzió mindkét esetben több bitre is kiterjeszthető a katalógus utmutatásai alapján.

- A kódváltók és a memória áramkörök (ROM-ok) közötti rokonság

Láttuk, hogy a "bonyolultabb" kódolási feladatot pl. a TEXAS már nem kapukból összeintegrált MSI áramkörrel, hanem előre programozott, állandó tartalmu memóriával (a "csak olvasható memória" rövidítése a továbbiakban ROM: Read Only Memory) oldotta meg. Ebből látszik a kódváltók (ezzel együtt a kombinációs hálózatok) és a ROM funkciójának hasonlósága, rokonsága. A ROM is ugyanazt teszi, mint a kombinációs hálózat: adott számú bemenetére - a cím bemenetekre - jutó 0-1 kombináció (bináris cím) hatására "előkeresi" az ezen a címen tárolt és előre programozott bit-kombinációt, és ezt a kimenetekre adja, vagyis ugyanugy "tesz", ahogy a kombinációs hálózatok; egy adott bemeneti kombinációra a késleltetési ("hozáférési") idő elteltével egy egyértelműen meghatározott (programozott) kimeneti jelkombinációval válaszol (4.11. ábra). Ilyen értelemben a kódoló-dekódok is "olyan ROM-nak tekinthetők, amelyek az adott bemeneti jelkombinációk hatására "emlékezetből" kiadják a megfelelő kimeneti kombinációkat (pl. BCD-re a 7 vonal jelét). Más kérdés, hogy az "egyszerűbb" kódolási feladatokat nem érdemes ROM-ra bízni, mert ez utóbbi "univerzális" felépítése miatt bonyolultabb szerkezetű, nagyobb a fogyasztása, drágább. "Bonyolultabb" esetekben viszont a ROM a megoldás (pl. szám-betű, azaz alfanumerikus kijelzéshez "dekódolóként" mindig "karakter generátor ROM"-ot használnak). A ROM-ot még akkor is érdemes alkalmazni, ha nem használjuk ki "teljes tudását". Ha n a bemeneti vezetékek (cím vezetékek) száma, akkor

$$2^n$$

számu lehetséges kombináció, vagyis memória rekesz van, ennyiféle kimeneti kombinációt képes a ROM előállítani (pl. 6 bemenet esetén $2^6 = 64$ kimeneti jelkombinációt "programozhatunk bele").

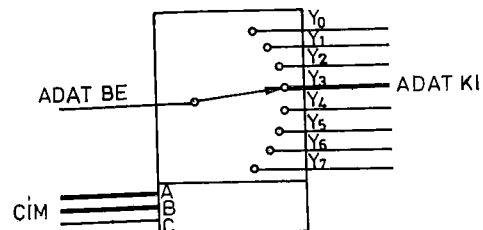


4.11. ábra

4.2.2. Demultiplexerek, multiplexerek

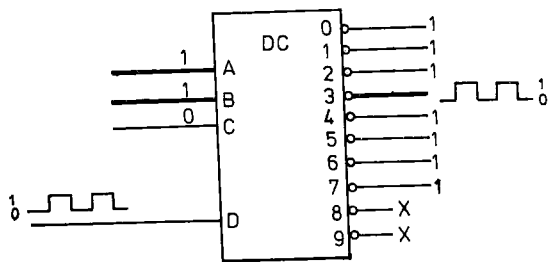
- Demultiplexerek

Funkciójuk nagyon hasonló az "1 az N közül" dekódolókéhez (olyannyira, hogy gyakran helyettesíthetők is egymással). A bemeneti (általában bináris) számkombináció itt is egy adott kimenetet jelöl ki ("dekódol ki") a többi közül, de ezenkívül van egy (vagy több) adat-bemenete is (esetenként engedélyező bemenetként is jelölhetik). Ezzel a "többlet" bemenettel a teljes demultiplexer vezérelhető, tiltható. Így olyan egységként kezelhető, amely a bemeneti "cím" által kiválasztott kimenetre kapcsolja a bemenetre adott vezérlőjelet, úgy mint egy több állásu választó kapcsoló (4.12. ábra, a vastag vonal azt a helyzetet mutatja, amikor a demultiplexer a 3-as "címet" kapja: ekkor a közös bemenetet a 3-as számú kimenettel köti össze.



4.12. ábra

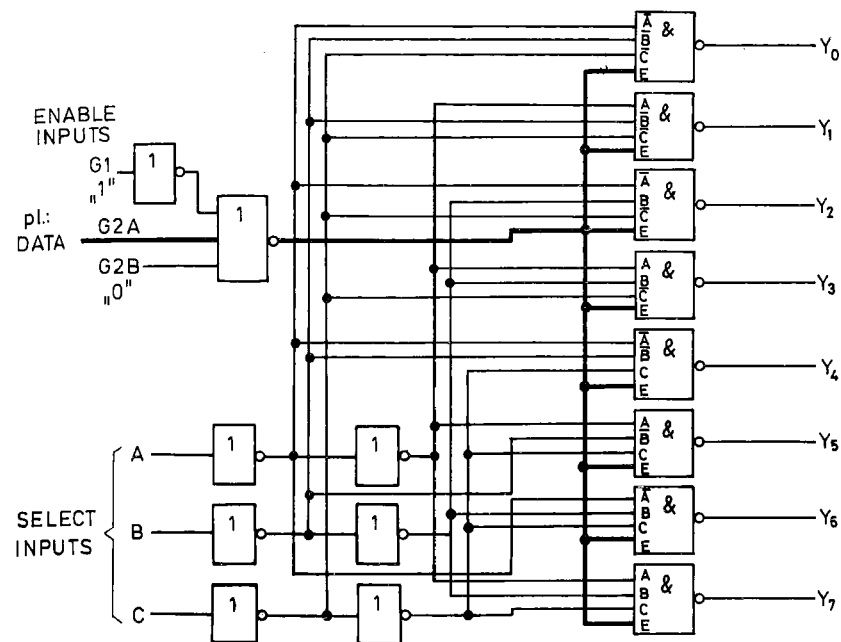
A bipoláris demultiplexer változatok több állású mechanikus kapcsolóhoz hasonlítása csak jelképes: az összeköttetés csak digitális jelekre "érvényes" és csak egyirányban, a jelbemenettől a kimenetig. A TTL-ben "alap tipusként" a már megismert "1 of 8", ill. "1 of 10" dekódolók használhatók (a katalógusokban általában a dekódolók és demultiplexerek közös címszó alatt szerepelnek: "Decoders/Demultiplexers". Például a már említett BCD-decimális 7442 (45) dekódoló 8 kimenetű demultiplexerként használható: az A, B, C a cím bemenet, D az adat bemenet (D: Data, 4.13. ábra).



4.13. ábra

Ha ez 0, akkor az A, B, C bináris cím által kijelölt kimenet 0 lesz (a dekódoló működik), ha D = 1, akkor 0-tól 7-ig mindegyik kimenet biztosan 1-ben lesz (a bemeneti "szám" 8-nál nagyobb) közöttük az is, amelyiket kiválasztottuk. (Felmerülhet a kérdés: az utóbbi esetben mindegyik kimenet 1, akkor hol van itt a demultiplexelés? A válasz: az A, B, C bemenet által kiválasztott - pl. 3-as - kimeneten valóban 1 van, ha D = 1, viszont csak ennek az egynek változik 0-ra a jele, ha D = 0 lesz, vagyis ahogy a D adatbemeneten alakul a jel, pontosan

ugy alakul a kiválasztott - pl. 3-as - kimeneten is. Ha pl. négyesjegyű adunk D-re, akkor pontosan ez jelenik meg az egyetlen kiválasztott kimeneten is, a többi "nyugalmi" 1-ben marad.

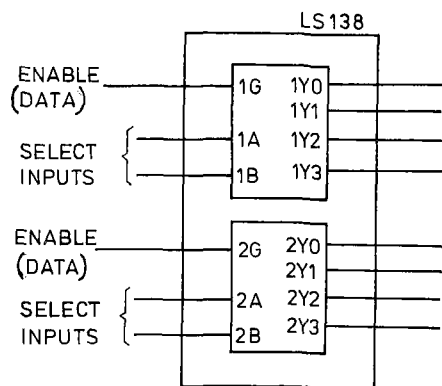


LS139, S139

4.14. ábra

A TTL-ben kimondottan demultiplexer célra készült IC a 8 kimenetű 74LS138: ez is aktív 0-val működik és a jó felhasználhatóság kedvéért 3 adatbemenete is van (egyezik a 3205-tel). Belső kapcsolási rajzát a 4.14. ábra mutatja: tartalmaz egy teljes "1 a 8 közül" dekódolót (a 4.4. ábrán látható BCD-decimálishoz hasonló), de minden egyes dekódoló NAND kapu bemenetéhez elvezetik a tiltójelet, amelyet a bemeneti adatjel(ek)ből állítanak elő. Legegyszerűbb esetben a G2A vagy G2B bemenetet használhatjuk adat bemenetnek (miközben a nem használt 0 szinten van, a G1 pedig 1-en). Ha a beérkező adat 0, akkor valamennyi NAND kapu engedélyezve van és a cím által meghatározott kimeneten 0 lesz. Ha a bejövő jel 1, akkor va-

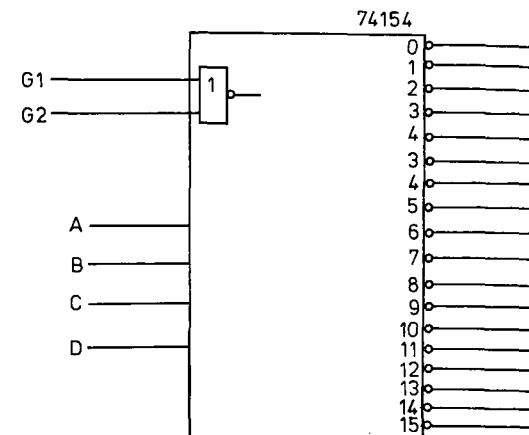
lamennyi NAND kapu tiltott állapotban lesz a mindegyikhez odaérkező 0 szint miatt, így valamennyi kimenet 1-be kerül. A végeredmény ebben az esetben is az, hogy a cím által kiválasztott kimenet jele pontosan követi az adat bemenet jelét, az összes többi kimenet 1-ben marad. Gyakran használt TTL demultiplexer típusok még pl. a 74 LS 139, S139 (normál változatban nem létezik) ez egy tokban 2 db 4 kimenetű demultiplexer, mindegyiknek 1 adat bemenete van (1G ill. 2G, 4.15. ábra), valamint pl. a 74154, amely 4 vonalról 16 vonalra működő dekódoló/demultiplexer kettős adatbemenettel (hasonlóan az előzőkhöz, a nem használt adat-bemenetet 0-ra kell kötnünk, 4.16. ábra).



4.15. ábra

A bipoláris ECL-ben is készítenek dekódoló/demultiplexer IC-eket pl. a 10K sorozatban a 10161-es, ill. 162-es típusu "1 of 8" aktiv 0k, ill. aktiv 1 változatot, vagy a 10171, ill. 172-es típusu "1 of 4" aktiv 0, ill. aktiv 1 változatot. Késleltetési idejük a cím, ill. az adat bemenettől a kimenetig 4 ns! (A TTL S változatok késleltetési ideje körülbelül ennek kétszerese.)

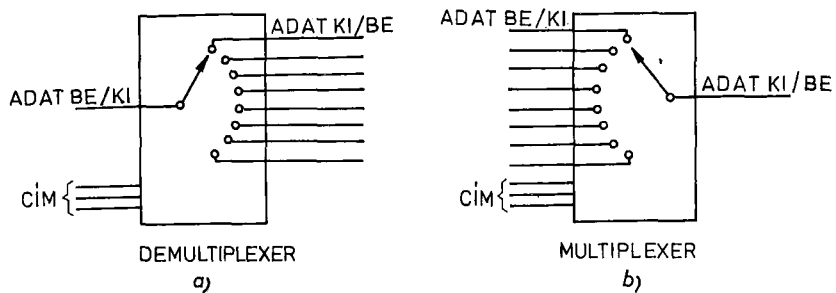
A CMOS demultiplexerek szerkezete általában másmilyen, jobban hasonlítanak a 4.12. ábra többállású kapcsolójához. - Ezekben ui. jelkapcsolásra a CMOS speciális áramkörét az "átteresztő kaput" ("transmission gate"-et) használják. Ezek - mint annak idején láttuk -, kétirányú kapcsolók (bilateral switches), amelyek a jelet vezérelhetően vagy átengedik, vagy



ha $G1 = G2 = 0$, akkor a címezett kimenet = 0
máskülönben az összes kimenet = 1

4.16. ábra

nem. Ezért az ezek segítségével felépülő CMOS demultiplexerekben a jel az éppen kiválasztott kapcsolón keresztül kétirányban haladhat, nincs kitüntetett irány: a közös pont lehet bemenet és a kapcsoló szegmensei lehetnek kimenetek, de használhatjuk úgy is az áramkört, hogy a szegmenseket tekintjük bemenetnek, a közös "leszedő érintkezőt" kimenetnek. Az előbbi esetben ugyanez az áramkör demultiplexer, az utóbbi esetben multiplexer feladatot lát el (4.17. ábra a, ill. b). Az ilyen felépítésű áramköröket a katalógusok multiplexer/demultiplexer címszó alatt ismertetik. Fontos tudnunk, hogy - amint erről már szó volt - a transmission gate-tel felépülő multiplexer/demultiplexerek analóg jelek kapcsolására is alkalmasak. Felhasználható pl. feszültségosztóban range váltó ellenállások kapcsolására, digitálisan vezérelhető erősítésű erősítők visszacsatoló ellenállásainak átkapcsolására, több csatorna analóg jelének kiválasztására vagy egy csatorna "szétosztására" stb. Egyedüli és alapvetően fontos korlátozás, hogy az analóg jelnek a pozitív és negatív tápfeszültség tartományon belül kell maradnia. "Legegyszerűbb", multiplexer-demultiplexer célra használható áramkör a már ismertetett 4016 ti-

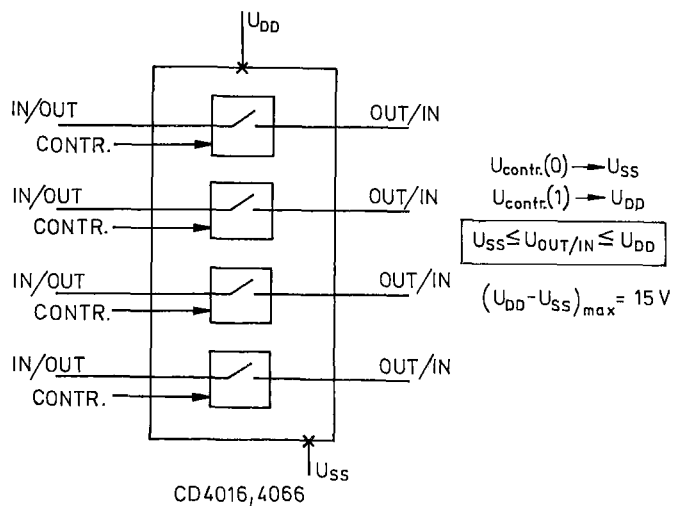


4.17. ábra

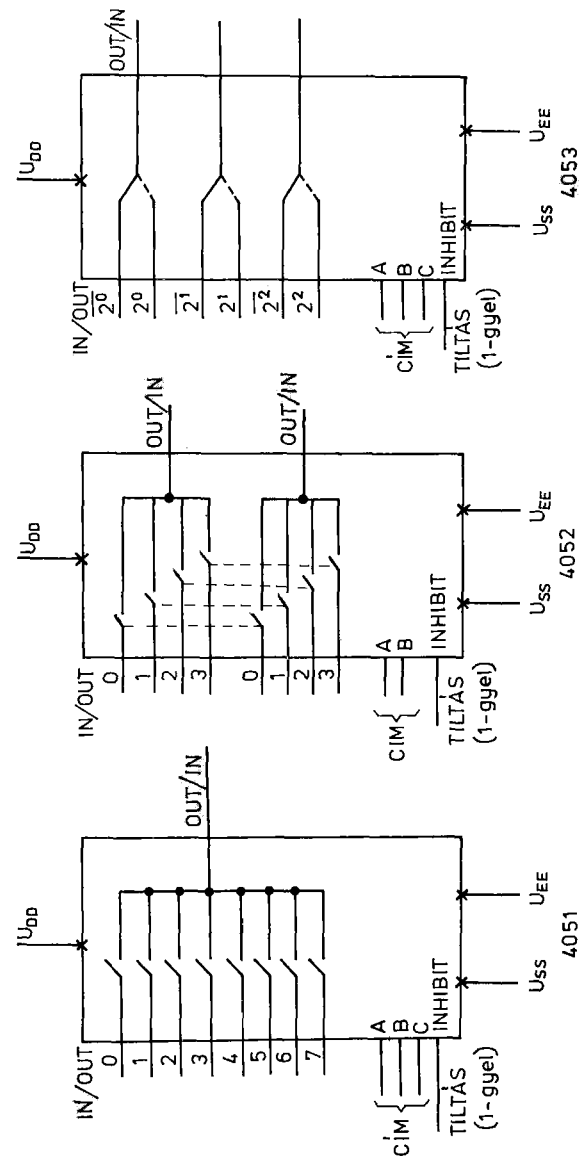
pusu 4 csatornás kétirányu kapcsoló, ill. ennek újabb generációs változata a 4066. Ezekben nincs dekódoló, a 4 db kapcsolót 4 vezérlő bemenettel kell egyenként működtetni. A kapcsolók ki/bemenetei is függetlenek egymástól, tetszés szerinti elrendezést alakíthatunk ki belőlük. Egy kapcsoló részletes rajzát a 3.85. ábrán, az IC tömbvázlatát a 4.18. ábrán láthatjuk. "Fejlettebb", dekódolt típusok a

- CD 4051 (MC 14051) egyszer 8 csatornás, a
- CD 4052 (MC 14052) kétszer 4-csatornás és a
- CD 4053 (MC 14053) háromszor 3-csatornás

multiplexerek/demultiplexerek.



4.18. ábra

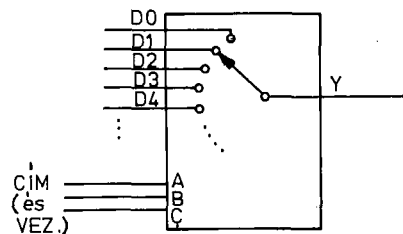


4.19. ábra

Tömbvázlatuk a 4.19. ábrán látható. Külön előnyök, hogy - analóg alkalmazás esetén - pozitív-negatív jelfeszültséget is kapcsolhatunk velük, miközben a vezérlő bemenetekre digitális szintű (0 V-hoz földelt) jelet adhatunk: a digitális szintek U_{DD} (1) és a 0 V (0) feszültségük lehetnek, miközben a kapcsolt analóg jel a pozitív U_{DD} és a negatív U_{EE} tápfeszültség között lehet (pl. ha $U_{DD} = +5$ V, $U_{SS} = 0$ V, $U_{EE} = -5$ V, akkor a logikai szint: 0 V és 5 V, az analóg jeltartomány: -5 V... +5 V). Az említett leggyakrabban használt típusokon kívül létezik 16 csatornás és 2x8 csatornás CMOS multiplexer/demultiplexer (4067B és 4097B) és még ezenkívül más gyárak választékában sok, analóg jel kapcsolására is alkalmas típus van (pl. SILICONIX BK...) - ezekről később még szó lesz.

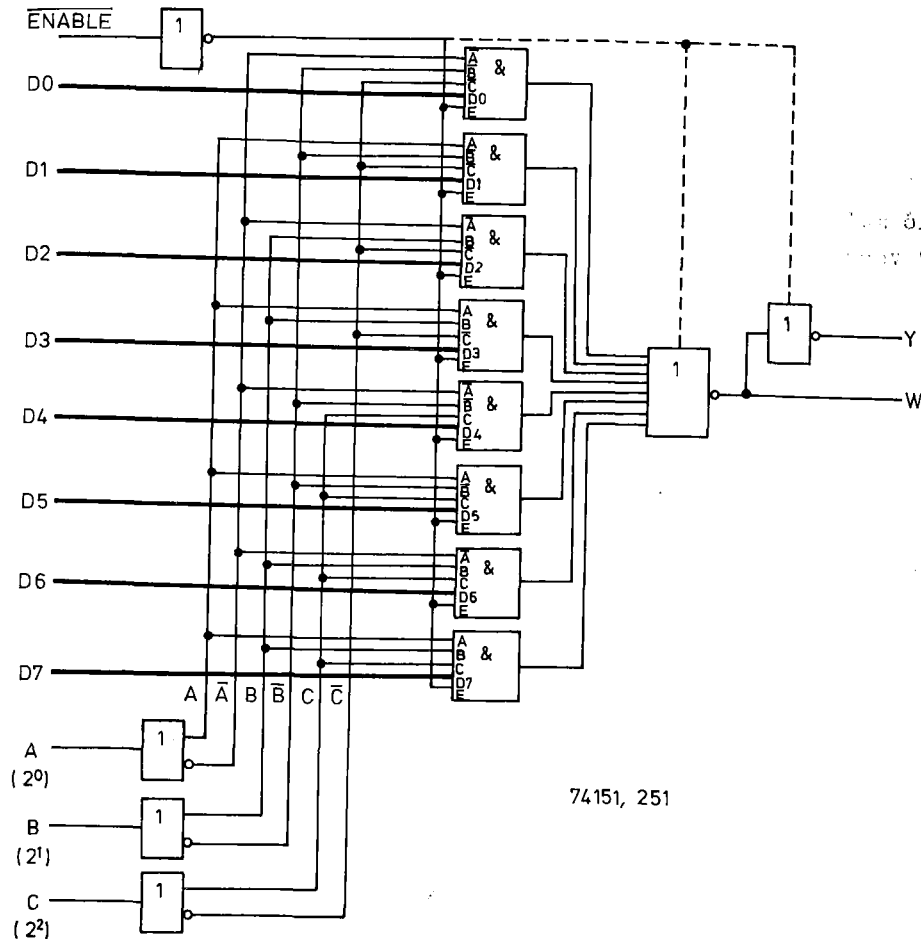
- Multiplexerek (adat-elosztók: Data Selector-ok)

A funkció az előzőekből már ismert: a több bemenet jele közül az jut a közös kimenetre, amelyiket a vezérlő cím-jellel kiválasztottunk (4.20. ábra).



4.20. ábra

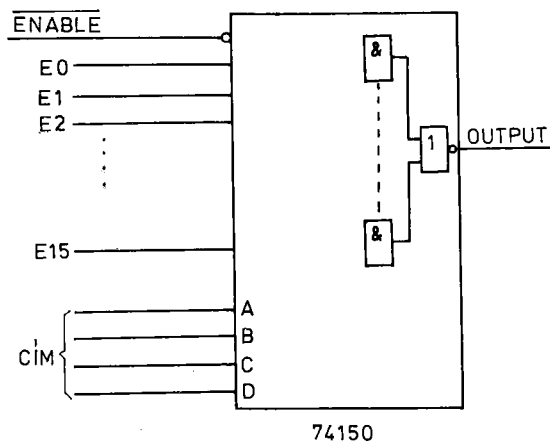
A bipoláris TTL változatok közül (amelyek csak digitális jelre működnek) legismertebb a 8 vonalról 1 vonalra (8-LINE-TO-1-LINE) dolgozó 74151A (LS,S151), ill. ennek 3-state kimenetű változata a 74251 (LS,S251) - ezek belső kapcsolási rajzát mutatja a 4.21. ábra. A megfelelő bemenet kiválasztásához ebben az esetben is "teljes dekódolásra" van szükség, ezt végzi az ÉS kapu sor. A 0 című D0 bemenetet akkor kell engedélyezni, ha a beérkező cím bináris 0, azaz, amikor \bar{A} és \bar{B} és \bar{C} egyaránt 1-es. Az első ÉS kapu tehát ezeket a jeleket kapja, valamint magát a D0-át, ezenkívül az egész áramkör enge-



4.21. ábra

délyező jelét. A következő kapu D1-et engedi át, ha a cím bináris 1-es ($A = 1$ és $\bar{B} = 1$ és $\bar{C} = 1$ és az engedélyező vezeték 1-es) és így tovább. Egyszerre mindig csak egy kiválasztott ÉS kapu kimenetén jelenhet meg 1-es akkor, amikor a hozzá tartozó D adat bemeneten is 1-es van. Ezeket "gyűjti össze" a közös VAGY kapu (negált kimenettel, hiszen TTL MSI-ben a szerkezet szokás szerint AND-OR-INVERT). A negált kimenet ki is van vezetve, ez a W, a ponált "vissza-invertált" kimenet az Y. A 3-state változatban az ENABLE-re adott 1 nemcsak az ÉS kapukat tiltja, hanem a kimeneteket is Hi-Z állapotba viszi

(szaggatott vonal). Hasonló felépítésű a 16 vonalról 1 vonalra dolgozó 74150 - ennek természetesen 4 cím-kijelölő bemenete van, azonban (vigyázat!) csak negált kimenete (4.22. ábra), amelyen a kiválasztott E adat vonal jelének negáltja jelenik meg vagy tiltáskor H szint. Kettős 4 vonalról 1 vonalra működő multiplexer a 74153 (3-state változat: 74LS353) és négyes 2 vonalról egy vonalra választó ("morze kapcsoló") a 74157, 158 (3-state változat a 74LS258A). Vannak változatok, amelyek a bejövő adatot tárolják is (74298, LS398).



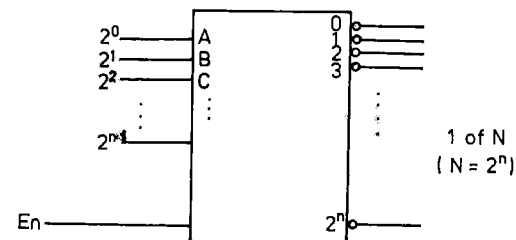
4.22. ábra

Az ECL MSI áramkörök között is van multiplexer pl. 8-ról 1-re: 10164, kettős 4-ről 1-re: 10174.

A CMOS-ban - ahogyan azt már láttuk - a bilaterális kapcsolóval felépített típusok egyformán használhatók multiplexer/demultiplexer célra. Vannak azonban "normál" kapukból felépített CMOS multiplexerek is, amelyeknél az adat kimenet nem cserélhető fel a bemenettel és csak digitális jelet dolgoznak fel. Ilyen kettős 4 bemenetű multiplexer pl. az MC 14539B. Igazság szerint ide sorolható a már említett "kapu különlegeség" a 4019-es négyes ÉS-VAGY választó kapu is, rajzát a 4.1. ábrán találjuk meg. Ehhez nagyon hasonló az MC 14519B, működése csak annyiban tér el, hogy akkor, amikor mindkét vezérlő bemenet 1-ben van, a kimeneteken a bemenet-párok ekvivalenciafüggvénye jelenik meg.

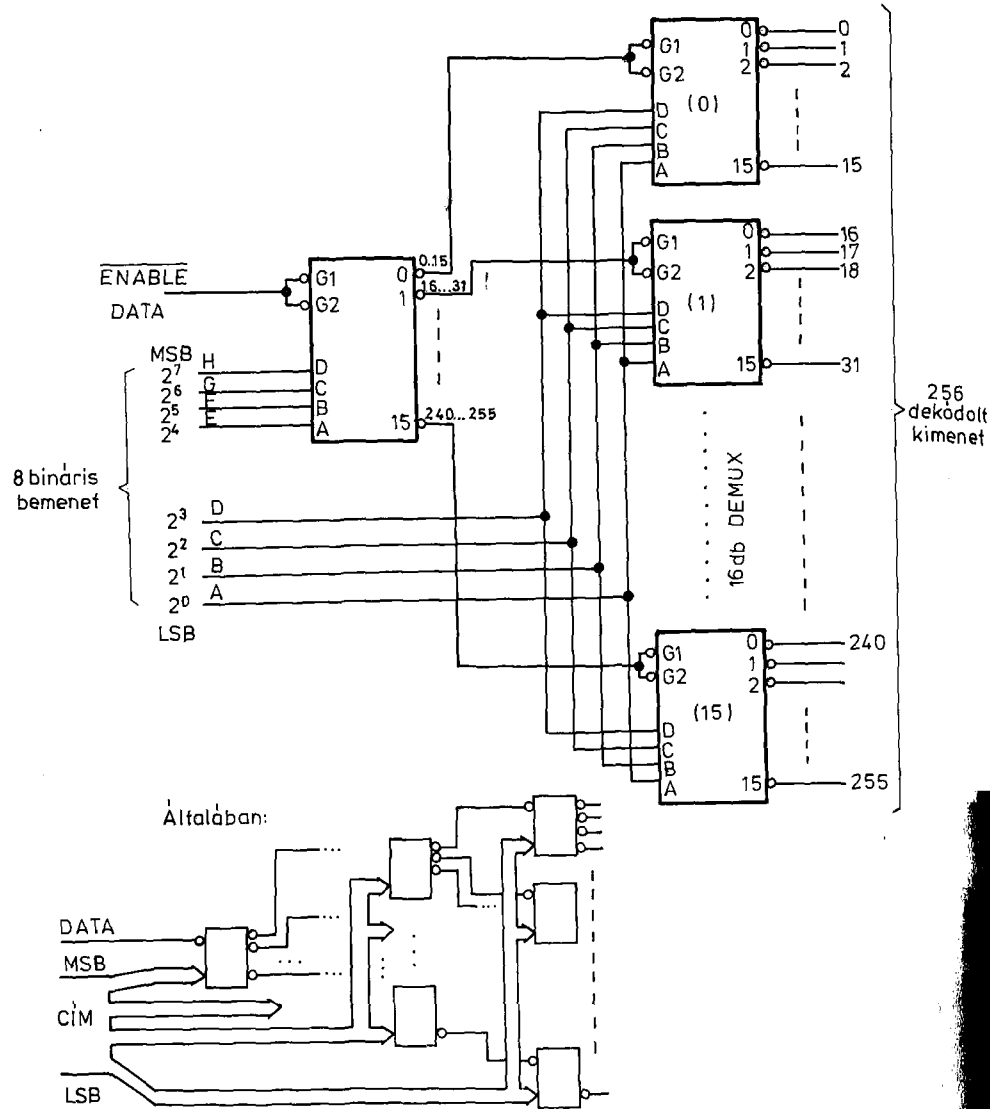
- A demultiplexerek és multiplexerek bővítése

Gyakran előfordul, hogy a rendelkezésre álló kimenet, ill. bemenet-szám kevés, ilyenkor azonos elemek ismételt alkalmazásával bővítenünk kell az áramkört.



4.23. ábra

Az ún. "teljes dekódolás" az a feladat, ami rendszerint nagyszámú bővítést igényel (4.23. ábra). Tegyük fel, hogy egy 256 kimeneti áramkört kell készítenünk. Azt, hogy a 256 kimenet közül melyik legyen egyedül aktív, a hálózat bemenetére adott bináris cím dönti el. Szükséges ezenkívül egy tiltó (vagy "adat") bemenet is, amely az összes kimenetet letiltja, ill. a kiválasztott kimenetet 0-ba vagy 1-be vezérelheti. A 256 állapotot 8 bittel jelölhetjük ki, de ilyen, 8 bemenetű - 256 kimeneti áramkör készen nem kapható. Válasszunk 4 bemenetű, 16 kimenetű demultiplexert építő elemként (pl. 74154, 4.16. ábra). A 256 kimenethez 16 db demultiplexert kell egymás mellé helyezni (4.24. ábra jobb oldala). Valamennyi áramkör párhuzamosan ugyanazt a dekódolandó cím-jelet kapja, annak "alsó" (LSB-vel kezdődő) 4 bitjét A-tól D-ig, 2^0 -tól 2^3 -ig. Természetesen így valamennyi áramkör "egyszerre" működne, ha nem használnánk a tiltó bemeneteket is (G1 és G2). Ezek akkor engedélyezik az áramkört, ha 0-ban vannak (G1 = G2 = 0). Megfelelő vezérlésükkel, egy-egy áramkör engedélyezésével 16-os csoportokban aktiválhatjuk a kimenet-sort: az "alsó" 4 bit-nek 16 kombinációja van, 16-onként "ismétli önmagát". Arról kell gondoskodnunk, hogy az első 16 "esetben" csak a legelső áramkör lépjen működésbe (0...15), azután a következő 16-os csoportot dekódoló áramkör (16...31) és így tovább. A 16-os csoportokat



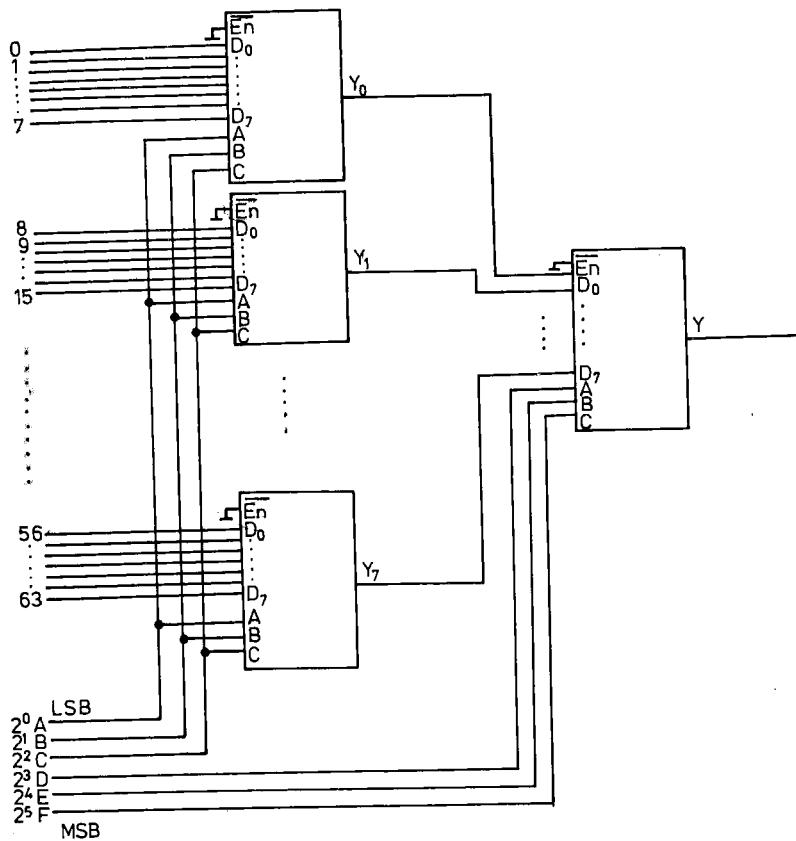
4.24. ábra

az alsó 4 bit "feletti" bitek jelölik ki, ezeket dekódolja a sor elé helyezett újabb 16-os demultiplexer. Kimenetei engedélyezik sorra az egymás alatti demultiplexereket. Például 0000 0000-tól 0000 1111-ig (0-tól 15-ig) a legfelső demulti-

plexer működik, hiszen az EFGH biteket dekódoló áramkör a cím bemenetein 0000-át "érez", ezért "0"-ás kimenetére ad logikai 0-át, az összes többi kimenete "nyugalmi" 1-ben van, ami tiltja az összes többi, egy oszlopban levő áramkört. A 16-os számnál már E = 1 lesz (0001 0000), így az MSB biteket dekódoló áramkör "1"-es számú kimenetén jelenik meg logikai 0, ezért az oszlopban a következő demultiplexert engedélyezi (16-tól 31-ig) és így tovább. Az összes eddig áttekintett engedélyezés természetesen csak akkor megy végbe, ha a legelső engedélyező bemenet (ENABLE) is aktív, különben mindegyik kimenet 1-ben marad. Ha erre a bemenetre nem engedélyező jelnek tekintett 0 szintet adunk, hanem valamely bemeneti adatot (Data), akkor a 8 bites cím által kiválasztott kimenet logikai szintjét vezérelhetjük vele (a többi 1-ben van - demultiplexelés).

Az ismertetett elv tetszés szerinti bit-számra kiterjeszhető, a "fa áramkör" tovább építhető, több szintre bővíthető (és természetesen nemcsak 1 of 16 áramkörök felhasználásával). Olyan eset is előfordulhat, hogy nincs szükségünk a "teljes dekódolás" eredményére, csak bizonyos tartományokra: ilyenkor az áramkörből csak a szükséges részletet építjük ki (helyenként esetleg csupán kapuk felhasználásával).

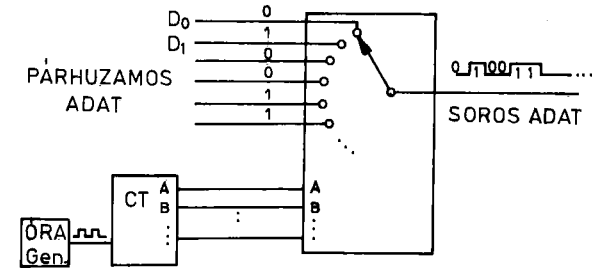
Multiplexerek esetében is hasonló a bővítés elve, csak természetesen - mivel most sok bemenet és egy kimenet van - "fordított fa"-áramkört kell összeállítanunk. A 4.25. ábrán 8 bemenetű multiplexerekből (pl. 74151) egy 64-ről 1 vonalra dolgozó áramkörre látunk példát. A címet kijelölő bitvezetékek közül a 3 LSB vezeték mindegyik multiplexer számára közös, miáltal valamennyi "első vonalban" levő multiplexer kimenetén ($Y_0 \dots Y_7$) megjelenik a 3 LS bit által kijelölt bemenet jele. Ezekből azonban a "gyűjtő" multiplexer már csak egyet választ ki a "maradék" 3 MSB cím-bit értékétől függően, vagyis mindig csak egy kiválasztott bemenet jele jut a kimenetre. Az elv itt is tetszőleges bit számra, ill. bemenet számra alkalmazható, csupán arra kell figyelni, hogy a felhasznált multiplexer típus nem ad-e negált adat-kimenetet (problémát ez sem okoz olyankor, amikor páros számú invertáló multiplexeren halad keresztül a jel, ha ez nem teljesül, akkor az eredményt invertálnunk kell vagy inverz kimenetet kell felhasználnunk - pl. 74151-nél a W kimenetet).



4.25. ábra

A demultiplexerek és multiplexerek legfőbb alkalmazási területei működési elvükből, funkciójukból szinte magától értetődően adódnak. Sok olyan eset van, amikor valamely bemenő vonal jelét vezérléstől függően sok vezetékre kell "szétosztani" vagy fordítva, sok bemenet jelét kell adott vezérlési sorrendben "lekérdezni", egy kimeneti vezetékre adni. Ilyenkor általában sorrendi hálózat is részt vesz a működésben: pl. párhuzamos-soros átalakításkor a párhuzamos jelet egy multiplexer bemenetére adjuk, miközben a multiplexert bináris számsorrendben cimezzük, ezzel a bemeneti bit-eket egymás után a közös kimeneti (soros) vezetékre "küldjük". A cím egyenkénti növelése egy órajellel vezérelt bináris számlálóval történik

(4.26. ábra). Mindezzel a sorrendi áramkörök tárgyalásakor részletesebben foglalkozunk. Ugyancsak később, a mikroprocesszoros rendszer felépítésének tanulmányozásakor kerülnek elő az "1 of N" dekódolással/demultiplexeléssel kapcsolatos részletek, pl. a cím-buszra csatlakozó egységek cím-dekódolásával kapcsolatban.



4.26. ábra

A multiplexereknek van egy érdekes alkalmazása, amit most, a kombinációs hálózatok tárgyalásakor érdemes megismerni. Ez a "minterm generátor" vagy "univerzális kapu" funkció. Logikai függvények megvalósításakor legtöbbször először a mintermes alakot irtuk fel: egy logikai összegben azokat a szorzatokat szerepeltettük, amelyek "részt vesznek" a függvényben (az igazságtáblázatban a hozzájuk rendelt sorban $Y = 1$). Vegyük a következő, igazságtáblázattal adott 3 változós példát:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Ebből az igazságtáblázatból a függvényt a következő együtthatós alakban is felírhatjuk:

$$Y = 0 \cdot \bar{A}\bar{B}\bar{C} + 1 \cdot \bar{A}\bar{B}C + 0 \cdot \bar{A}B\bar{C} + 0 \cdot \bar{A}BC + 1 \cdot A\bar{B}\bar{C} + 1 \cdot A\bar{B}C + 1 \cdot AB\bar{C} + 0 \cdot ABC$$

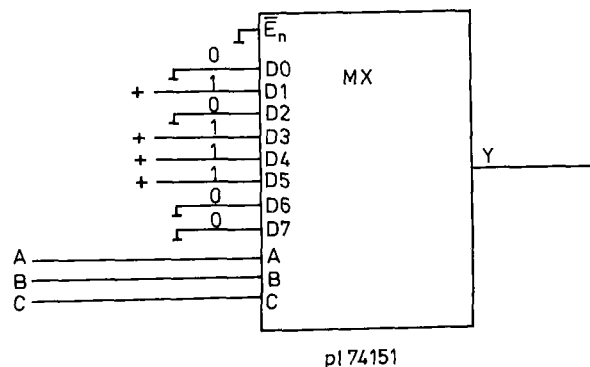
Ezután nézzük meg a 4.21. ábrán a 74151-es multiplexer belső kapcsolási rajzát: észrevevessük, hogy az előbbi függvényben levő valamennyi logikai szorzatot előállítják a benne levő ÉS kapuk. Nekünk csupán annyi a teendőnk, hogy ezeket az ÉS ka-

pukat - aszerint hogy az általuk előállított szorzat szerepel-e a függvényben vagy nem - a hozzájuk tartozó D bemenet segítségével engedélyezzük, vagy letiltjuk, 1-et vagy 0-át "írjunk" a szorzat elé. Ezek szerint a multiplexer A, B, C bemenetét használhatjuk változó bemenetként és a $D_0 \dots D_7$ bemenetek 1-re, ill. 0-ra kötésével "állithatjuk elő" a kívánt mintermeket. Arra kell csak vigyáznunk, hogy a D bemenetek indexei nem egyeznek meg a minterm számokkal, mert a D számozásában (címezésében) A az LSB (2^0 helyiérték), a mintermet pedig ABC sorrendben számozzák, ezért itt C az LSB. Jelen esetben az igazságtáblázatban A-nak 2^0 , B-nek 2^1 , C-nek 2^2 súlyt tulajdonítva:

$$D_0 = D_2 = D_6 = D_7 = 0 \quad \text{és}$$

$$D_1 = D_3 = D_4 = D_5 = 1.$$

$Y = \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + A\bar{B}\bar{C}$ előállítás:



4.27. ábra

A függvényt tehát végülis egy 74151-es multiplexerrel a megfelelő D bemenetek 1-re, ill. 0-ra kötésével, a cím bemeneteket jelbemenetként használva a 4.27. ábra szerinti kapcsolással valósíthatjuk meg. Az ilyen fajta "rendszeres" függvény-megvalósítás előnye, hogy bármely 3 változós függvényt, egyetlen IC-vel állithatunk elő, ráadásul programozhatóan. A D bemeneteket tetszés szerint köthetjük 1-re vagy 0-ra a minden-

kor előállítandó függvény igényei szerint, bármely változtatás huzalok áthelyezésével (esetleg kapcsolók átkapcsolásával) egyszerűen, utólagosan is végrehajtható! Egyszerűsítéssel itt nem kell bajlódni, hiszen minden minterm együtthajtóját "programozzuk". Hátrány, hogy 4 változó fölött már bővitenünk kell a hálózatot a 4.25. ábra szerint, ami már bonyolultabb, drágább. Minden esetre ez az első és legegyszerűbb példa logikai függvények nem "rendszeretlen" kapuzással (RANDOM-LOGIC), hanem programozható, univerzális áramkörrel való előállítására!

4.2.3. Aritmetikai elemek

Felmerülhet a kérdés: szükség van-e egyáltalán MSI aritmetikai áramkörökre, amikor mikroszámítógépekkel kalkulátorokkal egyszerűen végezhetünk aritmetikai műveleteket. A válasz: szükség van - ha nem is olyan mértékben mint régebben, és nem is ugyanolyan fajta áramkörökre, mint amelyeket hagyományosan használtak. A létjogosultságnak két alapvető oka van:

1. A mai bipoláris áramkörök nagyságrendekkel gyorsabban végeznek el egy-egy műveletet, mint a szokásos, MOS áramkörös mikroprocesszoros rendszerek vagy kalkulátor IC-k. Így, ha nagyon gyors működésű számító-egységre van szükség, akkor bipoláris aritmetikát célszerű alkalmazni (van, amikor mikroprocesszoros rendszer "mellé rendelnek" egy gyors, bipoláris aritmetikai műveletvégző egységet).

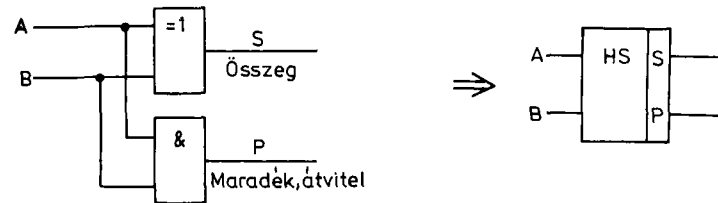
2. Az áramkör családot típusválasztékában sok olyan aritmetikai kategóriába sorolt áramkör van, amelyet célszerűen fel lehet használni digitális áramkörök építésében - kihasználva a bináris aritmetikai műveletek és a logikai műveletek közötti rokonságot. A továbbiakban a részletek mellőzésével foglalkozunk néhány jellemző példával.

- Összeadó

Két bit maradék nélküli összeadására a kizáró-VAGY (EOR) kapu alkalmas, mivel ez a bemeneti változók 0-1 értékeire pontosan olyan "választ" ad, mint ami a bit-értékek összeadásának felel meg:

$$0 \oplus 0 = 0 \quad 0 \oplus 1 = 1 \quad 1 \oplus 1 = 0$$

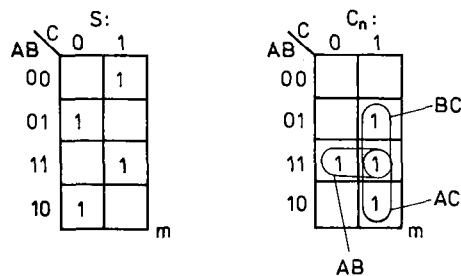
A maradék képzéséhez még egy ÉS kapu szükséges, hiszen maradék akkor keletkezik, ha mindkét összeadandó 1-es. Az így kapott áramkör egy 1 bites félösszeadó (half-adder, 4.28. ábra). Hiányossága, hogy több bites összeadás esetén nem alkalmazható, hiszen ilyenkor az előző helyiértéken keletkező maradékot, átvitelt is figyelembe kell venni és hozzáadni az eredményhez (és a keletkező maradék képzésénél is figyelembe kell venni).



4.28. ábra

A teljes összeadónak ezek szerint 3 bemenetűnek kell lennie, ez a három bemenet: az egyik összeadandó bit: A, a másik: B, valamint az előző helyiértékről érkező, ott keletkezett átvitel C_{-1} , egyszerűbben jelölve C. A kimenetek: az összeg, S_n (szokásos jele még a Σ , szumma), és az átvitel a következő helyiértékre: C_n (Carry = átvitel, szokásos még a P jel is). A bemenetek és kimenetek közötti kapcsolatot leíró igazságtáblázat és a Karnaugh-táblák a következők:

A	B	C	S	C_n
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



4.29. ábra

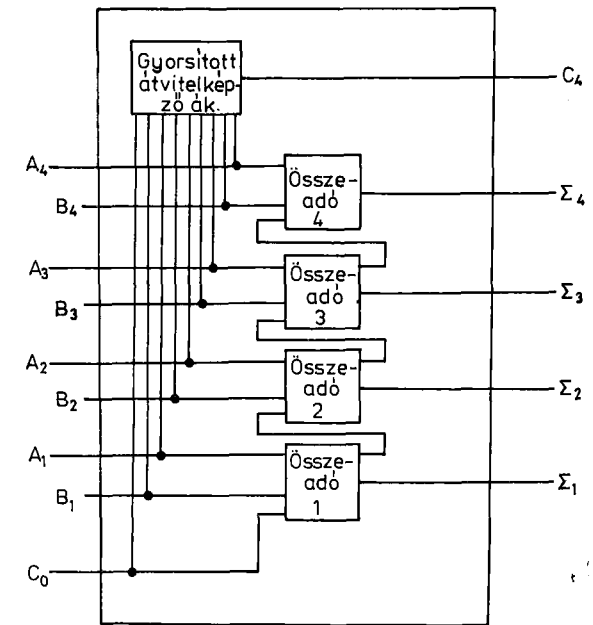
Az összeg-kimenet, S függvényét nem lehet egyszerűsíteni, csak a C_n -et, így:

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

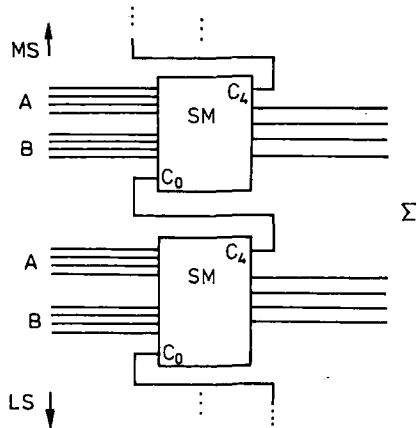
$$C_n = AB + AC + BC$$

Az S-re - mivel nem lehetett egyszerűsíteni - "megmaradt" a modulo 2 összeadást jelentő függvény, amelyet pl. kizáró-VAGY kapukkal lehet realizálni, a C_n átvitelre a majoritás logikai kapcsolat jellemző (többségben lévő 1-esek esetén a kimenet 1-es, különben 0, ezzel már találkoztunk a 2.3. fejezetben).

A TTL MSI és a CMOS MSI áramkörök típusválasztékából jellemző példaként a 4 bites bináris teljes összeadókat említjük meg, amelyek az előbb ismertetett összeadó egységekből 4-et tartalmaznak egyetlen tokban. A bitek közötti "belső" átvitel jelét nem vezetik ki, csak a legutolsó összeadás után keletkezőét (ezt, a C_4 átvitelt igazság szerint nem a 4 bit összeadásának eredményéből állítják elő, hanem gyorsított átvitel képzéssel, hiszen a bemeneti érték kombinációkból azonnal el lehet dönteni, lesz-e átvitel az utolsó bit helyén vagy nem, így "nem kell várni" a végeredményre, gyorsabb lesz a működés (l. a 4.30. ábra vázlatát).



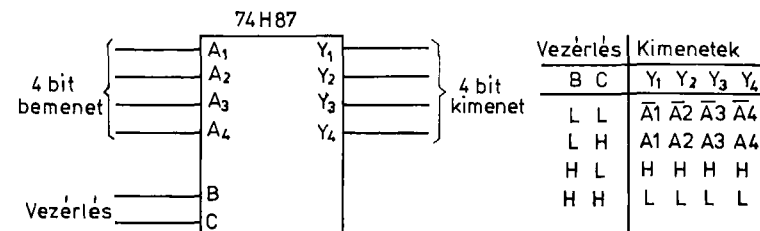
4.30. ábra



4.31. ábra

Az áramkörök figyelembe veszik a több bit összeadásakor az előző 4-es csoportban képződő átvitelt is (C_0). Az áramkör tehát összeadja a négy A bitet a négy B bittel, ehhez hozzáadja C_0 -t és az eredményt a négy Σ kimeneten adja ki, valamint előállítja az átvitelt is (C_4) a következő 4-es csoport számára. A TTL-ben legismertebb típus a 7483A, CMOS-ban a CD4008, az előbbi 16 ns, az utóbbi 325 ns alatt ad össze két négy bites bináris számot (a carry kimenet jele mindkét esetben gyorsabban áll elő: 10 ns, ill. 45 ns múlva). Többször 4 bit összeadása esetén, amikor több IC-t egymás után kapcsolunk (4.31. ábra) az összeadás lelassulhat, mert az átvitelnek egyik IC-ből a másikba tovább kell haladnia, így mire a láncon "végighullámozik" (innen a gyakori elnevezés: "ripple-carry") és módosítja az összeadás eredményét, tetemes idő telhet el. Sok bit esetén ezért, ha a sebesség kritikus, az egész rendszeren át működő gyorsított átvitel képzést alkalmazhatunk, speciális erre a célra készült "előrelátó átvitelképző" (look ahead carry) áramkörök felhasználásával (a részletekkel itt nem foglalkozunk). A kivonáshoz vezérelhető komplementáló elemre van szükség, ilyen típus a 74H87 (csak nagysebességű változatban létezik, azért, hogy járulékos késleltetés minél kisebb legyen), ez 4 bit "átengedésére" vagy komplementálására (bitenkénti invertálására), vagy L, ill. H szinttel való helyettesítésére

alkalmas. A 4 féle üzemmódot 2 vezérlő bemenettel (B,C) lehet beállítani (4.32. ábra). Alkalmazását egy 1-es komplement kódú, ill. egy 2-es komplement kódú összeadó-kivonó egységben már láttuk az 1.3.1. pontban (az 1.16. ábrán az M: "módus" vezérlés egyezik a C-vel, miközben B = 0 kell legyen).

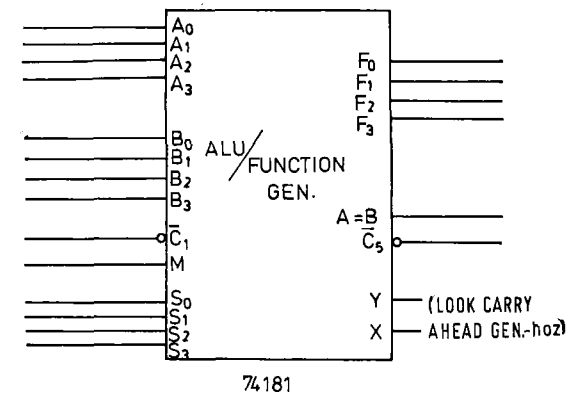


4.32. ábra

Az összeadó-kivonó egységnek sokoldalabb áramkör az

- Aritmetikai logikai egység/függvény generátor (ALU/Function Generator).

Példa áramkör a bipoláris 74181 (CMOS-ban hasonló típus az MC 14581B). Kétszer 4 bemenetére érkező jelen 16 logikai és 16 aritmetikai műveletet képes elvégezni (4.33. ábra).

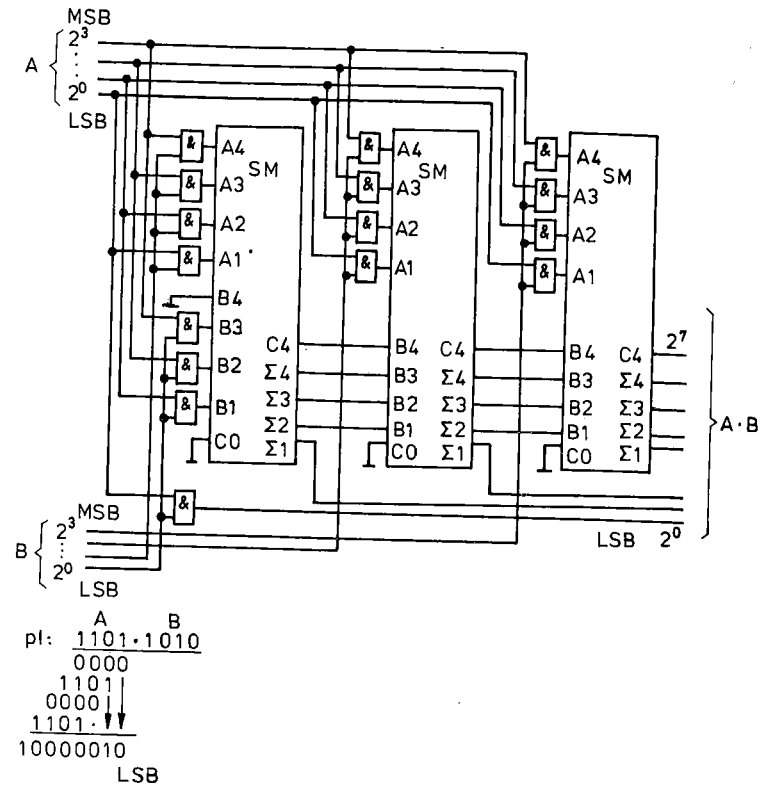


74181

4.33. ábra

A műveletek az S0, S1, S2, S3 vezérlő bemenetekkel jelölhetők ki, valamint az M (Mode control) jellel, ez választ a logikai vagy az aritmetikai műveletek csoportjából:

SO S1 S2 S3	M = H	M = L	
	LOGIKAI MŰVELETEK	ARITMETIKAI MŰVELETEK C ₁ = H	ARITMETIKAI MŰVELETEK C ₁ = L
L L L L	$F = \bar{A}$	$F = A$	$A + 1$
L L L H	$\overline{A + B}$	$A + B$	$(A+B)+1$
L L H L	$\bar{A} B$	$A + \bar{B}$	$(A+\bar{B})+1$
L L H H	0	-1(2-es kompl.)	0
L H L L	$\overline{A \cdot B}$	$A + A\bar{B}$	$A+A\bar{B}+1$
L H L H	\bar{B}	$(A+B)+A\bar{B}$	$(A+B)+A\bar{B}+1$
L H H L	$A \oplus B$	$A-B-1$	$A-B$
L H H H	$A\bar{B}$	$A\bar{B}-1$	$A\bar{B}$
H L L L	$\bar{A} + B$	$A+AB$	$A+AB+1$
H L L H	$A \odot B$	$A + B$	$A+B+1$
H L H L	B	$(A+\bar{B})+AB$	$(A+\bar{B})+AB+1$
H L H H	AB	AB-1	$A \cdot B$
H H L L	1	$A + A^{\bar{x}}$	$A+A+1$
H H L H	$A + \bar{B}$	$(A+B)+A$	$(A+B)+A+1$
H H H L	$A + B$	$(A+\bar{B})+A$	$(A+\bar{B})+A+1$
H H H H	A	A-1	A



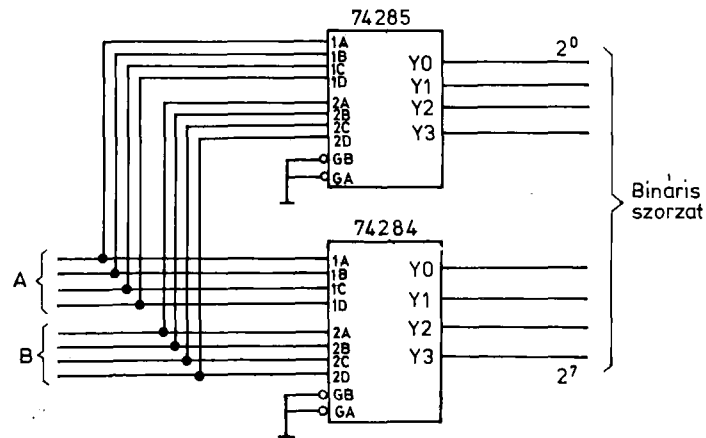
4.34. ábra

Az áramkörhöz a 74182-es Look Carry Ahead Generator, gyors átvitelképző csatlakoztatható előnyösen, amivel a műveleti idő nagymértékben lecsökkenthető. Az ECL családban az előbbi ALU-hoz hasonló nagysebességű 4 bites változat az univerzális aritmetikai építőelem (10181).

- Szorzók (párhuzamos, bináris: Parallel Binary Multiplier)

Párhuzamos bitek közvetlen összeszorozását és kapukkal és az így keletkező "részlatszorzókat" egy-egy helyiérték eltolással összeadó teljes összeadó áramkörökkel is elvégezhetjük. A 4.34. ábra egy ilyen 4x4 bites szorzót mutat (TEXAS ajánlás).

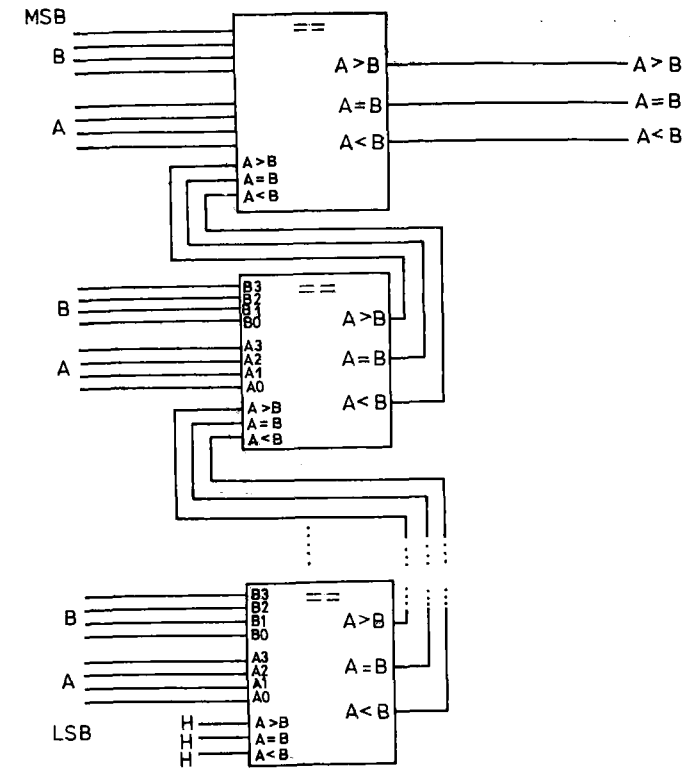
A TTL MSI típusválasztékban létezik közvetlen szorzó IC pár is: 74284-285. Ezzel a párral 4x4 bitet lehet közvetlenül összeszeszorozni, a 8 bites eredmény 40 ns alatt áll elő (Texas javaslat rajza a 4.35. ábrán). A több bitre való kiterjesztés hatványozottan növeli a szükséges IC-számot (l. a katalógust). Érdemes megjegyezni, hogy a szorzó MSI-k között vannak ún. "impulzus-üzemű szorzók" (RATE-MULTIPLIER-ek), amelyek egy bejövő impulzussorozat frekvenciáját ("átlag gyakoriságát") szorozzák meg egy 1-nél kisebb számmal. Ezzel a kategóriával a sorrendi hálózatoknál megismerkedünk majd.



4.35. ábra

- Digitális összehasonlító (Magnitude Comparator)

Áramkör építéshez jól használható univerzális elem. TTL-ben a 7485 (LS85, S85) a használatos típus. Két 4 bites bináris (vagy BCD) "szót" hasonlít össze ($A_0...A_3$ és $B_0...B_3$) és az összehasonlítás eredményét 3 kimenet ($A > B$, $A < B$, $A = B$) valamelyikének H-ba vitelével adja ki. Ezenkívül 3 ugyanilyen reláció-jelű bemenete is van annak érdekében, hogy az áramkört négynél több bit összehasonlítására bővíteni lehessen. Az összehasonlítást mindig az LSB-nél kell kezdeni, az ezen biteket összehasonlító áramkör $A > B$, $A < B$, $A = B$ kimeneteit kell a magasabb helyiértéken lévő összehasonlító áramkör ugyanazon megjelölésű bemeneteihez kötnünk, a végeredmény az MSB-t összehasonlító IC kimenetein áll elő (4.36. ábra). A digitális komparátor alkalmazásával a későbbiekben többször fogunk találkozni (programozható számlálókkal, impulzusszélesség-modulátorokkal, pontos alkatrészeket alig igénylő digitál-analóg átalakítókkal kapcsolatban). CMOS-ban hasonló típus az MC 14585B és természetesen a 74C85 (ennél vigyáznunk kell, mert bekötése nem a normál, hanem a 74L85 TTL megfelelőjének bekötésével egyezik).



4.36. ábra

- Paritás generátorok/ellenőrzők (Parity generator/checker áramkörök)

Rendszerint ezeket is az aritmetikai elemek közé sorolják. 9 bit párosság/páratlanság (EVEN/ODD) vizsgálatát képes elvégezni pl. a 74180 típus (ha a bemeneteken lévő 1-esek száma páros, akkor EVEN, ha páratlan, akkor ODD kimeneten jelenik meg aktiv jel). 11 bites "párosság fa áramkör" (Parity tree: kizáró-VAGY kapukból felépülő négyszintes hálózat) a CMOS MC 14531B. (A "paritás generátor" funkciót úgy kell érteni, hogy ha pl. a bemeneteken az 1-ek száma nem páros, akkor a páratlanság-kimeneten keletkező bitet használhatjuk párosra kiegészítő paritás-bit gyanánt). Gyors működésű paritás generátorok és ellenőrzők az ECL típusválasztékában is vannak (10160, 10170).

- Egyéb aritmetikai elemek

Az aritmetikai elemek közé sorolják legtöbbször a kizáró-VAGY kapukat is (négyyszer 2 bemenetű a 7486, LS86, ..., az open collector-os a 74136, LS136), a kizáró-NemVagy kapukat (ENOR: 74LS266), valamint a már említett vezérelhető komplemantáló elemet a 74H87-et.

4.3. Kombinációs hálózatok megvalósítása LSI-vel

Az eddigiekben is szó volt arról a követendő alapelvről, hogy az adott feladatot a lehető legnagyobb mértékű integrációval készülő áramkörökkel valósítsuk meg: leginkább LSI-vel, ha nincs ilyen, akkor MSI áramkörök "ügyes" összekapcsolásával és csak, ha szükséges, akkor külön kapukkal, SSI áramkörökkel. Az SSI és az MSI kombinációs áramkörökről az előző pontokban már szó volt, most az LSI-vel való realizálás lehetőségeit vizsgáljuk meg (a részletes ismertetés a későbbiekben egy külön fejezetben található meg a következő kötetben).

Nagyon sokszor előfordul, hogy egy adott feladat áramköri megvalósításához nem érdemes hozzálátnunk, mert valamely gyár típusválasztékában kész LSI egység van (vezérlések, műszer áramkörök, mikroszámítógép-kiegészítő elemek stb. - ezek természetesen nemcsak a kombinációs hálózat-részt tartalmazzák, hanem a szükséges sorrendi elemeket is, regisztereket, számológát, stb.). Az is lehet, hogy a kapható LSI áramkör "nem pontosan" olyan, mint amilyen a mi feladatunk megoldásához szükséges, ilyenkor meg kell kísérelnünk a feladat módosításával, átdolgozásával figyelembe venni a gyakorlati lehetőségeket, az áramkör kínálatot.

Igen nagy sorozatszám esetén kifizetődő lehet az is, hogy valamely gyárnál megrendelünk egy céljainknak megfelelő LSI integrált áramkört. Egy új IC típus kifejlesztése természetesen nagyon drága, igen nagy a kockázat a "megtérülést" illetően (az ilyen áramköröket szokás CUSTOM-DESIGNED, felhasználó által tervezett IC-nek nevezni).

A ma legjobban elterjedt, leginkább gazdaságos megoldás az, hogy az integrált áramköröket gyártó vállalatok olyan "egy-

séges", adott feladattal "előre meg nem bizott" ("uncommitted") univerzális, sok egységet magában foglaló LSI áramkör-családokat hoznak ki, amelyeket a felhasználó - adott szabályok szerint - "testre szabhat". Ez történhet úgy is, hogy az IC-ben lévő áramkör-készlet figyelembevételével a felhasználó megküldi a gyárnak a "szabásmintát", vagyis az elemek összekötésére vonatkozó utasítást, programot, aminek alapján a gyártó az IC legutolsó rétegét, az összekötő fémezést elkészíti, létrehozva ezzel a kívánt áramkört. Ez a megoldás még mindig drága, csak kellő sorozatnagyságnál fizetődik ki. Másik, kisebb darabszámnál kifizetődő lehetőség az, hogy a gyár által elkészített áramkör "minden lehetséges összekötést" tartalmaz, a végleges áramkört a felhasználónak "helyszínen végzett programozással" ("field programming"), a felesleges összekötések megszüntetésével, "leégetésével" kell kiiktatnia.

Az említett áramkörök legismertebb változatai, elnevezései:

- ULA: Uncommitted Logic Array: "feladattal előre meg nem bizott" logikai elrendezés;
- PLA (PAL): Programable Logic Array: programozható logikai elrendezés ("maszk-programmal"), Programable Array Logic: programozható "rendezett" logika;
- FPLA: Field Programable Logic Array: helyszínen programozható logikai elrendezés;
- BOÁK: Berendezés Orientált Áramkörök, ez a magyar gyűjtőnév (amig jobb nincs).

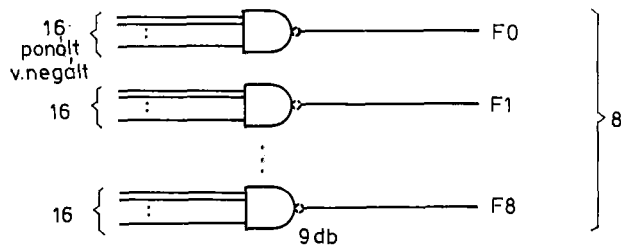
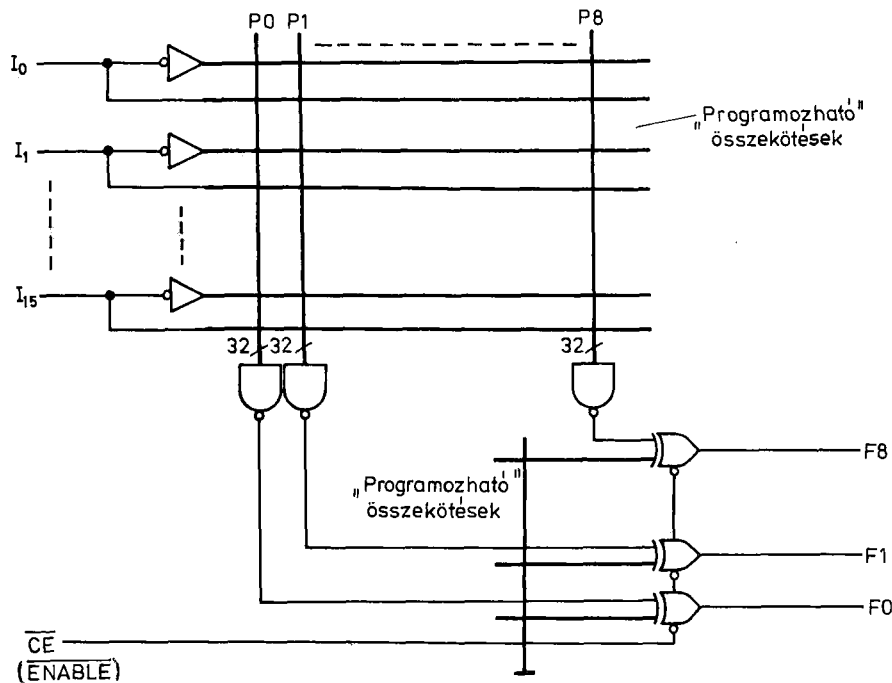
Az áramkörök - gyártótól és tervezett felhasználási területtől függően - készülnek MOS (ma már CMOS) és bipoláris (TTL, I²L) technológiával is. Az utólag programozható változatokban az "összes lehetséges helyen" levő belső összekötésekbe mindenhol beiktatnak egy-egy kis keresztmetszetű hidacsát, anyaguk nikkal, ezek azok, amelyeket "égetni kell" a programozás során, megfelelő készülékkel úgy, hogy csak a szükséges áthidalásokat hagyjuk meg.

Ennek az áramkör típusnak a bemutatására válasszunk egy jellemző példát, a SIGNETICS FPLA családját.

- Az FPLA szerkezete, működési elve

A legegyszerűbb változat az FPGA (Field Programmable GATE Array: helyszínen programozható KAPU elrendezés) ilyen például a bipoláris (Schottky):

82S102 (open collector-os kimenettel),
82S103 (three-state kimenettel).



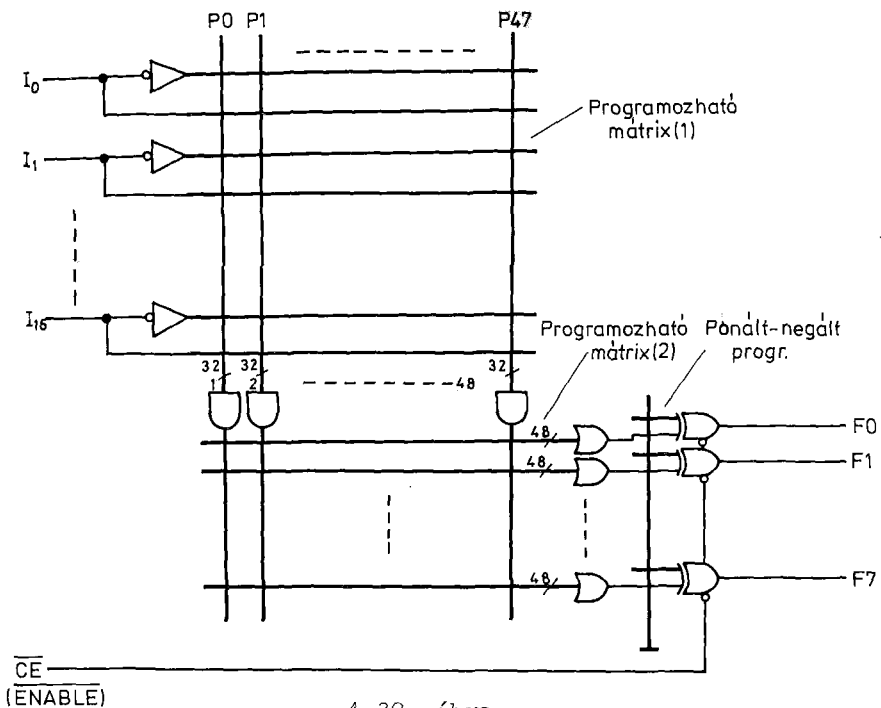
4.37. ábra

Ezekben 9 db 16 bemenetű ÉS, ill. NEMÉS, AND-NAND kapu van a 4.37. ábra katalógusból vett rajza szerint. A bejövő 16 változó ($I_0 \dots I_{15}$) negáltját is előállítja egy-egy inverter, így bármelyik kapu számára a ponált és a negált is rendelkezésre áll. A ponált-negált változók vezetékeit keresztező függőleges vezetékek jelképezik a NAND kapuk bemeneteit: "alap állapotban" mindegyik NAND kapu mindegyik változóval és ponáltjával összeköttetésben van. Programozáskor kell a "mátrix pontokban" a "felesleges" összeköttetéseket kiegészíteni, megszüntetni. A NAND kapuk kimeneteit programozhatóan negálhatjuk is (így AND kapuvá válnak): erre a célra szolgálnak a kimeneti vezetékekbe beiktatott kizáró-VAGY kapuk, amelyek mint vezérelhető inverterek működnek. Ha "másik" (vezérlő) bemenetüket földpotenciálra hagyjuk (a földvezeték keresztezi a vezérlő vezetékeket), akkor ponált, ha programozáskor leégetjük bármelyik vezérlő bemenetet, akkor negált érték áll elő (a kizáró-VAGY kapu vezérelhető inverter funkcióját lásd a 2.2. fejezetben).

Bonyolultabb, "többet tudó" hálózat az FPLA (Field Programmable Logic Array, innen kapta elnevezését az egész áramkör-család). Ezekben kétszintű kapu-hálózat van: ÉS kapukat VAGY/NEMVAGY kapuk követnek, amivel a logikai függvényeket "szokásos" ÉS-VAGY (ÉS-NEMVAGY, AOI) alakjukban valósíthatjuk meg

82S100 (three-state), ill. a
82S101 (open coll.)

áramkör szerkezetét a 4.38. ábra katalógusból vett rajza mutatja. A bemenetek száma itt is 16 ($I_0 \dots I_{15}$) a negáltakat belső inverterek segítségével itt is előállítják. A változó ponált-negált vezetékeit viszont most 48 (!) db ÉS kapu bemeneti vezetékei keresztezik. Az ÉS kapuk kimenetei egy újabb kereszteződő, "mátrix" hálózatra mennek, itt találkoznak az őket követő 8 db VAGY kapu bemeneteivel. A VAGY kapuk kimeneti jelei adják a teljes hálózat 8 kimenetének jelét az EOR kapuk vezérlésétől függően ponáltan (ÉS-VAGY függvény) vagy negáltan (AOI függvény). A programozás itt is a mátrix pontok "főlősleges" összeköttetéseinek megszakításával történik az égetés során. Végül is egy IC-vel 8 db 16 változós ÉS-VAGY függvényt



4.38. ábra

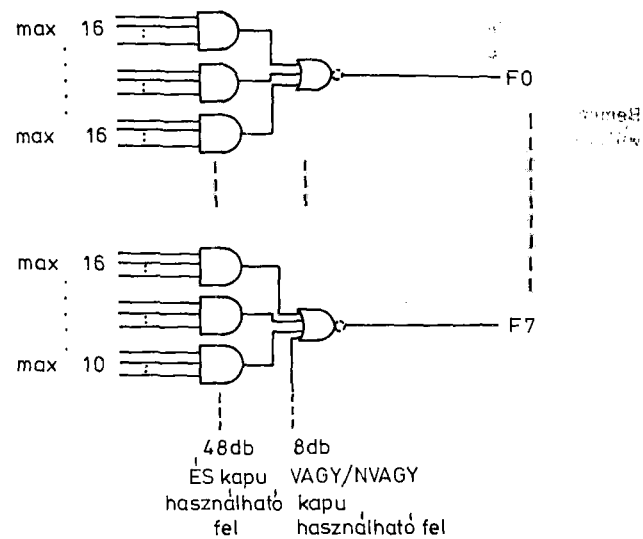
realizálhatunk azzal a korlátozással, hogy összesen 48 db ÉS kaput használhatunk fel (4.39. ábra), vagyis ennek a PLA-nak az alkalmazásakor a realizálandó függvényeket a gazdaságos kihasználás érdekében minimálni kell! Fontos adat, hogy az említett típusok jelterjedési késleltetési ideje 35 ns, max 80 ns, ami ilyen bonyolult hálózatra nagyon kedvező érték!

Az FPLS (Field Programable Logic Sequencer = programozható logikai sorrendi hálózat) a leginkább összetett áramkör, a kombinációs hálózat-részen kívül tárolókat is tartalmaz. A

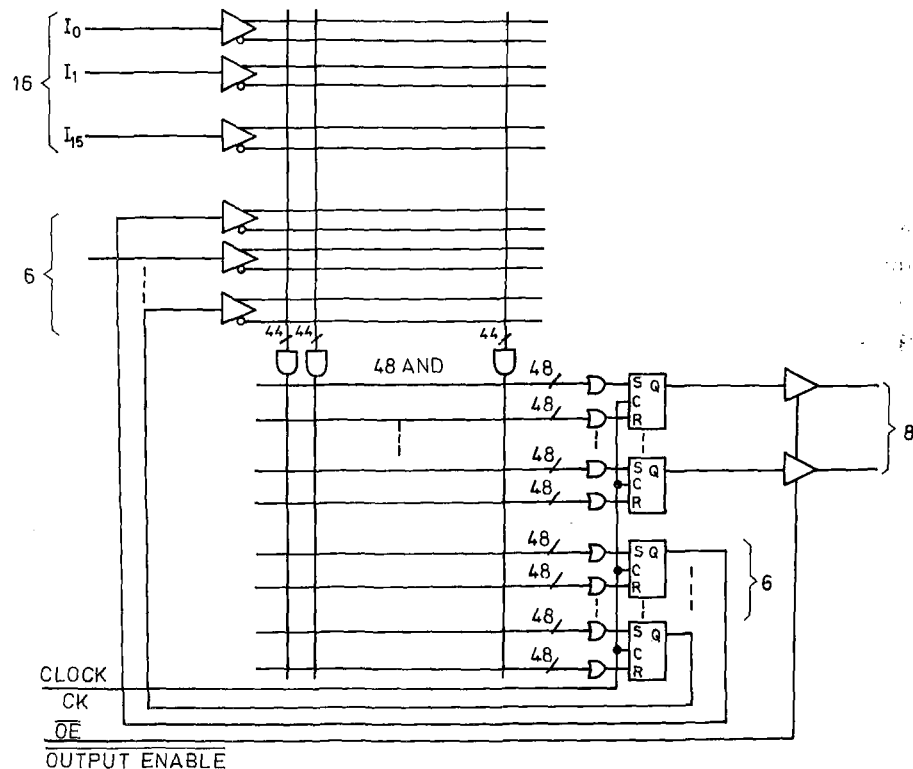
82S104 (open collector-os), ill. a

82S105 (three-state)

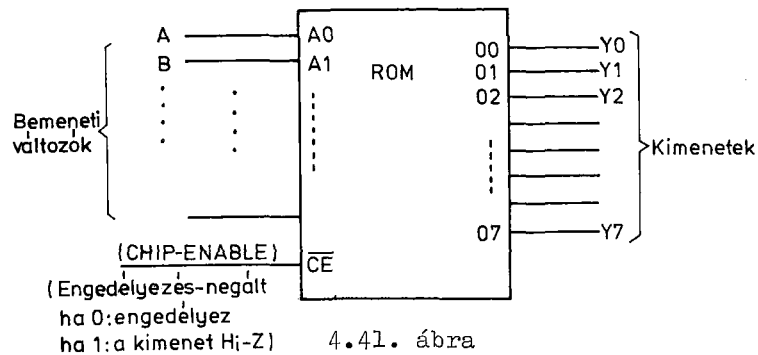
tipus például a "szokásos" 16 bemenetű, 48 ÉS kaput, 2x8 VAGY kaput tartalmazó áramkörön kívül a VAGY kapuk és a kimenet között 8 tároló áramkört, ezenkívül a bemenetre visszacsatolható, még 2x6 VAGY kapuval vezérelt tárolót tartalmaz (4.40. ábra). Így a bemenetek száma a visszacsatolt bemenetekkel együtt $16+6 = 22$. A beépített tárolók sokféle sorrendi funkció LSI-vel való megvalósítására teszik alkalmassá ezt az áramkört (lásd a későbbiek!).



4.39. ábra



4.40. ábra



- Kombinációs áramkörök megvalósítása ROM-mal,
összehasonlítás a PLA és a ROM között

A dekódolókkal kapcsolatban már szó volt az állandó tartalmu memóriák és a kombinációs hálózatok rokonságáról. Több bemenetű, több kimenetű hálózatok gazdaságosan megvalósíthatók ROM-mal, a bemeneti változókat a ROM cím-bemeneteire adjuk, aminek hatására a kimeneteken megjelenik az adott címen tárolt szó (4.41. ábra). Azt, hogy adott bemeneti kombinációkra milyen kimeneti jelek álljanak elő, a ROM programozásakor határozzuk meg. A ROM-oknak is van gyárilag programozott: MASK-PROGRAMMED változata és van "házilag" programozható PROGRAMMABLE ROM: PROM kivitele, ill. "újra programozható", törölhető-programozható EPROM változata. Kisebb sorozathoz az EPROM a leginkább kifizetődő. A ma legtöbbször használt típusok 8 kimenetűek ((byte szervezésűek) és annyi bemenetűek, ami az adott "memória kapacitás" címezéséhez szükséges (pl. MOS, 2kbyte-os EPROM, a 2716, ennek 2048 bemeneti kombinációjának kell lennie, amit 11 cím bemenettel lehet megvalósítani, a 4kbyte-os 2732-nek 12 bemenete van, és így tovább - minderről részletesen szó lesz a mikroszámítógép elemekkel kapcsolatban a következő kötetben). Az EPROM-ok ezidő szerint is könnyen beszerezhetők, nem drágák, ezért ahol csak lehetséges, ajánlott a felhasználásuk ("égető készülék" a legtöbb olyan helyen van, ahol digitális elektronikával foglalkoznak). A ROM-mal (EPROM-mal) történő logikai függvény megvalósításhoz nem kell egyszerűsíteni a függvényt, hanem inkább "bővítenünk", hogy egy, az összes bemeneti változó kombinációt tartalmazó

"igazságtáblázatot" rajzolhassunk - ez lesz a ROM "beégetéséhez" szükséges programtáblázat, nagyjából a következő formában (pl. 2 byte kapacitású ROM-ra):

BEMENET (bináris)										KIMENET (bináris)								
K	J	I	H	G	F	E	D	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	0 ⁰	0 ¹	0 ²	0 ³	0 ⁴	0 ⁵	0 ⁶	0 ⁷
0.	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0
1.	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0	0
.
2047.	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	1	0

BEMENET (hexadecimális)			KIMENET (hexadecimális)		
0.		0 0 0			5 A
1.		0 0 1			B C
.		.			.
.		.			.
.		.			.
2047.		7 F F			4 A

Az ilyen és ehhez hasonló ROM (EPROM) program táblázatokban "rejtve marad" a "logikai tartalom", egyszerűen csak a bemeneti változó-kombinációkhoz tartozó kimeneti állapotok összefoglalását tartalmazzák, annak ellenére, hogy tetszés szerinti logikai kapcsolatot valósítottunk meg. Ha a rendelkezésre álló bemenetek vagy kimenetek száma kevésnek bizonyul, akkor több tokot felhasználva bővíthetjük az áramköröket, felhasználva a 3-state, ill. open collectoros kimenetek párhuzamosíthatóságát, ill. azt, hogy mindegyik IC-nek \overline{CE} (Chip Enable: IC engedélyező) bemenete is van (a bővítésekről a memóriákkal kapcsolatosan később lesz szó, de remélhetőleg a demultiplexereknél tanult elvek felhasználásával ez most sem okoz problémát). A MOS ROM-ok, EPROM-mok "hozzáférési ideje" a címzéstől

a kimeneti "adat" jel előállításáig eltelt idő: 200...500 ns. Ha ennél nagyobb sebességre van szükség, akkor bipoláris ROM-ot PROM-ot kell felhasználnunk (EPROM ilyen fajtából általában nem kapható), ezek hozzáférési ideje 30...50 ns, viszont nagyobb a fogyasztásuk és kisebb a kapacitásuk (kevesebb a változók száma).

Végül hasonlítsuk össze az PLA-t és ROM-ot, mint a logikai kapcsolatok megvalósításának LSI eszközeit (tekintve, hogy látszatra a funkciójuk teljesen egyforma, esetleg érthetetlen, hogy egyáltalán miért gyártanak külön PLA-t és ROM-ot is):

- A PLA és a ROM valóban "rokon" hálózatok: sok bemeneti változót tudnak feldolgozni, több kimenetük is van, vagyis nagyszámu kaput tudnak helyettesíteni, és ami a legfontosabb: programozhatók (maszk-kal vagy helyszínen).

- A PLA típusoknak rendszerint több bemenetük van (pl. 16) mint a ROM-oknak (2 byte-hoz 11, 4 byte-hoz 12 bemenet tartozik, stb. az ennél nagyobb kapacitású ROM-ok már drágábbak, a nagy kapacitás miatt nehezebb a programozásuk is).

- A PLA-nak több bemenete van ugyan, de mivel korlátozott a benne lévő kapuk száma, az előállított logikai függvényekben csak korlátozott számú logikai szorzat, minterm lehet! Az említett 82S100-as sorozatban például 48 db ÉS kapu van, ami annyit jelent, hogy az előállítható 8 ÉS-VAGY függvényben összesen, együttvéve "csak" 48 ÉS kapcsolat, minterm lehet (egy-egy ÉS kimenetet több VAGY kapu bemenetén is felhasználhatunk, de a 48 szorzat-számot nem haladhatjuk túl - lásd a 4.38. ábrát). A ROM esetében ilyen korlátozás nincs: kevesebb ugyan a bemenet szám, de ennek összes kombinációját felhasználhatjuk függvény képzésre. Összesen tehát 2^n számú mintermet "építhetünk be" a függvényekbe (n a bemenetek száma, pl. 10 bemenet esetén 1024, 11 bemenetnél 2048...stb. az előállítható mintermek száma, ez sokkal több, mint 48).

A PLA tehát olyan ROM-nak fogható fel, amelynek "igen sok" cím bemenete van, de nem helyezhetünk el minden címre tartalmat, nem rendelhetünk minden lehetséges bemeneti jelkombinációhoz kimeneti "válasz" jelet.

- Az előző korlátozás bizonyos esetekben előny is lehet: amikor nincs igényünk nagyon bonyolult függvények előállítására (egy-egy függvényhez elegendő átlagban 6 minterm), akkor a PLA gazdaságosabb, mert "kisebb" lévén kevesebb teljesítményt fogyaszt, egyszerűbb, gyorsabb a programozása.

- A PLA általában sokkal gyorsabb a ROM-nál: a PLA-t ui. kisebb terjedelmének, fogyasztásának köszönhetően, bipoláris (Schottky) technológiával is lehet integrálni, a ROM-ok java része viszont MOS technológiával készül (ha bipoláris, akkor csak 6...8 bemenete lehet), a MOS áramkörök készletetése viszont nagyobb (bipoláris PLA kb. 35...80 ns, ROM: kb. 200...500 ns).

- Nem mellékes kérdés az ár, ill. a beszerezhetőség sem. A viszonyok természetesen napról napra változnak, de egy biztos: EPROM-ot már most nem túl drágán lehet vásárolni, nincs megrendelési, "átfutási idő", programozása a meglévő "hagyományos" készülékeken elvégezhető. A PLA alkalmazása nagy körültekintést igényel, először el kell dönteni, hogy FPLA-t, vagy maszk-programozott PLA-t igénylünk-e. Ha FPLA-t, akkor minden egyes IC-t programozni kell, ehhez a ROM-étől eltérően külön programozó készülékre van szükség. PLA alkalmazása esetében a megfelelő félvezető gyárral fel kell venni a kapcsolatot, a tervezést a gyárral közösen, egyeztetve kell elvégezni, a gyártmányokat ellenőrizni kell stb. A fejlődés előrehaladtával ez utóbbi ut már egyre könnyebben járható és az előállított LSI áramkörök is egyre olcsóbbak lesznek - legalábbis ezt ígérik a gyártók (köztük a magyar félvezető gyártók) is.

A PLA-val, ROM-mal felépülő logikai áramkörök ellentétét képezik a kapukból, huzalozással felépülő un. RANDOM LOGIC-nak ("rendszeretlen" logikának), azt mondhatnánk, hogy ezek a "rendszeres logikák" (szokásos elnevezés az "ARRAY LOGIC").

Befejezésül felsorolunk néhány olyan feladatot (készüléket, berendezést), amelyben az igen nagyszámu logikai áramkör igény miatt mindenképpen gazdaságos az LSI alkalmazása (akár PLA, akár ROM formájában):

- nyomtató vezérlő,
- folyamatszabályozásban használható többsatornás digitális szabályozó,

- forgalomirányító ("zöld hullám") digitális elektroniká-
ja,
- elektronikus digitális programkapcsoló,
- elektronikus gépjármű-gyújtás szabályozó,
- kódváltók: kódolók, dekódolók, karakter generátorok,
- CRT controller: katódsugárcsőves megjelenítő vezérlő
áramköre,
- digitális mágnesszalagos, kazettás, cartridge-es leme-
zes adatrögzítők vezérlése,
- személyi számítógépek kombinációs áramköreinek integ-
rált áramkörben való egyesítése (pl. SINCLAIR ULA vagy pl. a
COMMODORE-ban éppen 82S100-as PLA van),
- adatfeldolgozó rendszer (Data Acquisition System) idő-
zítő vezérlője,
- IEC busz interface és vezérlő,
- hordozható adó-vevő (pl. CB) csatornaválasztó vezérlő-
je,
- programozható frekvenciaosztó digitális frekvencia szin-
tetizátorhoz,
- IEEE-488 busz illesztése beszéd-szintetizálóhoz,
- funkció dekódolás CAMAC modulokhoz,
- programozható hullámforma generátor,
- cím dekódolás mikroprocesszor-rendszerekben,
- interrupt priority control (megszakítás-prioritás ve-
zérlés) mikroprocesszor rendszerben,
- ROM-ot áthidaló PLA a ROM programozási hibáinak javi-
tására,
- kisfrekvenciás digitális frekvenciamérő "reciprok kép-
zője",
- logikai analizátor,
- szinkron és video generátor,
- TV-játék kiegészítő logika,
- Digitális "adat-figyelő" és felismerő elektronika,
- illesztés kalkulátor IC-hez,
- billentyűzethez csatlakozó kódoló (Keyboard Encoder),
- számkombinációs elektronikus zár,
- digitális multiméter (automatikus méréshatárváltással)
tizedespont és dimenzió kijelzés vezérlésére, meghajtására,

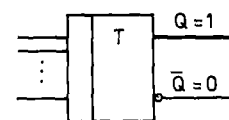
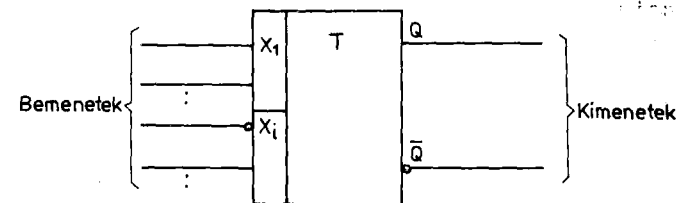
- összehasonlító digitális órához (előre programozott idő-
pontok "felismerésére"),
- tervszerű megelőző karbantartás idejét jelző monitor,
- időosztásos telemetriai rendszer vezérlője,
- érzékelők jelének digitális uton való linearizálása,
stb.

5.1. A flip-flop-ok modellje

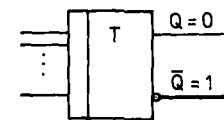
A logikai hálózatok működési elvét, modelljét a 2.1. fejezetből már megismerhettük. Szó volt arról, hogy a sorrendi hálózatok - a kombinációs hálózatoktól eltérően - a bemenetükre adott "primer változók" állapotain kívül figyelembe veszik a kimenetről visszacsatolt "szekunder" jeleket is, amelyek a hálózat előző állapotáról adnak információt. Ha a "szekunder változók" a bemenetre közvetlen visszacsatolással kerülnek, akkor a sorrendi hálózat aszinkron működésű, ha bemeneti tároló-soron keresztül, órajellel szinkronizálva csatlóznak vissza, akkor szinkron működésű az áramkör.

Ebben a fejezetben azokkal az aszinkron és szinkron alapáramkörökkel foglalkozunk, amelyek a sorrendi hálózatok leg egyszerűbb példáiként kezelhetők és egyben elemi alkotórészei is a bonyolultabb sorrendi áramköröknek, ezek a billenőkörök, flip-flopok.

A flip-flopok tömbvázlatát az 5.1. ábra mutatja. Két kimenetük van (ha nincs is mindig kivezetve): Q és \bar{Q} , a két kimeneti jel tehát mindig egymás negáltja. A bemeneti vezérlőjelek (amelyek lehetnek logikai szintek vagy szintváltozások, ugrások, impulzusok) a kimenet éppen felvett állapotától függetlenül fejtik ki hatásukat és amennyiben állapotváltozást hoznak létre, akkor ez hirtelen, billenés-szerűen következik be (innen a szokásos "billenőkör" elnevezés), a Q és \bar{Q} ugrásszerűen "cserél állapotot". A hirtelen állapotváltozást, a billenést a flip-flopokban mindig jelenlévő aktív elemek hozzák létre, amelyek teljesítményerősítése 1-nél nagyobb, az aktív elemek beépítése a flip-flopok létrehozásának alapvető feltétele.



1-es állapot



0-ás állapot

5.1. ábra

A flip-flopok kétféle lehetséges állapotát megegyezés szerint 0-val vagy 1-gyel jelöljük, aszerint, hogy a Q kimenet 0 (L) vagy 1 (H) szintű:

a flip-flop 1-ben van, ha $Q = 1$ és $\bar{Q} = 0$

a flip-flop 0-ban van, ha $Q = 0$ és $\bar{Q} = 1$

5.2. Bistabil flip-flopok

Aszerint, hogy milyen bemenetei vannak és, hogy ezek milyen igazságtáblázat szerint vezérlik a flip-flopot, a következő típusokkal foglalkozunk:

RS

JK

T

D

A flip-flopok aszerint, hogy a billenés milyen időzítéssel jön létre, lehetnek:

aszinkron és

szinkron működésűek (órajellel szinkronizált).

Az aszinkron működésű tárolók állapotváltozása a bemenetekre adott vezérlőjel hatására közvetlenül jön létre a késleltetési idő elteltével.

Az órajellel vezérelt flip-flopok állapotváltozása csak akkor jöhet létre, ha a szinkronizáló, ÓRA (CLOCK) bemenetükre megérkezik a szinkronizáló órajel, óraimpulzus.

A flip-flopok vezérlése lehet:

- statikus és
- dinamikus.

A statikus vezérlő bemenetekre az igazságtáblázat szerint logikai 0 vagy logikai 1 egyenszinteket kell adni, ezek határozzák meg a flip-flop új állapotát.

Dinamikus a vezérlés akkor, ha a flip-flop billenése a dinamikus vezérlő bemenetre adott jel meghatározott irányu változásának hatására jön létre (1-0 vagy 0-1 átmenetre, tippustól függően).

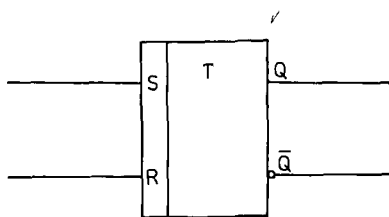
5.2.1. Az RS flip-flopok

A minden jelző, kiegészítő elnevezés nélküli RS megjelölés mindig statikus vezérlésű (aszinkron) áramkört jelöl, amely a bemenetekre adott logikai (egyen-)szintekkel vezérelhető.

A statikus, aszinkron RS flip-flopoknak két vezérlő bemenete van:

S SET: beíró bemenet,

R RESET: törlő, nullázó bemenet (5.2. ábra).



5.2. ábra

A statikus RS flip-flop SET bemenetére 1-et adva a tároló 1-be billen és a vezérlés megszűnése után is 1-ben marad, RESET bemenetére 1-et adva 0-ba billen és 0-ban marad. Ha mind-

két bemenet 0 szinten van, akkor a flip-flop az előző állapotát tárolja, billenés nem történik. Mindkét bemenetre egyszerre nem adható 1-es vezérlés, ez tiltott kombináció. A kimenetek állapotára ebben az esetben nincs előírás (ilyenkor az is lehet, hogy az egyik kimenet jele a másiknak nem lesz negáltja).

A működést egy igazságtáblázatban foglalhatjuk össze, amelyben bemeneti változónak az R, ill. S jelet tekintjük, kimenő jelnek a Q billenés utáni értékét, melyet Q_{n+1} -el jelölünk. A vezérlés általánosságban az n-edik ütemben megy végbe, az új állapot az n+1-edik ütemben jön létre:

R	S	Q_{n+1}	
0	0	Q_n	az eredeti Q_n állapot változatlan marad (tárolás)
0	1	1	a flip-flop 1-be billen
1	0	0	a flip-flop 0-ba billen
1	1	-	tiltott bemeneti kombináció, a kimenetre nincs előírás.

Az igazságtáblázatban szerepel Q_n is, amely azt mutatja, hogy az RS flip-flop n+1-edik állapota az n-edik állapotnak is függvénye. Ezt jobban olyan teljes igazságtáblázattal szemléltethetjük, amelyben a bemeneti változók között Q_n lehetséges értékei is szerepelnek:

R	S	Q_n	Q_{n+1}	
0	0	0	0	az eredeti állapot változatlan marad (tárolás)
0	0	1	1	
0	1	0	1	0-ból 1-be billen
0	1	1	1	1-ben marad
1	0	0	0	0-ban marad
1	0	1	0	0-ba billen
1	1	0	-	tiltott bemeneti kombináció
1	1	1	-	

Ahhoz, hogy az RS flip-flop áramkörét megvalósítsuk, fel kell írni Q_{n+1} egyszerűsített logikai függvényét. Az egyszerűsítéshez célszerű KARNAUGH-táblát rajzolni, amelynek változói R, S, Q_n és a kimeneti függvény: Q_{n+1} . A megvalósításhoz, szokás szerint, inverteres, univerzális kapukat használunk, amelyeknek biztos, hogy 1-nél nagyobb erősítésük van, mert ez feltétel a sorrendi áramkörök esetében. A "hagyományos" RS flip-flopok NOR kapukból épültek fel - ilyenek voltak a két tranzisztoros változatok, a tranzisztoros bistabil multivibrátorok - ezért először mi is a NOR megoldást keressük.

Az RS flip-flop minterm táblája az igazságtáblázat alapján (5.3. ábra):

RS	Q_n	0	1
00			1
01		1	1
11		X	X
10			

Minterm tábla Q_{n+1} -re

5.3. ábra

$$Q_{n+1} = S + \bar{R}Q_n$$

A tiltott bemeneti kombinációk mintermjeit a táblában X-szel jelöltük és don't care mintermekként kezeljük, hiszen ilyen vezérlés nem fordulhat elő. Az egyszerűsítést elvégezve kapjuk a Q_{n+1} függvény egyik formáját, melyet karakterisztikus egyenletnek szoktak nevezni, mert a flip-flop működését jellemzi. A

$$Q_{n+1} = S + \bar{R}Q_n$$

egyenlet azt mondja ki, hogy a kimenet új állapota akkor lesz 1-es, ha vagy a SET bemenetre 1-et adunk, vagy a RESET-re 0-t adunk, de a flip-flop eredetileg 1-ben volt ($Q_n = 1$ volt).

A minterm tábla alapján a széleken és a tábla belsejében végrehajtott 0-1 cserével megszerkeszthető a MAXTERM tábla (a don't care eseteket ismét X-szel jelölve, 5.4. ábra):

RS	Q_n	0	1
11		1	
10			
00		X	X
01		1	1

MAXTERM tábla

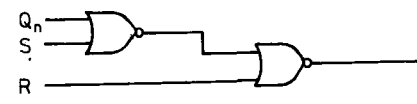
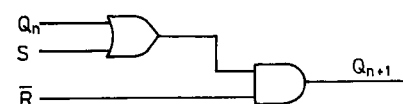
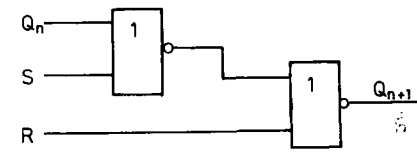
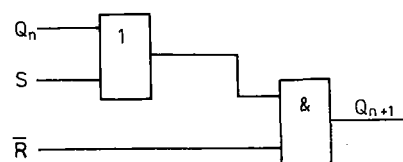
Q_{n+1} -re

5.4. ábra

$$Q_{n+1} = \bar{R}(S + Q_n)$$

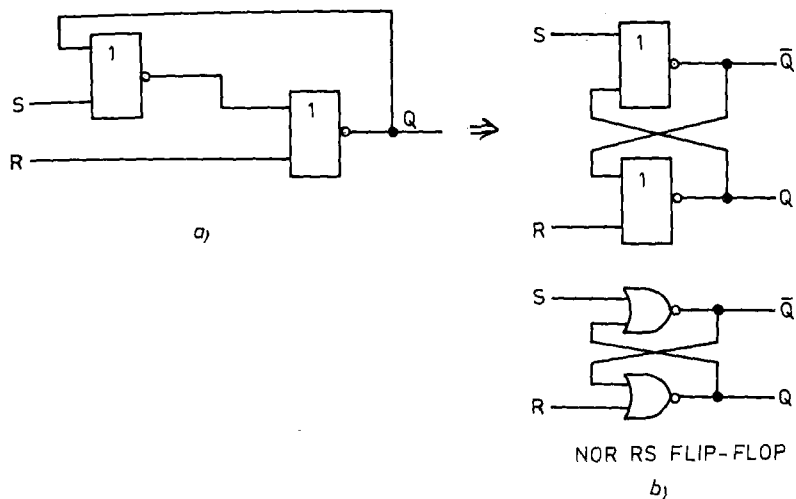
A karakterisztikus egyenlet ezen formája azt fejezi ki, hogy a flip-flop akkor lesz 1-ben, ha a RESET nem 1-es és vagy a SET 1-es, vagy a flip-flop már eredetileg is 1-ben volt. A MAXTERM-es formából megrajzolható a NOR kapuból felépülő RS flip-flop változat; a VAGY-ÉS hálózat kapuit átrajzoljuk NOR-ra és a közvetlenül a kimeneti kapura menő (\bar{R}) változót ne-gáljuk (5.5. ábra):

$$Q_{n+1} = \bar{R}(S + Q_n)$$



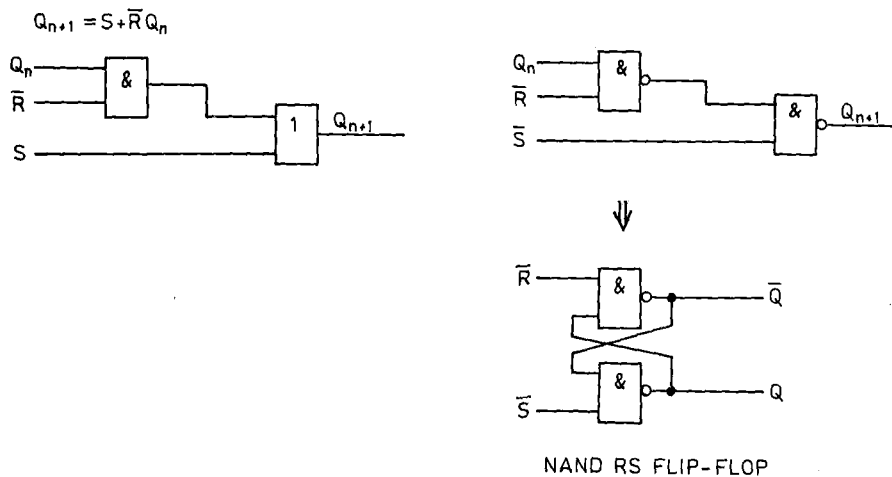
5.5. ábra

A Q_n és Q_{n+1} a hálózatban természetesen ugyanaz a pont kell legyen, hiszen Q_n és Q_{n+1} ugyanaz a függvény (csak éppen különböző időpontban, 5.6. ábra).



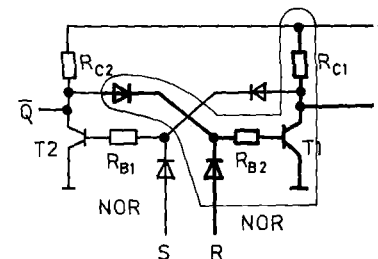
A végeredmény, az RS flip-flop szokásos, NOR kapukból felépített szimmetrikus változata az 5.6b ábrán látható.

A NAND változat felrajzolásához a minterm-es alakot használjuk fel (5.7. ábra):

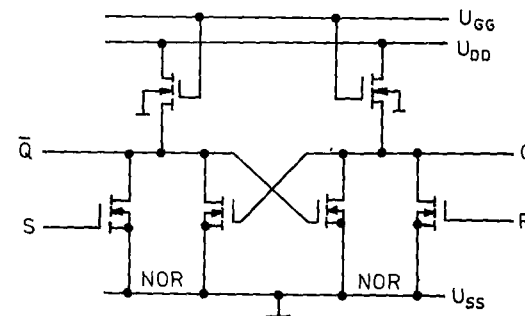


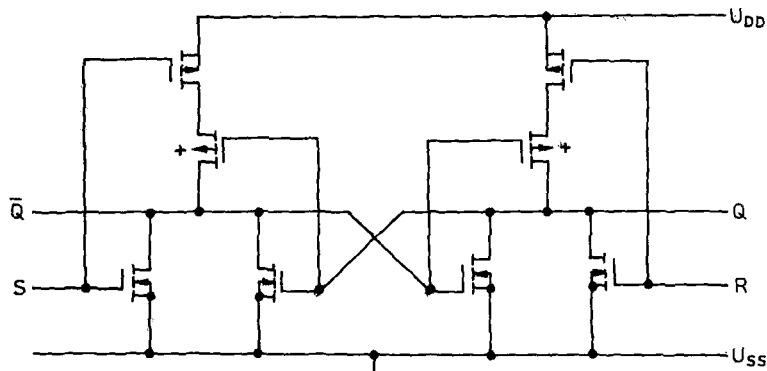
A bemenetekre a SET és RESET negáltját kell adnunk, ami annyit jelent, hogy a NAND RS flip-flop a megfelelő bemenetre adott logikai 0 hatására billen. Az áramhuzó rendszerekben (DTL, TTL, I²L), ahol a bemenetek "nyugalmi helyzete" logikai 1-es, a logikai 0-val (földeléssel) vezérelhető NAND változat acélszerűbb, ezért rendszerint ilyen RS flip-flopokat állítunk össze, ill. ilyen integrált változatokat használunk fel.

Említettük már, hogy a régebbi, tranzisztoros bistabil áramkörök is tulajdonképpen két, tranzisztoros NOR kapuból épültek fel (5.8. ábra). Egy-egy NOR kapu egyik bemenete (egyik VAGY-dióda anód) az R, ill. S vezérlő bemenetekhez, másik bemenete (másik VAGY-dióda anód) a másik tranzisztor kollektorához van kötve.

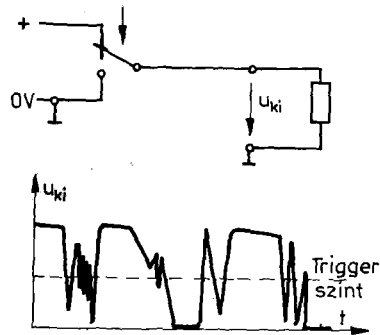


A MOS áramkörök statikus bistabil RS tárolói is NOR kapukból épülnek fel az 5.6. ábra elvi rajza szerint (5.9. ábra) és ugyanúgy a CMOS RS flip-flopok is (5.10. ábra).





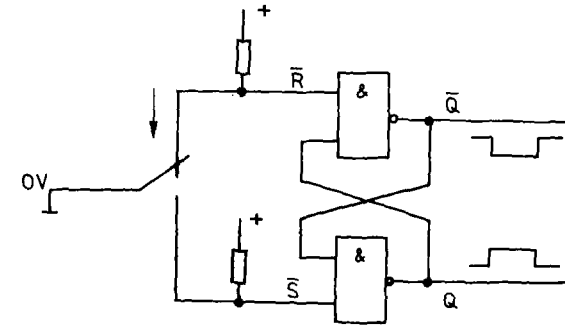
5.10. ábra



5.11. ábra

Az RS flip-flopok felhasználása sokrétű. Digitális jeltároláshoz, berendezésekben vezérlési feladatok ellátásához stb. sokszor építünk be RS flip-flopokat. Egy nagyon gyakori példa mechanikai érintkezők pergésmentesítése RS tárolóval. Kézi működtetésű és jelfogó érintkezők digitális berendezésekhez való csatlakoztatásakor sokszor problémát okoz a pergés, vagyis az a jelenség, hogy az érintkező átkapcsolásakor nem egyetlen szint-átmenet keletkezik, hanem egy egész impulzus sorozat (5.11. ábra), mert az "elengedés" és "hozzáérés" pillanatában a villamos kapcsolat bizonytalan. A jelet fogadó egység ezt annyi impulzusnak érzékeli, ahányszor átlépi a logikai 0 és logikai 1 szintet elválasztó triggerelési szintet és

a többszörös jel téves működéshez vezethet. (A "leggyorsabb"-mechanikai kapcsolók is peregnék, pl. MILTAC stb., kizárólag egyes higanyos, ill. higany-nedvesítésű fajták képesek pergésmentes jelet adni!). Amennyiben MORZE kapcsoló áll rendelkezésre, a pergést egy NAND flip-floppal küszöbölhetük ki, az 5.12. ábra szerint.

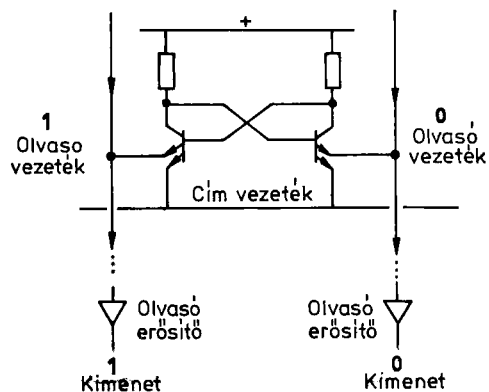


5.12. ábra

A kapcsoló nyugalmi (felső) helyzetében - mivel a felső kapu egyik bemenete 0 V-on van - a flip-flop 0-ás állapotban van, $Q = 0$. Amikor az átkapcsolás (a nyíl irányában) megindul, a felső érintkezőn pergés jön létre, de logikai 0-1 között ingadozó feszültség nem változtatja a flip-flop állapotán. Amikor a mozgó érintkező első ízben az alsó ponthoz ér, a flip-flop átbillen, $Q = 1$ lesz. Az itt létrejövő pergés, az ismételt logikai 0 vezérlés az \bar{S} bemeneten, nem változtatja a $Q = 1$ állapotban. Amikor a kapcsoló nyugalmi helyzetbe visszatér és először hozzáér a felső érintkezőhöz, a flip-flop visszabillen, $Q = 0$ lesz és ezen már a pergés nem változtatja. A Q kimenetről tehát zavarmentes "egyszeres" jelet vehetünk le. A kapcsolóval szemben támasztott egyetlen fontos követelmény, hogy átkapcsolás közben egy pillanatra se érjen össze mind a három pontja (egyes tolókapcsoló típusokra ez nem teljesül). Érdekes megjegyezni, hogy kézi adatbevitelre, vezérlésre drágább berendezésekben inkább mechanikai érintkezőt nem tartalmazó (Hall generátoros, kapacitív, stb.) kapcsolókat használnak fel.

LSI áramkörök RS tároló-elemei

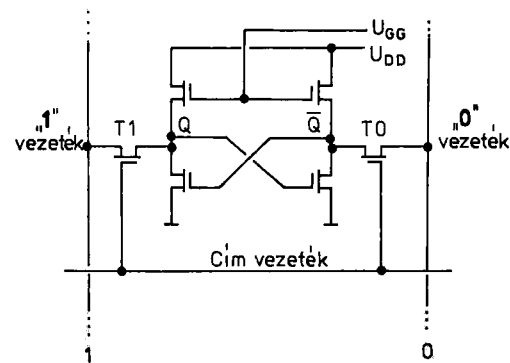
Külön kategóriát képviselnek azok a bistabil tároló változatok, amelyek félvezető, sok-bites memóriák (R/W Read-Write, kiolvasható-beírható memóriák, más néven RAM, Random Access Memory, "véletlen" hozzáférésű memóriák) elemi tároló celláit alkotják. Ezek közül megemlítünk néhány jellemző típust.



5.13. ábra

Bipoláris (TTL) RAM-ok jellemző tároló eleme látható az 5.13. ábrán. Ilyenből van több tíz, esetleg több száz egy áramkörben. Felépítése hasonló a tranzisztoros RS tárolóhoz, de a vezérlést az emitterek kapják. Egyszerre itt is csak az egyik tranzisztor vezethet, mert a vezető tranzisztor 0 V körüli kollektor feszültsége lezárva tartja a másikat. A cím vezeték (vezetékek) alaphelyzetben földpotenciálban vannak és a flip-flop tárolja a beírt bit értéket. Amikor a tároló tartalmát ki akarjuk olvasni, "megcimezzük" ezt a cellát oly módon, hogy a cím vezetékét pozitív feszültségre kapcsoljuk. Az éppen vezető tranzisztor emitterárama ilyenkor az olvasó vezetékbe folyik (az olvasó vezeték mindegyik tárolóhoz közös). A tároló állapotától függően vagy az "1-es" olvasó vezetéken folyik áram, vagy a "0-ás" olvasó vezetéken, amelyet a megfelelő olvasó erősítő érzékel és vagy az 1-es, vagy a 0-ás memória kimeneten lesz logikai 1 feszültség. A beírás szintén a

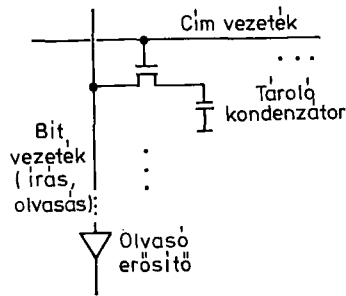
kiválasztott cella cím vezetékének pozitív feszültségre vite-lével történik, miközben vagy az 1-es (ha 1-et akarunk beírni és tárolni), vagy a 0-ás olvasó vezetékét 0 V-ra visszük, amivel a megfelelő tranzisztor vezetővé válik, majd a cím vezeték alaphelyzetbe, 0-ba vitele után a cella tárolja a beírt értéket.



5.14. ábra

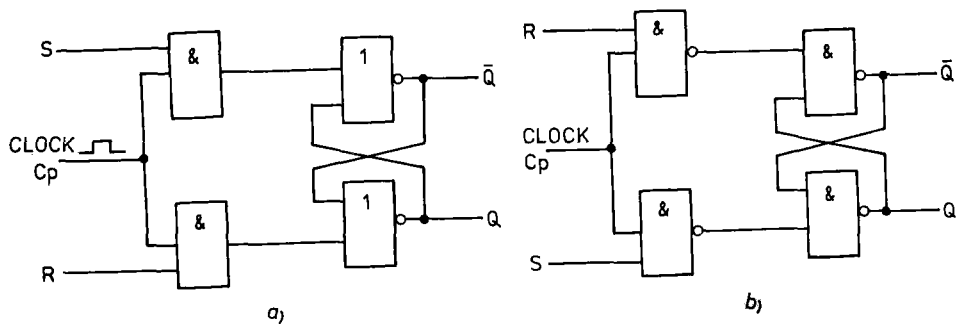
A MOS tárolók a legnagyobb változatosságot mutatják, így csak egy-egy példát mutatunk be. Egy ún. statikus RAM cellát mutat az 5.14. ábra. Alaphelyzetben az RS flip-flop tárolja az előzőleg beírt értéket. Ha valamelyik cella tartalmát ki akarjuk olvasni, akkor az illető cella (cellák) cím vezetékére logikai 1 szintet adunk, így a T1 és T0 vezetővé válik és a flip-flop Q, ill. \bar{Q} kimenetét összeköti a közös "1", ill. "0" bit vezetékkel, így a kimeneten megjelenik a kiválasztott cella tartalma. Beírásakor ugyanúgy a kiválasztott tároló (tárolók) cím vezetékére adott logikai 1-gyel vezetővé tesszük T1-et és T0-át és a közös "1", ill. "0" vezetékre adott jellel a kívánt állapotba billentjük ezt a kiválasztott tárolót.

A dinamikus MOS tárolókra az a jellemző, hogy a tárolást vagy a GATE kapacitással, vagy külön erre a célra létrehozott kapacitással valósítják meg. Emiatt - mivel a kiskapacitású tároló "kondenzátor" a maradékáramok miatt gyorsan veszít töltéséből - a tárolt információt időnként, "frissíteni" kell, azaz kiolvasni, majd visszairni a tárolóba. A legegyszerűbb dinamikus tároló cellát, amely egyetlen tranzisztort tartalmaz,



5.15. ábra

az 5.15. ábra mutatja. A tároló kondenzátort az LSI áramkörben MOS szerkezettel vagy a szilícium GATE "anyagából készült" fegyverzettel hozzák létre. Olvasáskor a megcimezett tranzisztor vezet és a tároló kondenzátor jelét a közös vezetékre kapcsolja. Beírásakor, ill. frissítéskor a megcimezett, vezetővé vált tranzisztor a közös vezetékre adott jelfeszültségre tölti fel a tároló kondenzátort, majd a cím-jel megszűnése után megszakad és a kondenzátor tárolja a jelet.



5.16. ábra

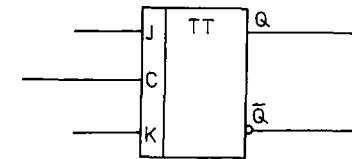
Az órajellel vezérelt (CLOCKED) RS flip-flop a legegyszerűbb szinkronizált tároló, és az összetettebb dinamikus vezérlésű flip-flopok alapáramköre. Annyiban különbözik a statikus RS változattól, hogy a bemeneti R és S jel egy-egy ÉS kapun halad át (5.16a ábra). Az ÉS kapuk másik bemenete az engedélyező ÓRA (CLOCK) jelet kapja. Ameddig az órajel logikai

0, a vezérlő bemenetekre adott jel hatástalan, a flip-flop állapota változatlan marad. Amikor az órajel "megérkezik", azaz logikai 1 lesz, a vezérlő jeleket az ÉS kapuk átengedik és a flip-flop felveheti a vezérlés szerinti állapotát. Az órajel ezután ismét visszatér 0-ra és tiltja a további állapotváltást. Az órajellel vezérelt RS flip-flop NAND változatát az 5.16b ábra mutatja. Működése hasonló az előzőhöz, amíg a CLOCK jel logikai 0, addig a bemeneti NAND kapuk kimenetén logikai 1 van, ami nincs hatással a flip-flop állapotára. Az órajel 1-re ugrásának pillanatában a flip-flop az R-S bemenetek által meghatározott állapotba billen. (A bemeneteken itt nem \bar{R} és \bar{S} van, mert a negálást a bemeneti NAND kapuk elvégzik.)

5.2.2. A JK és a T flip-flop

A JK flip-flop vezérlő bemenetei

J beíró bemenet,
K törlő bemenet.

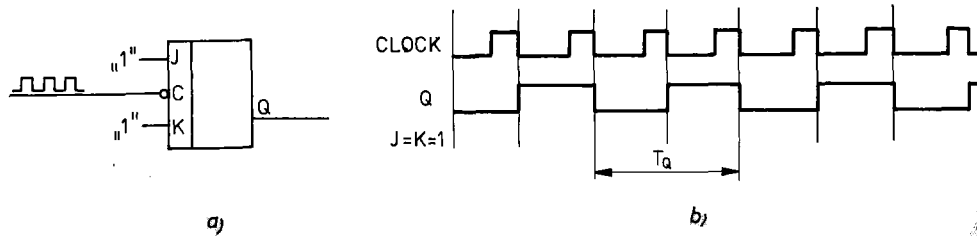


5.17. ábra

Általában órajellel szinkronizált JK flip-flopokat használnak, ezért mi is ezzel foglalkozunk (5.17. ábra). Igazságtáblázata hasonló az RS flip-flopéhoz, azzal a különbséggel, hogy a megengedett a $J = K = 1$ vezérlés is, vagyis a J és K bemenet egyidejűleg 1 is lehet. Ilyenkor a tároló ellenkező állapotba billen:

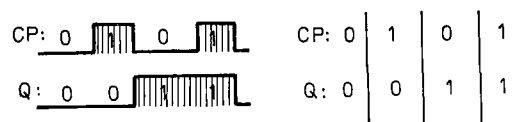
K	J	Q_{n+1}	
0	0	Q_n	az órajel megérkezése után sem billen,
0	1	1	az órajel megérkezésekor 1-be billen,
1	0	0	az órajel megérkezésekor 0-ba billen,
1	1	\bar{Q}_n	az órajel megérkezésekor <u>ellentétes állapotba</u> billen.

A JK flip-flop megvalósításakor ebből a $J = K = 1$ vezérlési esetből kell kiindulnunk, hiszen a többi állapotot az RS flip-flop is előállítja. Az említett vezérlési eset azt jelenti, hogy a flip-flopnak számláló, frekvenciaosztó típusúnak kell lennie, mert minden órajelre ellentétes állapotba kell billennie, ami annyit jelent, hogy egy teljes ciklus (amíg a kimenet ellentétesre vált, majd eredeti állapotba kerül) két órajel időtartama, két egymás utáni órajel hatására megy végbe (5.18. ábra).

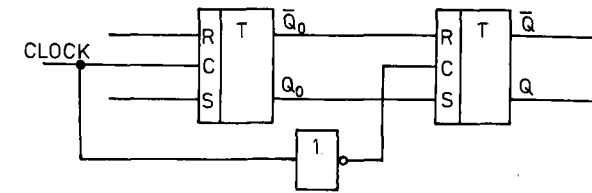


5.18. ábra

Egy teljes (T_Q) periódusban az órajel és a Q kimenet felvett értékeit figyelembe véve összesen 4-féle kombinációnak kell előállnia (5.19. ábra).



5.19. ábra



5.20. ábra

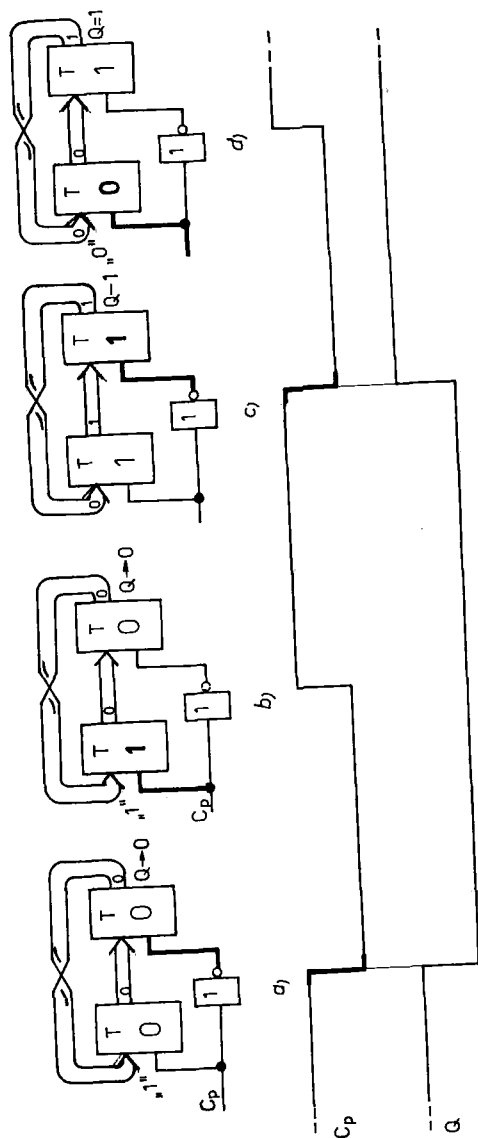
A 4-féle kombinációt legegyszerűbben 2 flip-floppal állíthatjuk elő, így kapjuk az un. MASTER-SLAVE ("ur-szolga") - flip-flopot. A név arra utal, hogy egyetlen tárolóban két flip-flop van, és a MASTER állapotát a SLAVE "szolgáian" követi (másik órajel fázisban). Az elvi felépítés az 5.20. ábra szerinti. A MASTER kimenetére csatlakozik a SLAVE bemenete. Mindkét flip-flop órajellel vezérelt változat (pl. órajellel vezérelt RS flip-flop az 5.16. ábra szerint), de a SLAVE a MASTER-hoz képest ellentett fázisú órajelet kap - ezt a be-rajzolt inverter jelképezi. Amikor az egyik flip-flop billen, a másik éppen tiltva van. Számlálás, vagyis minden órajelre ellentétes állapotba billenés úgy következhet be, hogy a MASTER bemenetére a SLAVE kimenetének negált jelét csatoljuk vissza, innen "tudja" a MASTER, hogy a következő órajelre ellentétes állapotba kell billennie. A folyamatot az 5.21. ábra szemlélteti. Kezdetben a MASTER és a SLAVE is 0-ban van. A SLAVE kimeneti jele (ez a teljes flip-flop kimeneti jele) "megfordítva" érkezik a MASTER bemenetére, ott logikai 1-es "várakozik" a MASTER még nem billenhet 1-be, mert a logikai 0 órajel tiltja (a ábra). Amikor az órajel 1-be ugrik (b ábra) a MASTER már 1-be billenhet, de kimenő jele nem billentheti a SLAVE-t, mert annak számára az órajel logikai 0 szintű (az inverter miatt). A SLAVE és így a Q kimenet tehát 0-ban marad.

Amikor az órajel 1-ből 0-ba ugrik (c ábra), akkor a SLAVE óra-bemenete 1-es lesz (az óra inverter miatt) és így a SLAVE 1-be billen (mivel a MASTER 1-t "készített elő" a bemenetére), $Q = 1$ lesz.

A "fordított" visszacsatolás a MASTER bemenetére logikai 0-át visz, előkészítve ezzel a 0-ba billenést. Amikor az órajel ismét 1-be ugrik (d ábra) a MASTER 0-ba billen, majd az

órajel 0-ra visszatérésekor a MASTER "0" tartalma átíródik a SLAVE-be, $Q = 0$ lesz, visszajutottunk az alaphelyzetbe (a ábra).

A Q kimeneten példánk szerint mindig akkor van állapotváltozás, amikor az órajel 1-ből 0-ba ugrik, az órajel minden 1-0 átmenetekor a flip-flop ellentétes állapotba billen, Q ellentétesre változik.



5.21. ábra

A működés és a tömbvázlat alapján felrajzolhatjuk a MASTER-SLAVE flip-flopnak azt az áramköri vázlatát, amely a $J = K = 1$ esetet valósítja meg, azaz minden órajelre ellentétes állapotba billen. A MASTER és a SLAVE is egy-egy órajellel vezérelt RS flip-flop, de a SLAVE invertált órajelet kap. A Q és \bar{Q} kimenetről (SLAVE kimenetekről) invertálva visszacsatoljuk a jelet a MASTER RS bemeneteire (Q az R-hez, \bar{Q} az S-hez, 5.22a ábra).

Ebből JK flip-flopot egyszerű kiegészítéssel kaphatunk: a bemeneti kapukat még egy-egy bemenettel látjuk el, az S oldalon az ujjabb bemenet lesz a J, az R oldali ujjabb bemenet a K (5.22b ábra).

Könnyen belátható, hogy az így kiegészített kapcsolás a JK flip-flop igazságtáblázata szerint működik:

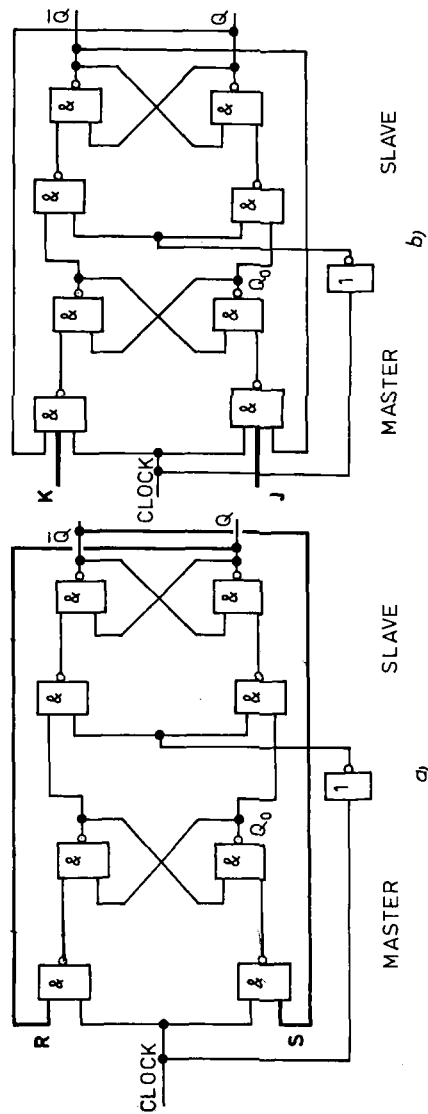
- a $J = K = 0$ vezérlés esetén mindkét bemeneti kapu tiltva van, ezért a MASTER az órajelre nem billenhet, s így a SLAVE sem változtathatja állapotát, a MASTER-SLAVE flip-flop tehát nem billen, $Q_{n+1} = Q_n$;

- a $J = 1$ és $K = 0$ vezérlésnél a K-hoz tartozó bemeneti kapu tiltva van és ha a flip-flop eredetileg 0-ás állapotban volt ($\bar{Q} = 1$) és az órajel 1-be megy, akkor a J-hez tartozó kapu mindegyik bemenetén 1-es lévén, kimenete 0 lesz, és a MASTER 1-be billen. Az órajel 0-ba ugrásakor a SLAVE is 1-be billen, $Q = 1$ lesz. További órajelek nem változtatnak a $Q = 1$ állapotban, ami a $J = 1$ hatására jött létre;

- a $K = 1$, $J = 0$ vezérlés a J-hez tartozó kaput tiltja, és a K-hoz tartozót aktiválja, a flip-flop csak 0-ba billenhet az órajel hatására;

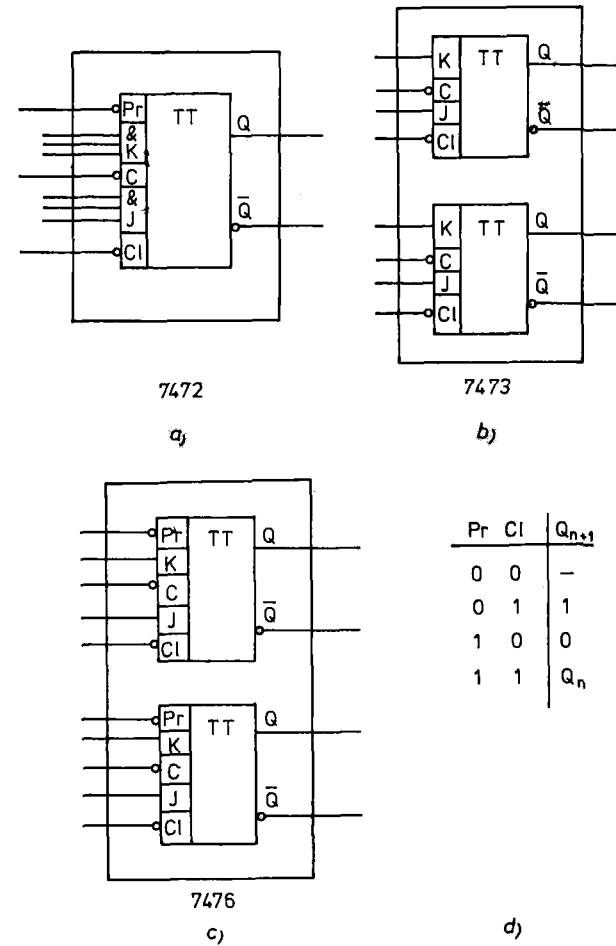
- a $J = K = 1$ vezérlés hatása ismert, a flip-flop minden órajelre ellentétes állapotba billen - ennek a vezérlési esetnek a megvalósításához választottuk a MASTER-SLAVE elrendezést.

A működés elve minden vezérlési esetben az, hogy a bemeneti jelek az első ütemben a MASTER-ba íródnak be, majd a második ütemben a MASTER tartalma átíródik a SLAVE-ba. A szabvány a "kettős tárolók" kapcsolási rajzjelébe "TT" feliratot ír elő.



5.22. ábra

Az univerzális célra gyártott integrált bipoláris JK MASTER-SLAVE flip-flopok felépítése az 5.22. ábrán látotthoz hasonló. Példaként megemlítünk néhány gyakrabban felhasznált TTL SSI típust és összefoglaljuk a működtetési előírásokat.

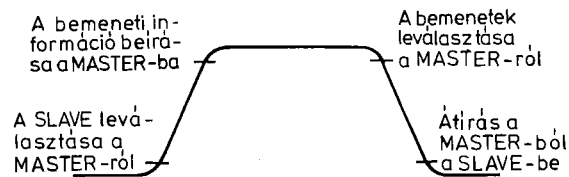


5.23. ábra

A 7472 típus egy tokban egy JK flip-flopot tartalmaz (5.23a ábra), de 3-3 J és K bemenete van. Ezek ÉS kapcsolatban vezérlik a flip-flopot, azaz mindhárom J, ill. K bemenetnek logikai 1-nek kell lenni ahhoz, hogy az illető bemenet aktív legyen.

A J, K és az órabemeneteken kívül még **Cl: CLEAR** (törlő) és **Pr: PRESET** (beíró) bemenettel is ellátták a flip-flopot. A PRESET és a CLEAR statikus bemenetek, a rájuk adott logikai 0 hatásos és órajel nem kell a billentéshez. A CLEAR-re adott logikai 0 a flip-flopot 0 állapotba billenti, a PRESET-re adott logikai 0 pedig 1-be, az igazságtáblázatot az 5.23d ábra mutatja. Ha a statikus vezérlő bemeneteket működtetjük (logikai 0-val), akkor a J, K és az óra bemenet hatástalan, a PRESET és a CLEAR bemeneteknek elsődleges, meghatározó szerepük van, a flip-flop 0-val vezérelt RS tárolóként működik (a Pr=Cl=0 vezérlés tiltott). A J, K és az óra bemenet akkor működik csak, ha mindkét statikus bemenet logikai 1-en van (Pr=Cl=1).

A 7473-as típusban egy tokban 2 darab JK MASTER-SLAVE flip-flop van egy-egy statikus nullázó (CLEAR) bemenettel. A 7476-os egy tokban szintén két JK flip-flopot tartalmaz PRESET és CLEAR bemenettel (5.23b és c ábra).

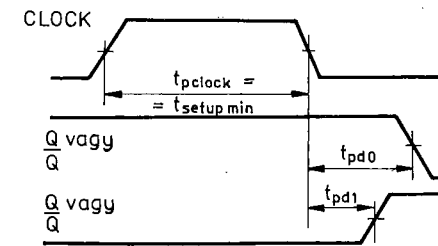


5.24. ábra

A katalógus a szokásos statikus adatokon és igazságtáblázatokon kívül további működtetési adatokat közöl. Megadja például, hogy az óra-impulzus különböző pillanatértékeinél mi történik a MASTER-SLAVE flip-flopban (5.24. ábra). Közli az idő adatokat, ill. előírásokat. Ezek közül a felhasználó szempontjából talán legfontosabb az INPUT SETUP TIME, bemeneti előkészítési idő. Az eddig tárgyalt MASTER-SLAVE flip-flopok, az órajel 1-0 átmenetekor billennek. A J és K bemeneten a vezérlő jeleket az órajel 1-0 átmenete előtt megfelelő idővel előbb elő kell készíteni ahhoz, hogy a flip-flop az igazságtáblázat szerint billenjen. Ez az idő az előkészítési idő. Az ismertetett TTL típusokra a katalógus a

$$t_{\text{setup}} \approx t_{\text{pclock}}$$

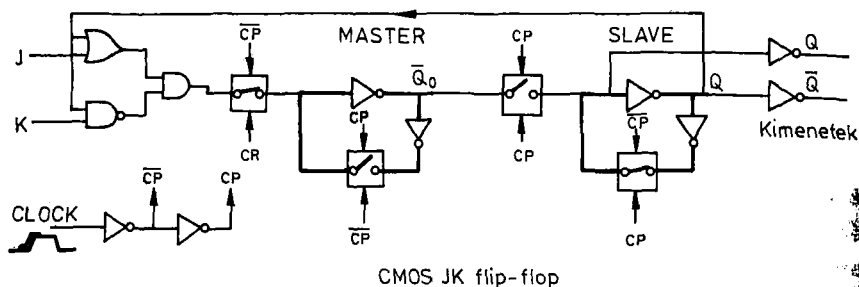
feltételt szabja meg, mely szerint az előkészítési idő nem lehet rövidebb, mint az órajel ideje, az az idő, amíg az órajel 1-ben van (l. az idődiagramot az 5.25. ábrán).



5.25. ábra

Ez azt jelenti, hogy a J és K bemenetre a vezérlő jelet legkésőbb az órajel 1-be ugrásának pillanatáig "oda kell készíteni" és már nem szabad megváltoztatni mialatt az órajel 1-ben van, mert ellenkező esetben előfordulhat, hogy nem az igazságtábla szerinti állapotot vesz fel a tároló. Az 5.22. ábrán látható kapcsolási rajzon jól követhető, hogy mi történik akkor, ha például a J és K bemenetre egyaránt logikai 0 vezérlést adunk, de egy pillanatra - azután, hogy az órajel logikai 1 lett - a J bemenetet logikai 1-re visszük, majd utána ismét logikai 0-ra. A J = K = 0 vezérlés miatt a flip-flop állapotának változatlanoknak kellene maradnia, így azonban, ha eredetileg 0-ban volt, 1-be billen az órajel 1-0 átmenetét követően. Ez azért következik be, mert amíg CP = 1 és J = 1, a MASTER a J bemenethez tartozó nyitott kapu jelére 1-be billen, és hiába visszük vissza a J bemenetet logikai 0-ra, ez a MASTER 1-es állapotát már nem változtatja meg. Az órajel 1-0 átmenetekor a SLAVE, vagyis Q is 1-en lesz. Ilyen zavaró impulzusok, "tüskék" a gyakorlatban sokszor előfordulhatnak, ha a J és K bemenetet "hazardos" logikai hálózat vezérli. Az ilyen "tüskék" annyira rövid idejűek lehetnek, hogy oszcilloszkóppal nem láthatók és ezért úgy tűnhet, hogy a flip-flop rossz. Az ilyen "titokzatos" esetekben gondolni kell erre a hibalehetőségre is.

A flip-flop időadatai között szerepel az órajelre vonatkozó jelterjedési (billenési)idő is. Értelmezése az 5.25. ábra szerinti. Ehhez hasonlóan megadják a PRESET- és CLEAR-től, a Q kimenetig a jelterjedési időt is. Az órajel nem lehet tetszőlegesen rövid, minimális időtartamát szintén előírja a katalógus ($t_{p\text{clock}}$ általában 20 ns körül). Ugyanúgy a PRESET és CLEAR vezérlő jel minimális szélessége is adott (általában 25 ns körül).

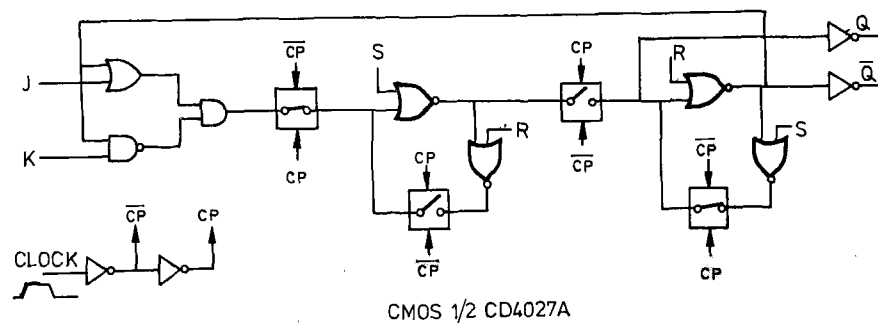


5.26. ábra

A komplementer MOS (CMOS, COS-MOS) JK MASTER-SLAVE flip-flopok felépítése eltér a TTL változatokétól. A CMOS áramkörök tárgyalásakor említettük, hogy ebben a rendszerben van egy különleges kapu: a TRANSMISSION GATE, áteresztő kapu. Ez a vezérlő jeltől függően, mint egy kapcsoló, vagy átenged egy jelet, vagy szakadásként viselkedik. Az áteresztő kapu felhasználásával a MASTER és a SLAVE flip-flop az eddigiektől eltérően 2-2 inverterből és 2-2 áteresztő kapuból épül fel (5.26. ábra, katalógus rajz).

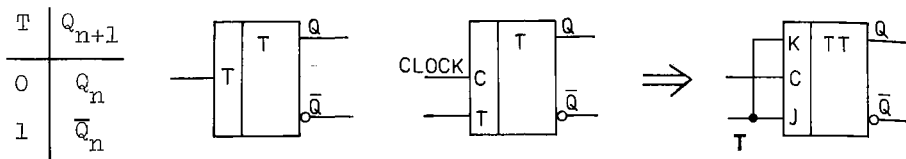
Mindegyik flip-flop akkor tárol, ha a visszacsatoló ágban lévő kapcsoló rövidzár, a hurokban a fázisfordítás 360° , a visszacsatolás pozitív, ezért a kimenet állapota fennmarad. Közben a soros kapcsoló szakadás. Beíráskor a soros kapcsoló bekapcsol, miközben a visszacsatoló ág kapcsolója megszakít. A MASTER és a SLAVE tároló ellenütemben működik és az egészet a CLOCK jel (C_p) vezérli. A bejövő órajelből inverterrel előállítják a negáltat is, ez a kapcsolók vezérléséhez szükséges. Amikor az órajel logikai 0 - ez a "nyugalmi" állapot - a SLAVE tároló vissza-

csatoló kapcsolója és a MASTER soros kapcsolója van bekapcsolva. Ha az órajel logikai 1-re ugrik, akkor a MASTER visszacsatoló ága záródik, soros kapcsolója megszakad, így a MASTER tárolja azt a jelet (ill. negáltját), melyet az órajel felugrása pillanatában a bemenetére vezetünk. A SLAVE ellenütemben működik, tehát az órajel logikai 1 időtartama alatt a soros kapcsoló rövidzár és a MASTER jele (ill. negáltja) a Q kimenetre jut. Amikor az órajel logikai 0-ra ugrik, a SLAVE tárolja ezt az újabb jelet. A J és K bemenethez tartozó kombinációs hálózat a JK flip-flop ismert igazságtáblázata szerinti működés megvalósításához szükséges (működése jól követhető). Itt is jelen van a SLAVE (Q) kimenetről a bemenetre menő visszacsatolás. Az univerzális célra gyártott CMOS típusokat szintén ellátják statikus vezérlő bemenetekkel, amelyek függetlenül hatnak az órajeltől és amelyeknek elsődleges szerepük van. A SET bemenetre adott logikai 1 a flip-flopot 1-be, a RESET-re adott 1-es pedig logikai 0-ba billenti. Az S és R jelet mind a MASTER-ba, mind a SLAVE-ba bevezetik, hogy az órajel bármely fázisánál a statikus bemenetek működtetésekor a tárolás megvalósuljon. A valóságos flip-flopok kapcsolása annyiban tér el az 5.26. ábrától, hogy a tárolókat megvalósító inverterek helyett NOR kapuk vannak, ezek második bemenetére jut az S, ill. R jel (lásd az 5.27. ábra katalógus-rajzát). Fontos tudnivaló, hogy a CMOS flip-flopokat vezérlő órajel-impulzus fel- és lefutási ideje nem lehet tetszőlegesen hosszú (max néhány μs).



5.27. ábra

A "T" flip-flop-nak egyetlen vezérlő bemenete van (T: TOGGLE, billentő). Ha ez a bemenet logikai 0 szintű, akkor a flip-flop állapota nem változik, ha logikai 1 szintű, akkor a flip-flop ellentétes állapotba billen:

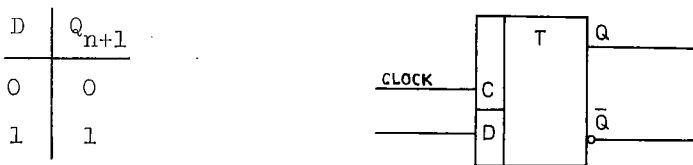


5.28. ábra

A T flip-flopoknak az órajellel szinkronizált változatát használják, ennek van fizikai értelme. Külön T-típust azonban nem gyártanak, hiszen a JK flip-flopból bármikor létrehozható. A "T" üzemmód a JK flip-flop egyik vezérlési esete. Ha a J és K bemenetet egymással összekötjük, akkor ez a közös pont használható T bemenetként (5.28. ábra). Ha logikai 0-t adunk rá, akkor mivel $J = K = 0$, az órajel megérkezésekor a flip-flop nem billen. Logikai 1-es vezérléssel, mivel $J = K = 1$, a flip-flop minden órajelre ellentétes állapotba billen, 2-es frekvencia osztóként működik. Azt mondhatjuk, hogy a számláló típusu 2-es frekvencia osztó nem más, mint egy logikai 1-gyel vezérelt T flip-flop.

5.2.3. A D flip-flop

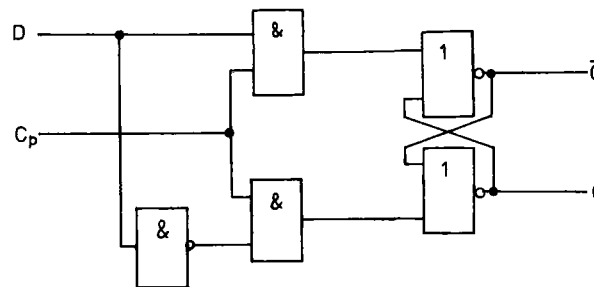
Egyetlen vezérlő bemenete van (D), igazságtáblázata igen egyszerű:



5.29. ábra

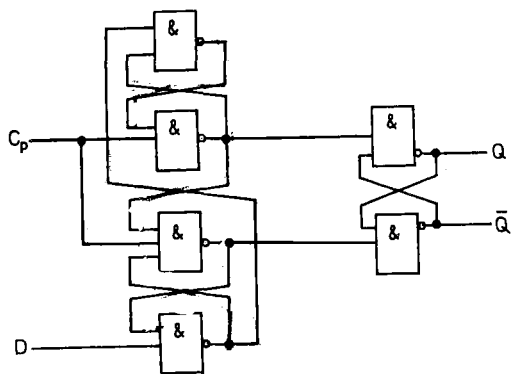
Amilyen a D bemenetre adott jel, olyan lesz a flip-flop állapota a következő ütemben. Fizikai értelme csak az órajellel szinkronizált változatoknak van (5.29. ábra), hiszen órajel nélkül a D flip-flop egyszerű késleltetéssel helyettesíthető lenne. Így viszont az n-edik ütemben, az n-edik órajel pillanatában a D bemenetre adott jel az n+1-edik órajel hatására íródik a kimenetre (billenti a tárolót), vagyis D flip-floppal 1 óraütemnek megfelelő késleltetést lehet megvalósítani. - Ezért kapta a D, DELAY (késleltetés) típus-nevet.

Az előbbi igazságtáblázat szerint működő, órajellel szinkronizált D flip-flopot legegyszerűbben az órázott RS flip-flop átalakításával hozhatunk létre (5.30. ábra).



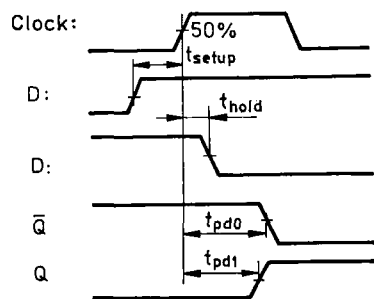
5.30. ábra

A SET bemenetből lesz a D bemenet, hiszen az erre adott logikai 1 hatására a tárolónak 1-be kell billennie (az órajel megérkezésekor). A RESET jelet egy D-ről jövő inverter állítja elő, ha $D = 0$, akkor a RESET 0-ba billenti a tárolót. Ennek az elrendezésnek nagy hibája, hogy amíg az órajel 1-ben van, a D-re adott jel azonnal megjelenik a kimeneten, és ha ez a jel változik, akkor Q értéke is változik. Valódi, órajellel szinkronizált működésről akkor beszélhetünk, ha minden óraütemben legfeljebb csak egy billenés jöhet létre. Hátránya ellenére jeltárolásra ("átmenő regiszterként", "LATCH"-ként) ezt a fajta D flip-flopot gyakran felhasználják. Jellemző típus a TTL-ben a 7475-ös és a 175-ös: egy tokban 4 ilyen tárolót tartalmaznak.



5.31. ábra

A D flip-flop javított változatát az 5.31. ábra mutatja. Ez a "belseje" pl. a TTL 7474-esnek. A D-re adott jel az óra impulzus felfutó élének hatására íródik ki a Q kimenetre. Ezután már a D bemenet hatástalan egészen a következő óra impulzusig. Ez a flip-flop un. ÉL-VEZÉRELT (EDGE-TRIGGERED) típus.



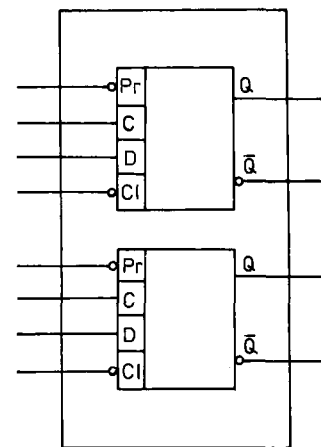
5.32. ábra

Az él-vezérlés azt jelenti, hogy a flip-flop billentése szempontjából az a vezérlő (D) jel a meghatározó, amely az órajel felfutó élének pillanatában van jelen. A valóságban természetesen a D jelet megfelelő előkészítési idővel (SETUP TIME) előbb a bemenetre kell készíteni még mielőtt az óra impulzus éle megérkezne és ezután még egy egészen rövid ideig változatlanul, a bemeneten kell tartani. Ez utóbbi idő a tartási idő (HOLD TIME). Az értelmezést az 5.32. ábra mutatja. Felmerül-

het a kérdés, hogy a D flip-flopot lehet-e számlálóként, 2-es frekvencia osztóként alkalmazni? Ilyenkor a flip-flopnak minden egyes órajel hatására ellentétes állapotba kell billennie. Ezt nyilvánvalóan úgy érhetjük el, hogy a \bar{Q} kimenetet összekötjük a D bemenettel és így mindig az előző állapot negáltját készítjük elő (megvalósítjuk a JK master-slave flip-flopokban meglévő összekötést a kimenet-negált és a bemenet között).

A TTL él-vezérelt D flip-flopokra általában előírják az órajel felfutási idejét is (nem lehet tetszőlegesen hosszú, mint más típusok esetében, legfeljebb 100...200 ns, ellenkező esetben "elvész" a jel a flip-flopban).

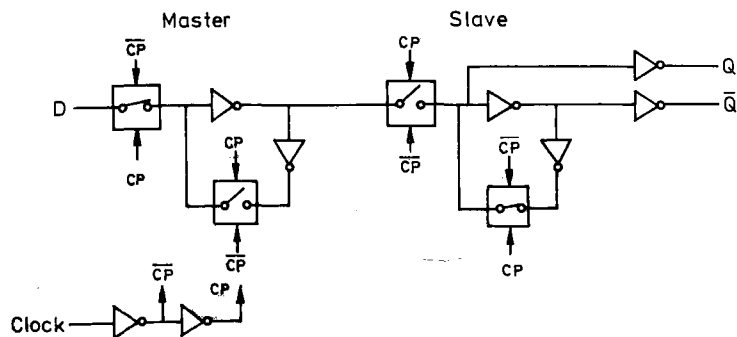
A JK flip-flopokhoz hasonlóan a D flip-flopokat is általában ellátják PRESET és CLEAR (vagy SET-RESET) statikus bemenetekkel, melyeknek prioritásuk van (a 7474 típus kettős flip-flopjának is van Pr és Cl bemenete, 5.33. ábra).



7474

5.33. ábra

A CMOS D flip-flopokat általában MASTER-SLAVE elven építik fel inverterek és a "transmission gate"-ek felhasználásával (5.34. ábra katalógus-rajza). Ebben az áramkör-családban a D flip-flop az alapvető tároló áramkör. Működése a kapcsolási rajz alapján jól követhető.



5.34. ábra

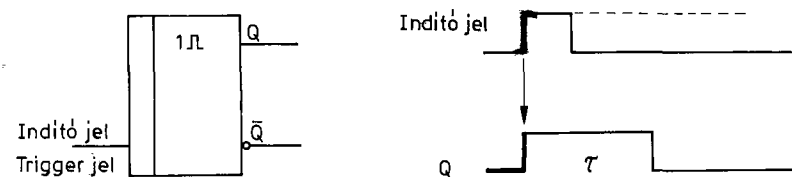
A TTL flip-flopok késleltetési ideje (az órajel bemenet-től a kimenetig) típusától, típuscsaládtól függően 6...20 ns, legnagyobb órajel frekvencia: 25...110 MHz, a CMOS flip-flopok átlagos jellemzői: 50...180 ns, ill. 4...15 MHz. Léteznek azonban extrém nagy sebességű ECL típusok is "előosztó" cél-ra, GHz-es (!) órajel frekvenciával.

5.3. Monostabil flip-flopok

5.3.1. Működési elv

A MONOSTABIL név arra utal, hogy a flip-flopnak, multi-vibrátornak "egy stabil állapota" van. Ez a stabil állapot rendszerint a 0 állapot: $Q = 0$ és $\bar{Q} = 1$. A bemenetre érkező indítójel "trigger-jel" hatására a monostabil flip-flop át-billen 1-be, $Q = 1$ és $\bar{Q} = 0$ lesz. Egy adott τ idő múlva - ezt általában külső RC időzítő elemekkel állíthatjuk be - a mono-stabil áramkör visszatér nyugalmi, 0 helyzetébe (l. az 5.35. ábra tömbvázlatát és idődiagramját) és csak újabb indítás ha-tására állít elő egy újabb τ idejű jelet. Az áramkör családok típusválasztékában általában többféle monostabil flip-flop és un. "TIMER" (időzítő) áramkör van különféle pontossági, ill. időzítési igények kielégítésére. Sokszor előfordul, hogy sem-miféle pontossági kikötésünk nincs (pl. egy mérőkört nullázó impulzus vagy egy START impulzus stb. előállításakor), ilyen-

kor kapukból is összeállíthatunk monostabil elemeket (szándé-kosan "hazárdos" hálózatot építünk, esetleg egy-két RC elem-mel kiegészítve).

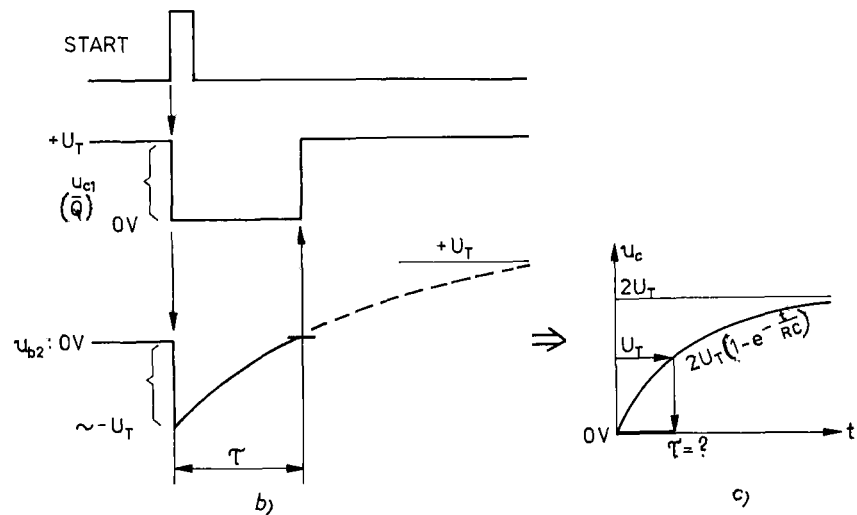
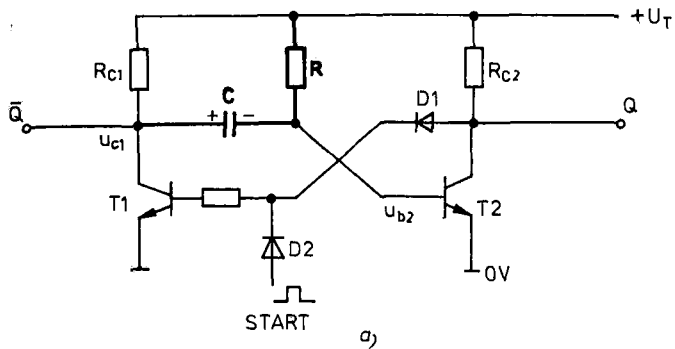


5.35. ábra

A "hagyományos" monostabil multivibrátort az 5.36a ábra mutatja: ez alkotja a legtöbb bipoláris monostabil IC típus alapáramkörét, "magját", működését ezért is célszerű megérte-nünk.

Alap állapotban T2 nyitott, mivel a pozitív tápfeszült-ségre csatlakozó R bázisáramot táplál bele. Kollektora gya-korlatilag 0 V-on van ($Q = 0$), ezért T1 - mivel nem kap nyi-tófeszültséget - zárt ($\bar{Q} = 1$), kollektora gyakorlatilag $+U_T$ feszültségen van. A C kondenzátor feltöltődik kb. U_T -re (bal oldala $+U_T$ feszültségen, jobb oldala T2 BE nyitófeszültségén, $+0,6$ V-on, ezt gyakorlatilag 0 V-nak tekintjük).

A D2 anódjára érkező pozitív START jel hatására T1 ki-nyit, kollektor feszültsége $+U_T$ -ről a 0 V felé tart. Ezt avál-tozást C "átviszi" (a kondenzátor, mint egy "akkumulátor", tartja egyenfeszültségét, amire előzőleg feltöltődött), ezért csökkenti jobb oldalának feszültségét is, lezárásba vezérelve T2-t. Emiatt T2 kollektor potenciálja növekszik és most már D1-en keresztül ez nyitja ki T1-et (a START jel már "elmul-hat"), a kapcsolás "átbillent" $Q = 1$, $\bar{Q} = 0$ állapotba. A jel-lemző feszültség jelalakokat az 5.36b ábra mutatja: amikor u_{C1} U_T -ről 0 V-ra ugrik, akkor a kondenzátor másik oldalán is ugyanilyen ugrás keletkezik: T2 bázis-potenciálja (u_{b2}) 0 V körüli értékről (0,6 V-ról) $-U_T$ körüli értékre "ugrik le". Ez-után a kondenzátor az R ellenálláson keresztül kezd "áttöl-tődni", az u_{b2} bázis-potenciál exponenciális törvényszerűség szerint $+U_T$ felé tart. Amikor eléri T2 nyitófeszültségét



5.36. ábra

(+0,6 V, közelítőleg 0 V) T2 kinyit, emiatt T1 lezár, a multivibrátor visszabillen nyugalmi ($Q = 0$ $\bar{Q} = 1$) helyzetébe. A keletkező impulzusszélesség (közelítően) ugy határozható meg, hogy az exponenciális töltődési függvényből meghatározzuk azt az időt, ami $-U_T$ -től $+U_T$ -re töltődés közben a 0 V eléréséig eltelik. A helyzet ugyanaz, mintha egy C kondenzátort 0 V-ról töltenénk fel R-en keresztül $+2U_T$ feszültségre (c ábra) és keresnénk azt az időpontot, amikor u_c eléri az egyszeres, $+U_T$ tápfeszültséget.

$$u_c = 2U_T (1 - e^{-\frac{t}{RC}}) = U_T \quad \text{ha } t = \tau$$

$$2U_T - 2U_T e^{-\frac{\tau}{RC}} = U_T$$

$$2U_T e^{-\frac{\tau}{RC}} = U_T$$

U_T -vel egyszerűsítve:

$$e^{-\frac{\tau}{RC}} = \frac{1}{2}$$

$$e^{\frac{\tau}{RC}} = 2$$

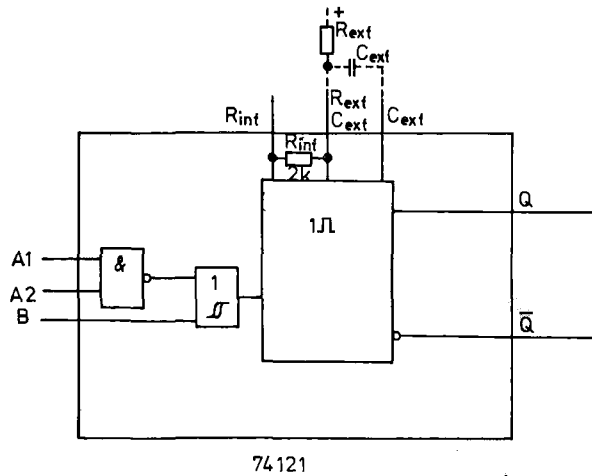
$$\tau = \underline{\underline{RC \ln 2}} \approx 0,7 \cdot RC$$

Ez, a monostabil billenési idejére adódó eredmény a monostabil multivibrátorok alapösszefüggése. Fontos észrevennünk, hogy U_T -vel egyszerűsítettük, ami fizikailag azt jelenti, hogy τ független a tápfeszültségtől, csak R-től és C-től függ (a valóságban marad egy kicsimértékű U_T -től való függés, mert nem számoltunk pontosan: T2 B-E nyitófeszültségét elhanyagoltuk). A "valóságos" integrált típusok ideje esetenként eltér az elvi értéktől különféle áramköri módosítások (diódák behelyezése, stb.) miatt.

5.3.2. Monostabil áramkörök

Az integrált monostabil áramkörök közül legismertebbek a FTL változatok, először ezek közül sorolunk fel néhány jellemző típust:

- 74121 egy tokban egy monostabil flip-flopot tartalmaz, vázlatát az 5.37. ábra mutatja. Meglehetősen pontos időzítés érhető el vele, mert a pontosság jóformán csak a kívülről csatlakoztatott R, C elemektől függ, maga az áramkör a tápfeszültség és hőmérséklet változás hatására legfeljebb 0,5% időhibát okoz. A kívülről hozzákapcsolt időzítő elemektől függően

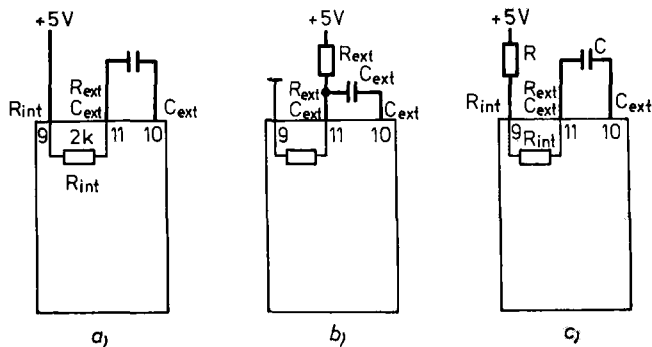


74121

5.37. ábra

40 ns...28 s kimeneti impulzusszélességet állíthatunk be ezzel a típussal. Az időzítő kondenzátort az $R_{ext} - C_{ext}$ és a C_{ext} kivezetésekre kell kapcsolnunk (10 pF...1000 μ F). Az időzítő ellenállást többféleképpen állíthatjuk be:

a) használhatjuk a belső, beépített $R_{int} = 2 k\Omega$ -os ellenállást, ehhez az R_{int} pontot kell +5 V-ra kötnünk (5.38a ábra). Mivel R_{int} monolitikus "IC ellenállás", nagy pontosságra nem számíthatunk;



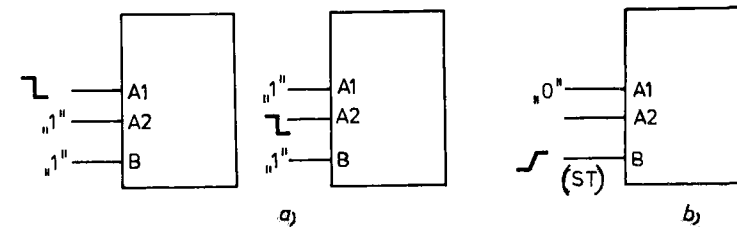
5.38. ábra

b) pontosabb, "reprodukálható" időzítéshez pontos külső ellenállást (R_{ext}) csatlakoztathatunk az $R_{ext} - C_{ext}$ pontra, az ellenállás másik végét +5 V-ra kell kötnünk (5.38b ábra) R_{ext} értéke 2 k Ω ...40 k Ω határértékű lehet, az impulzusszélesség

$$t_w = RC \ln 2$$

"elvi érték". R_{ext} ugyanazon a helyen van, mint az 5.36a kapcsolásban az R ellenállás. Ezért kell vigyáznunk a korlátozásokra: ha R túl kicsi netán rövidzár, akkor T2 a túl nagy bázisáram miatt tönkremegy, ha R túl nagy, akkor nem tud elegendő bázisáramot szolgáltatni T2 számára a kinyitáshoz nyugalmi helyzetben. A rövidzárát $R_{ext} - C_{ext}$ és a +5 V között tehát mindenképpen kerülnünk kell. Sokszor változtatható idő-késleltetésre van szükség, ilyenkor R_{ext} helyére változtatható ellenállást (potmétert) teszünk. A rövidzárást egy állandó, 2 k Ω -os soros ellenállással meg kell akadályoznunk, vagy

c) igénybe vehetjük soros korlátozóként az R_{int} ellenállást is (5.32d ábra).



5.39. ábra

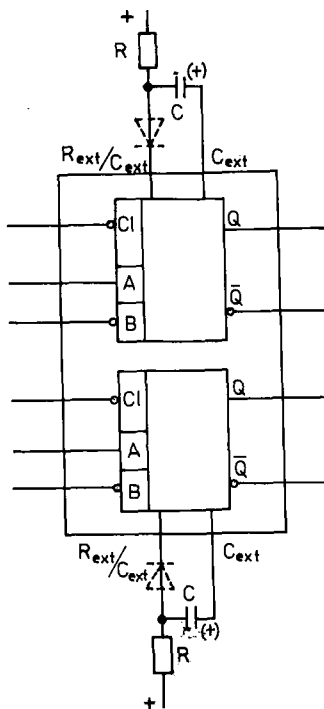
A monostabil flip-flop indítása több bemenet segítségével, többféleképpen mehet végbe:

a) az A1 vagy A2 bemenettel, él-vezérléssel 1-0 átmenettel (5.39a), miközben a másik A bemenet 1-en és a B bemenet is 1-en van;

b) a B bemenettel, amely Schmitt-triggeres, erre 0-1 átmenetet kell adnunk (tetszőlegesen lassu lehet), miközben vagy az A1-nek vagy az A2-nek 0-nak kell lennie (5.39b ábra):

A1	A2	B	Q
L	X	H	L
X	L	H	L
X	X	L	L
H	H	X	L
H	↓	H	⌈
↓	H	H	⌈
↓	↓	H	⌈
L	X	↑	⌈
X	L	↑	⌈

A 74121 hátránya, hogy nem újrainditható, nem lehet a visszabilenés után azonnal, ill. nagyon rövid időn belül indítani, mert pontatlanná válik az impulzus idő (nincs ideje C-nek feltöltődni).



5.40. ábra

- 74123 (74LS123) egy tokban 2 db újrainditható, RETRIGGERABLE változatot tartalmaz (5.40. ábra). Itt csak egy-egy A bemenet van, ettől eltekintve a vezérlés azonos a 121-es vezérlésével (indításhoz A-ra 1-0 átmenet, miközben B = 1, vagy B-re 0-1 átmenet, miközben A = 0). A B bemenet normál, nem Schmitt-triggeres kialakítású! Ez a típus statikus (aktív L) nullázó, CLEAR bemenettel is el van látva, ha erre logikai 0-t adunk, akkor Q = 0 lesz, bárhol is "tart" a monostabil flip-flop az időzítésben (felhasználása nem javasolt, mert - amiről nem ír a katalógus -, a CLEAR jel megszüntetése után az áramkör esetleg a kimeneti jelet "ott folytatja, ahol abbahagyta", azaz Q visszatér 1-re a hátralévő ideig!) A külső időzítő elemek elhelyezését is látjuk az 5.40. ábrán. A diódát (kisméretű Si típust) akkor kell elhelyeznünk, ha elektrolit kondenzátort használunk időzítésre (elkerüljük az ellenkező polaritású előfeszítést) vagy ha a CLEAR bemenetet is működtetni szándékozunk. A kimeneti impulzusszélesség (ha $C > \ln 2$, különben a katalógus diagramjait kell használnunk):

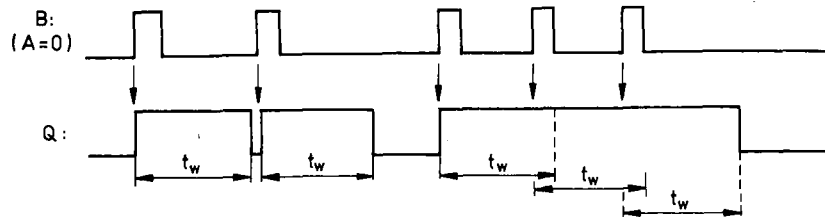
$$t_w = K \cdot RC \left(1 + \frac{0,7}{R}\right)$$

(R kΩ-ban
C pF-ban
 t_w ns-ban).

Az együtttható értéke dióda nélkül: K = 0,28,
diódával: K = 0,25,

bár meg kell jegyezni, hogy a számított és a valóságos időadatok a tapasztalat szerint nagy eltérést mutathatnak, a 74123 nem pontos időzítő áramkör! Az újraindithatóság viszont nagy előny, akár a 100%-ot is elérheti a kimeneti impulzus kitöltési tényező, az időzítés nem változik (legalábbis elvileg). A viszonyokat az 5.41. ábra idődiagramja szemlélteti: visszabilenés után azonnal újra elindíthatunk egy ciklust, sőt, ha akkor adunk indítást, amikor a kimenet még 1-ben van, akkor elől kezdődik a t_w ciklus, a kimenet nem tér vissza 0-ra, hanem az újraindítás pillanatától "számít" az újabb ciklus. (Óra-

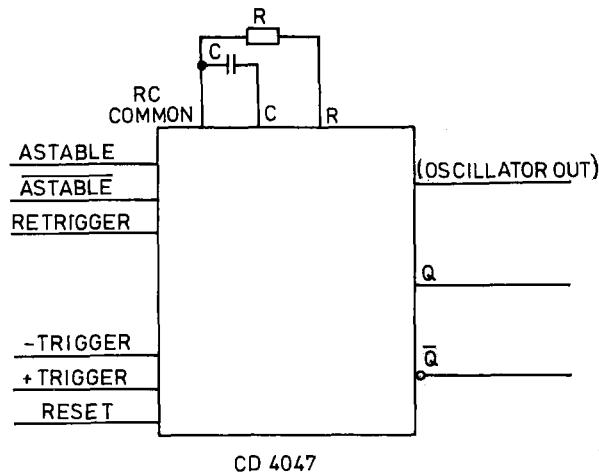
jelek vagy adat-jelek indikálására alkalmas pl. az ilyen újrainditható típus: a t_w időt kicsivel hosszabbra állítjuk a figyelni kívánt jel periódusidejénél és amikor jelsorozat érkezik, akkor az a bemenetet vezérelve állandóan újraindítja a flip-flopot, a kimenet folyamatosan 1 lesz, csak akkor billen vissza 0-ba, ha a jelsorozat végét ért).



5.41. ábra

A többi monostabil változat:

- 74122 (74LS122) újrainditható, 1 tokban 1 flip-flop van,
- 74221 (74LS221) kettős monostabil multivibrátor, a 121-eshez hasonló jellemzőkkel, Schmitt-triggeres (B) bemenettel, CLEAR lehetőséggel, bekötése egyezik a 123-assal, de nem újrainditható.



5.42. ábra

A CMOS család univerzális monostabil (és astabil) áramköre a

- CD4047.

Tömbvázlatát az 5.42. ábrán láthatjuk.

Indíthatjuk pozitív, 0-1 átmenettel, ekkor a +TRIGGER bemenetre adjuk az indítójelet (miközben a -TRIGGER bemenetet 1 szinten tartjuk), valamint indítható 1-0 átmenettel a -TRIGGER bemeneten (+TRIGGER = H). Újraindítás (retrigger-elés) a +TRIGGER és a RETRIGGER bemenetre adott egyidejű 0-1 átmenetre történhet. Az EXTERNAL RESET letiltja a kimeneti jelet ($Q = 0$ és $\bar{Q} = 1$ marad). Mivel az áramkör nagy bemeneti ellenállású CMOS, az R időzítő ellenállás is nagyobb lehet, mint a bipoláris változatoknál (kb. 1 M Ω -ig), a C időzítő kondenzátorra nincs felső határ, de polarizált, elektrolit vagy nagy veszteségű kondenzátor használata nem megengedett. Az impulzus idő közelítőleg:

$$t_M = 2,48 \cdot RC.$$

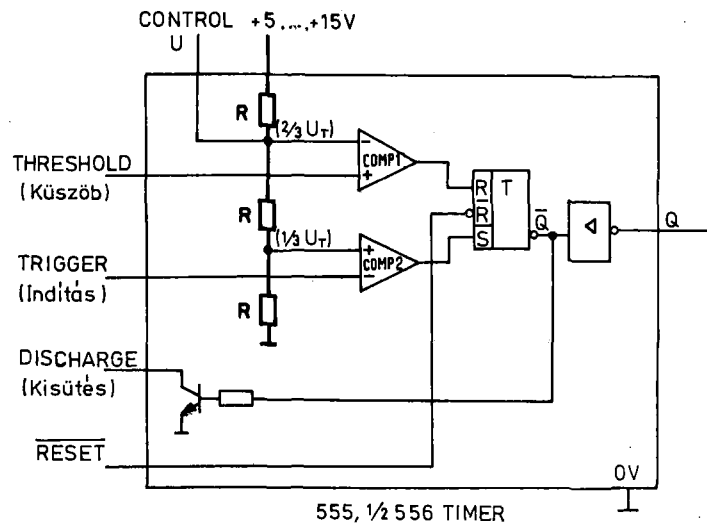
Hasonló típusok a 14500-as sorozatból pl. az

- MC 14528 B és 14538 B, amelyek egy tokban két "monoflopot" tartalmaznak, bemenetei, funkciói a 74123-hoz hasonlóak, de CMOS-ra jellemző R, C határértékekkel (R = 10 k Ω ... 1 M Ω , C = 100 pF...).

Előnyös, jól alkalmazható típus a: - 74C123, a TTL 74123 azonos kivezetés-elrendezésű "megfelelője" - természetesen kisteljesítményű változatban.

Közismert, sok készülékben használt TIMER IC az NE555 (SN72555...stb.), ill. az NE556, amely egy tokban két timert tartalmaz. Részben "lineáris", részben digitális áramkörökből épül fel. Az 555 szerkezetét, ill. az 556 egyik felének szerkezetét az 5.43. ábra vázolata mutatja. Három egyforma ellenállásból álló osztó hozza létre azokat a "belső feszültségeket", amelyeket két "analóg" komparátor hasonlít össze (nagy pontossággal), a két külső feszültséggel a THRESHOLD: küszöb, ill. a TRIGGER: indító jelszinttel. A komparátorok kimenete egy RS flip-flopot billent: a COMP1 komparátor kimenete a RESET, a COMP2 komparátor kimenete a SET-re megy. Amikor a flip-flop 0-ban van, kinyitja a DISCHARGE, kisütő tranzisztort. A

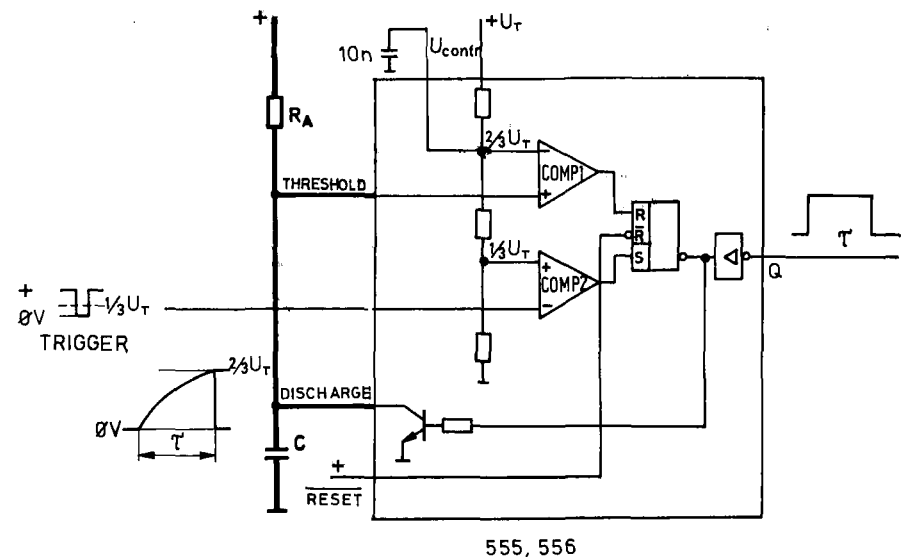
flip-flopot kívülről is lehet nullázni a $\overline{\text{RESET}}$ -re adott logikai 0-val. A feszültségosztó $1/3 U_T$ ill. $2/3 U_T$ feszültségeit, ha szükséges, módosítani lehet a CONTROL bemenetre adott feszültséggel.



5.43. ábra

Egy 555-tel (556-féllal) működő egyszerű monostabil kapcsolást mutat az 5.44. ábra. "Nyugalmi, stabil" helyzetben, amikor $Q = 0$, $\overline{Q} = 1$ az áramkörben levő kisütő tranzisztor rövidrezárja a külső C kondenzátort. A TRIGGER bemenetre adott "0" indító impulzus hatására a 2-es komparátoron keresztül a flip-flop SET jelet kap, 1-be billen, a kimeneten 1 jelenik meg, a kisütő tranzisztor "elengedi" a kondenzátort. Feszültsége a külső időzítő R_A hatására exponenciális törvényszerűség szerint növekedik U_T felé ($R_A C$ időállandóval). Amikor a kondenzátor feszültsége eléri a $2/3 U_T$ THRESHOLD feszültséget (ill. egészen kismértékben túllépi), akkor az 1-es komparátor kimenete pozitív lesz, a $\overline{\text{RESET}} = 1$ jel visszabillenteti a flip-flopot, miáltal a kisütő tranzisztor azonnal rövidrezárja a kondenzátort, a kimenet is visszatér 0-ba, lezajlott egy "monostabil" ciklus. Az időzítés tartama ($2/3 U_T$ -vel számolva):

$$t_M = 1,1 \cdot R_A \cdot C.$$

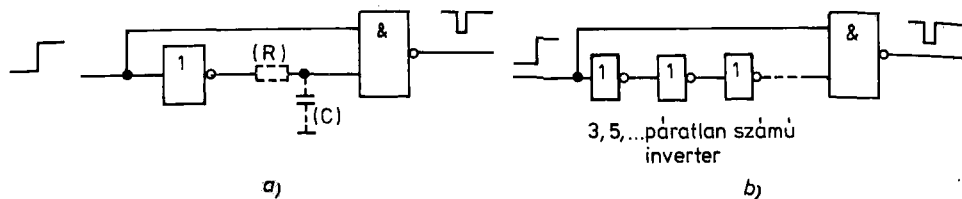


5.44. ábra

Az idő a tápfeszültségtől nem függ, mert annak megváltozásakor együtt változik az R-ek által előállított THRESHOLD feszültség és a kondenzátor töltődési sebessége! A tápfeszültség +5 V és +15 V közötti lehet. A monostabil ciklus megszakítható a $\overline{\text{RESET}}$ bemenet 0-ba vitelével. A $\overline{\text{RESET}}$ -re és a TRIGGER-re egyidejűleg adott impulzussal bármikor újraindítható az áramkör.

A meglehetősen precíz, kis bemeneti áramu komparátoroknak köszönhetően a monostabil-idő széles határok között beállítható R és C megválasztásával (R legnagyobb értéke 20 MΩ lehet!) és a beállított időzítés nagyon stabil (az áramkör által okozott "kezdeti hiba" 0,5%, hőfokégyüttható 30 ppm/°C).- Ezzel szemben az 555-556 nem "gyors" áramkör: 10 μs-nál rövidebb idő már nem állítható elő stabilan vele (emlékezzünk a TTL 74121 legrövidebb beállítható ideje 40 ns!).

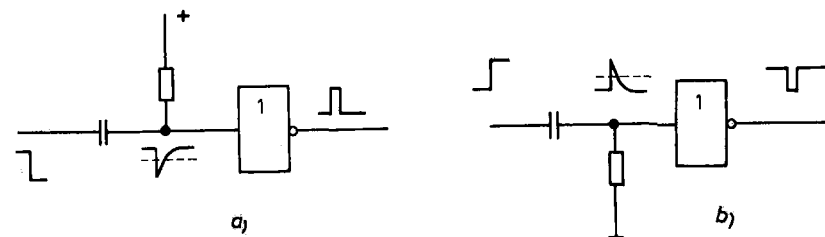
Kapu áramkörökből monostabil kapcsolásokat akkor állítunk össze, amikor az egyszerűség a cél és nincs kikötés az impulzusszélesség pontosságára (lényeg az, hogy egy felfutó vagy lefutó él hatására keletkezzen egy "határozott" impulzus, amelyet különböző hálózatrészek nullázására, tárolók beírásának vezérlésére stb. használhatunk).



5.45. ábra

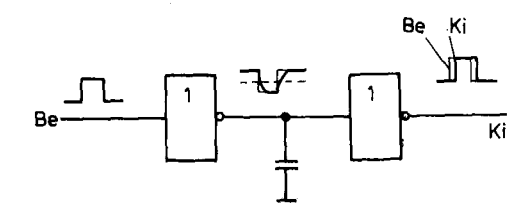
A TTL kapukkal - a kis bemeneti impedancia és kis késleltetési idő miatt rövidebb impulzusokat állíthatunk elő célszerűen (n.10 ns...100 ns). Rendszerint közös vezérléssel ellátott eltérő jelterjedési idejű hálózat-részek kimeneteinek kapuzásával állítunk elő 0-ba menő vagy 1-be menő rövid impulzusokat. Egy szokásos megoldást mutat az 5.45. ábra: pozitív felfutó élre ad a kimeneten egy 0-ba menő rövid impulzust. Amikor a bemenet 0, a NAND kapu kimenetén 1 van, az inverter kimenetén szintén 1. A bemeneti jel "felugrásának" pillanatában egy "rövid időre" a NAND kapu mindkét bemenetén 1 van, hiszen a vezérlő bemenetről is 1-et kap, és az inverter - késleltetés miatt - még nem ad 0 kimeneti jelet, erre az időre a NAND kimenet 0 lesz. Azután, amikor az inverter kimenete a bemeneti 1 hatására 0-ba kerül, a kimenet is visszatér 1-be. Az inverter késleltetését "mesterségesen" növelni lehet "lassító" kondenzátorral (C), esetleg soros ellenállással (R) kiegészítve. TTL-nél R értéke nem lehet nagy a bemenő nagy huzóáram miatt, legfeljebb 220...330 Ω (normál TTL), ill. 1 k...1,5 k Ω (LS TTL). A TTL 49-es sorozatában gyártanak külön erre a célra, DELAY ELEMENT áramköröket (pl. 49703: 2-szer 3 db soros inverter és 2 NAND kapu) ebből külső elemek igénybevétele nélkül az 5.45b ábra szerint alakíthatunk ki impulzus előállító kapcsolást (3 inverterrel 40...120 ns szélesség). Amennyiben nem 0-ba, hanem 1-be menő impulzusra van szükségünk, akkor a kimenetet még egy inverterrel egészíthetjük ki és akkor, ha a bemenetre nem 0-1, hanem 1-0 átmenet érkezik indítójelként, a bemenetet egészíthetjük ki inverterrel, az alapkapcsolást érdemes megtartani. Az ismertetett változat "integráló jellegű", esetenként "differenciálással" is előállítanak jelátmenetből impulzust. Erre mutat példát az 5.46. ábra: az a) változat 1-

0 átmenetre érzékeny, mert bemenete "nyugalmi helyzetben" 1-ben van, a felhuzó ellenállás miatt. A kimeneti polaritás attól függ, hogy invertáló vagy neminvertáló a felhasznált elem. A bemeneti 1-0 ugrás pillanatában a kondenzátor rövidzár és "átviszi" az ugrás-feszültséget, lehuzva ezzel 0-ra a TTL bemenetet. Röviddel ezután a kondenzátor feltöltődik (bemeneti pont 0 V, kondenzátor másik vége a felhuzó ellenállás miatt pozitív feszültségen), a kimeneti impulzus "megszűnik". Logikai 0-ba, ellenállás segítségével lehuzott bemenettel TTL-ben nehezebb a kapcsolásokat realizálni, mert a nagy bemeneti 0 huzóáram miatt csak kis ellenállásokat iktathatunk a bemenet és a föld közé (b ábra), az értékhatárokat már említettük. A kis ellenállás miatt az impulzus szélesség is kicsi lesz (oszilloszkópon sokszor nem is látható). A TTL változatoknál hátrány az is, hogy - mivel az áramkörök gyors működésűek és meglehetősen zavarérzékenyek - az esetleg a rendszerbe kerülő zavarjelek is "elindítják" a differenciáló kapcsolást (differenciálás = nagyfrekvenciák kiemelése!).

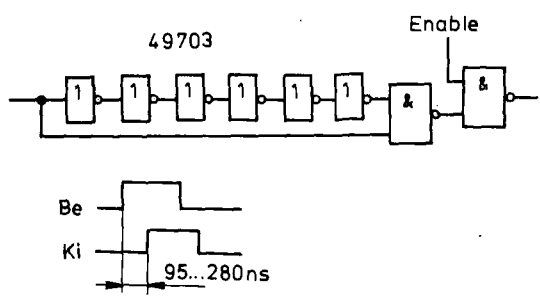


5.46. ábra

Sokszor előforduló feladat valamely impulzus késleltetése (legtöbbször nem fontos, hogy mennyivel csak az, hogy valamely esemény egészen biztosan később következzen be, mint egy másik, ilyenkor "késleltetett" impulzussal szinkronizálhatunk). Egy egyszerű késleltetőt mutat be az 5.47a ábra (célszerű a Schmitt triggeres bemenetű áramkörök használata). Felhasználhatjuk a már említett DELAY ELEMENT áramköröket (és természetesen bármely más inverter, ill. kapu áramköröket is a b ábra szerint, páros számú inverter "utkülönbség" beiktatásával).



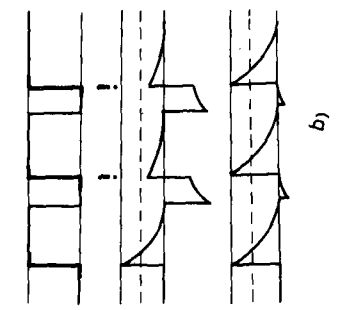
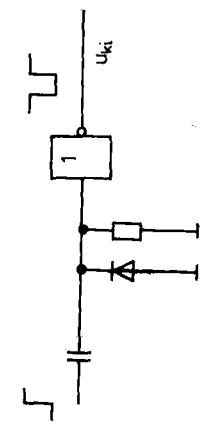
a)



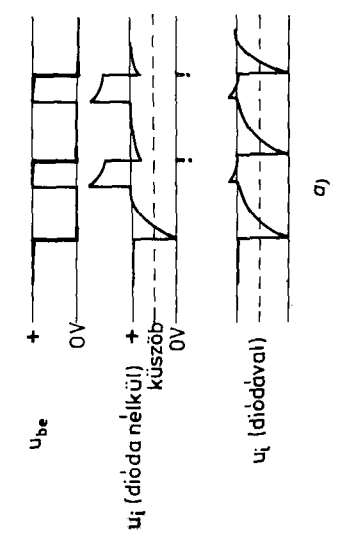
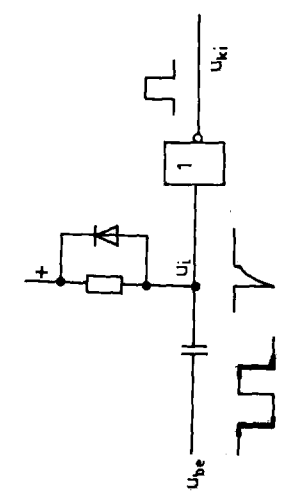
b)

5.47. ábra

A CMOS kapukkal sokkal könnyebb kivánci szélességű a polaritású impulzust előállítani: a bemenet árama gyakorlatilag zérus lévén, nem befolyásolja az időzítő RC tagok működését. Az 5.46. ábra "differenciáló", impulzusformáló kapcsolásait szinte korlátlanul nagy ellenállásokkal (1 MΩ-ig) megépíthetjük, de célszerű diódákkal kiegészíteni az 5.48. ábra szerint. Ha ezt nem tesszük meg, előfordulhat, hogy a kapcsolás nem minden bemeneti ugrást érzékel. Ennek oka az, hogy - pl. az a ábra kapcsolását figyelembe véve - a bemeneti jel 0-ba ugrásakor a kondenzátor lehúzza az inverter bemenetét, majd egy idő múlva (nagyágrendileg az időállandó elteltével) a kondenzátor jobboldalának potenciálja eléri az inverter küszöb feszültségét (ekkor lesz vége a kimeneti impulzusnak), ezután még pozitívabb lesz, a pozitív tápfeszültség közelébe kerül. Amikor a bemeneti jel visszatér 1-be, az inverter bemenetén a tápfeszültségnél jóval nagyobb pozitív feszültség alakul ki és



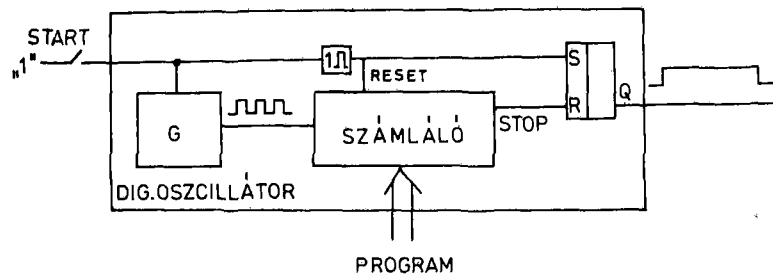
b)



a)

5.48. ábra

ha ezután rövid időn belül ismét egy 1-0 indító-átmenet érkezik, a kondenzátor jobb oldalának feszültsége "nem jöhet le" elegendően közel a 0 V-hoz, nem éri el az inverter bemeneti küszöbfeszültségét sem, nem keletkezik kimeneti impulzus. A dióda ezt a folyamatot megakadályozza: a vezérlőjel 1-be ugrásakor a kondenzátor jobb oldalának potenciálját nem engedi a tápfeszültség fölé, ezzel a kondenzátort "kisüti", így a következő indítójel a bemenetre kerülhet. Hasonló okból célszerű diódát helyezni a 0-1 átmenetre működő b. áramkörbe is.



5.49. ábra

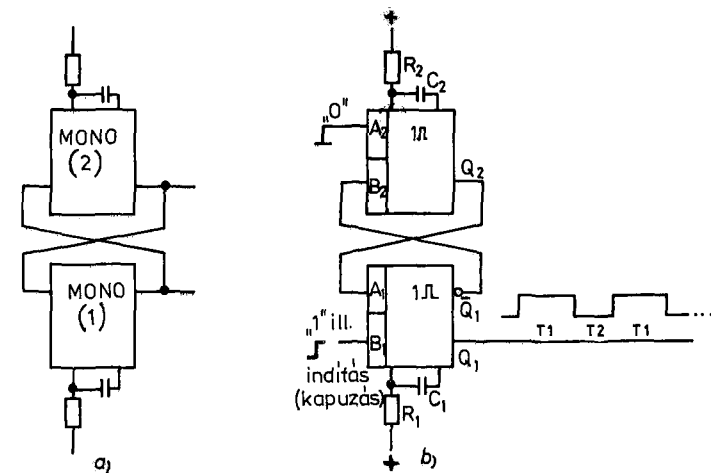
Hosszu idejű, pontos időzítésre (0...1 év) nem alkalmasak az RC taggal működő relaxációs áramkörök, erre a célra OSZCILLÁTOR/TIMER áramköröket gyártanak. Ezekben egy pontos frekvenciájú alap oszcillátor ("digitális oszcillátor" - lásd a következő pontot) van, ennek "periódusait" számolja egy hosszú, programozható számláló. Az előírt számérték elérésekor a kapcsolás visszatér alapállapotba (5.49. ábra, a működés hasonló egy "ébredtős kvarcóra" működéséhez). Ilyet felépíthetünk elemekből is, de kaphatók egyetlen tokba integrált "kész típusok" is (pl. MOTOROLA MC 14536B, 14541B...).

Az ismertetett monostabil elemeken, kapcsolásokon kívül természetesen még nagyon sok típus, változat létezik, ezeket katalógusokból, leírásokból kell adandó alkalommal megismernünk.

5.4. Astabil kapcsolások, digitális oszcillátorok

5.4.1. Működési elv

Az "astabil" kifejezés arra utal, hogy az áramkörnek egyetlen stabil állapota sincs (bistabilnak két, a monostabilnak egy stabil állapota van). Amikor az astabil hálózat valamely állapotát felveszi, azonnal elindul egy olyan folyamat, amely előbb-utóbb ezt az állapotot megszünteti, a kapcsolás ellenkező állapotba billen, de ezzel máris elindítja a visszabilenéshez vezető folyamatot és így tovább. Leginkább kézenfekvő példa a működés illusztrálására az egyik leggyakrabban alkalmazott astabil hálózat, amely két monostabil áramkör összekapcsolásával jön létre (5.50. ábra).



5.50. ábra

A két monoflop "egymást indítja": amikor az 1-es befejezte monostabil ciklusát, elindítja a 2-est, amikor ez visszatér alapállapotba, újra indítja az 1-est stb. Más elven is működhetnek astabil áramkörök, oszcillátorok (természetesen jelen esetben csak a digitális jelszintet szolgáltató "digitális oszcillátorokról" van szó), leggyakoribb elv az analóg függvény-

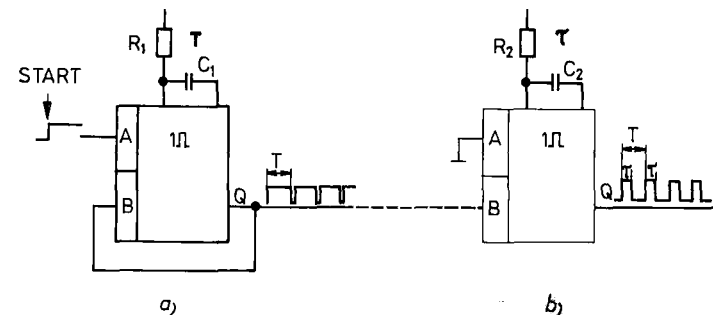
generátorokhoz hasonló: egy kondenzátort töltünk egy ellenálláson (vagy áramgenerátoron) keresztül, amíg a kondenzátor feszültsége egy adott szintet el nem ér, ezután ellenkező irányban töltjük, amíg egy másik szintet el nem ér, majd előlről kezdjük az egészet (a feszültségeket komparátorok Schmitt trigger stb. érzékelheti).

5.4.2. Astabil kapcsolások felépítése monostabil elemek összekapcsolásával

Meglehetősen nagy (kb. 0,5...2%-os) frekvencia stabilitás érhető el a monostabil áramkörök már megismert, egymást működtető összekapcsolásával, az 5.50. ábra szerint. Az ábra b részén egy célszerű, valóságos összekötésre látunk példát (Q_2-A_1 , \bar{Q}_1-B_2). Azért, hogy az oszcillátor biztosan elinduljon, célszerű a kezdeti időpontban (pl. egy készülékben egy mérési, működési ciklus kezdetén) indító jelet adni valamilyik "szabad" bemenetre (pl. B_1 -re egy 0-1 átmenetet, majd a B-t 1-ben tartani). Ennek hatására $Q_1 = 1$ lesz az R_1 és C_1 által meghatározott T_1 ideig, majd amikor visszabillen, a \bar{Q}_1 kimeneten keletkező 0-1 átmenet elindítja a 2-es monostabilt. Ez T_2 ideig 1-ben van, majd amikor visszabillen 0-ba, a Q_2 kimeneten keletkező 1-0 átmenet újra elindítja az 1-es áramkört és így tovább. A teljes periódusidő a T_1 és T_2 összege (elhanyagolva a flip-flopok belső jelterjedési késleltetési idejét). A kimeneti jel (amely egyébként bármelyik Q vagy \bar{Q} jele lehet) nem szükségképpen 50%-os kitöltési tényezőjű, hanem a $T_1(R_1C_1)$ és $T_2(R_2C_2)$ általunk beállított értékétől függ. A kapcsolás 1 Hz, esetleg az alatti frekvenciától néhány MHz-ig használható. A "szabad", ill. "indító" bemeneteket az oszcillátor kapuzására is használhatjuk: elektronikus vezérléssel ki- és adott fázisban bekapcsolhatjuk.

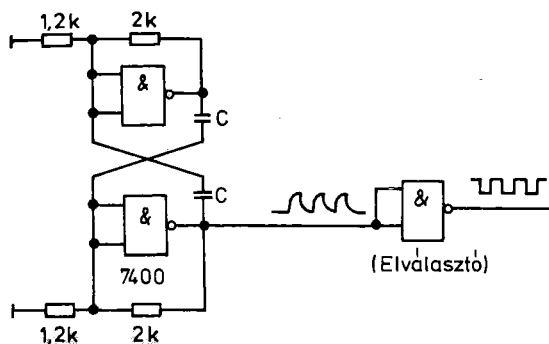
Önrezgő kapcsolást összeállíthatunk egyetlen monostabil flip-flopból is, ha újraindítható típust használunk (pl. a 74123). A Q kimenetet az 5.51. ábrán látható módon visszavezetjük az A indító bemenetre. Amikor a monostabil-idő letelik, Q visszabillen 0-ba, a keletkező 1-0 átmenet azonnal ujrain-

ditja a flip-flopot, Q ismét 1 lesz és így tovább. A kimeneten nagyon rövid 0-ba menő "tüskékből" álló impulzus sorozat keletkezik. A tüskék általában alkalmasak sorrendi áramkörök bilentésére, de ha szükséges, "megnyújthatjuk" egy kiegészítő monofloppal (5.51b ábra). Ennek az elrendezésnek nagy előnye, hogy egymástól független időzítő elemekkel állíthatjuk be a periódusidőt (R_1 és C_1) és az impulzus szélességet (R_2 és C_2). Az R_1 helyére változtatható ellenállást helyezve széles határok között változtatható frekvenciájú oszcillátort készíthetünk (kb. max 1:10...1:30 átfogással).



5.51. ábra

"Igénytelen helyekre", ahol a frekvencia pontos értéke nem játszik szerepet inverterekből, kapukból is előállíthatnak astabil kapcsolást, amely alapelvében egyik a két monostabilt alkalmazó kapcsolások működésével, csak aktiv elemként invertereket (kapukat), időzítőket a "rég, tranzisztoros" astabil multivibrátorokhoz hasonlóan csatoló RC tagokat használunk. Egy szokásos TTL inverteres (NAND kapus) elrendezést az 5.52. ábra mutat. Méretezési elvet, formulát nem mellékelünk, mert a frekvencia IC példánytól függően a legkülönbözőbb értékeket veheti fel (ha egyáltalán beindul az oszcilláció, az 1,2k és 2 kohmos ellenállásérték ebből a szempontból meglehetősen kritikus). A periódusidő nagyságrendje az RC időállandóból állapítható meg (a bemeneteken az ellenállások eredője közelítőleg 1 kohmos). frekvencia tartomány néhány 100 Hz-től néhány MHz-ig terjed (ésszerű időzítő elem értékekkel).



5.52. ábra

5.4.3. Egyetlen időzítő tagot igénylő kapcsolások

Az egymást vezérlő monostabil flip-flopokkal felépített kapcsolások hátránya, hogy két időzítő RC tag szükséges hozzájuk. Említettük már a digitális oszcillátorok másfajta felépítési lehetőségét: olyan kapcsolás kialakítását, amelyben egy kondenzátort töltünk felváltva egy alsó és egy felső feszültséghatár eléréséig.

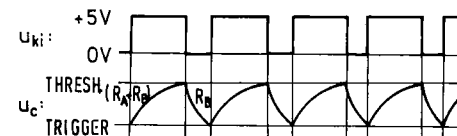
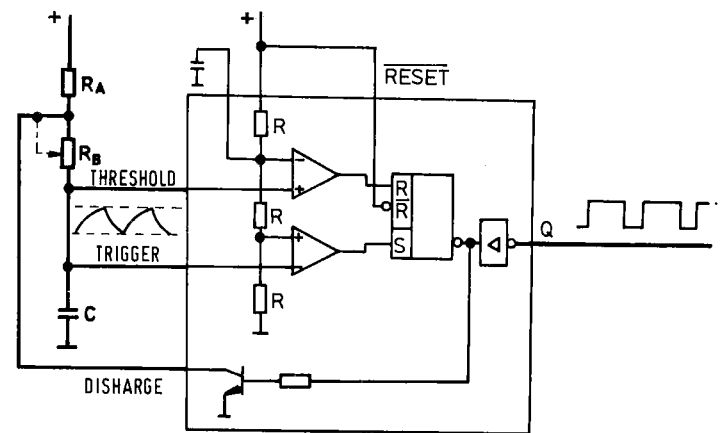
A CMOS család már megismert monostabil-astabil multivibrátora a CD 4047 is ebbe a kategóriába tartozik (tömbvázlatát az 5.42. ábrán láttuk). Szabadonfutó, oszcillátor üzemhez az ASTABLE és $\overline{\text{ASTABLE}}$ bemenetet H szintre kell kötnünk. Kapuzott (vagyis vezérlőjellel leállítható-indítható) oszcillátor működéshez az említett bemeneteket a következőképpen kell vezérelnünk:

- ha $\overline{\text{ASTABIL}} = \text{H}$, akkor engedélyezi az oszcillációt az
ASTABIL = H,
tiltja az oszcillációt az
ASTABIL = L;
- ha ASTABIL = L, akkor engedélyezi az oszcillációt az
 $\overline{\text{ASTABIL}} = \text{L}$,
tiltja az oszcillációt az
ASTABIL = H.

A Q és $\overline{\text{Q}}$ kimeneten pontosan 50%-os kitöltési tényezőjű négyszögjelet kapunk (a benne lévő 2-es frekvenciaosztó flip-flop-nak köszönhetően), a periódusidő közelítőleg:

$$T = 4,4 \cdot R \cdot C \quad (+5\% - 0\%),$$

ahol R, ill. C a kívülről rákapcsolt 1 db időzítő ellenállás, ill. 1 db időzítő kondenzátor értékét jelenti. Az OSCILLATOR OUT kimeneten kétszeres frekvenciájú, de nem 50%-os kitöltésű jelet vehetünk le egyidejűleg. A frekvencia stabilitásra akatalógus jó értéket ad meg: $\pm 2\%$ 100 kHz-en, és $\pm 0,5\%$ 10 kHz-en (természetesen ehhez még az időzítő elemek hibáját hozzá kell adnunk). Az időzítő elemekkel beállítható frekvencia tartomány határai közelítőleg 0,01 Hz...500 kHz.



5.53. ábra

Az NE555-556 timer IC felhasználásával szintén meglehetősen stabil oszcillátor építhető fel. Az astabil kapcsolást az 5.53. ábrán láthatjuk. Bekapcsoláskor a kondenzátor 0 V-

ról töltődik a pozitív tápfeszültség felé az $R_A + R_B$ ellenálláson keresztül, a kimenet l-ben van. Amikor a kondenzátor feszültsége eléri a THRESHOLD, küszöbfeszültséget, "jelez" a felső komparátor és visszabillenti a flip-flopot 0-ba. Ennek hatására a kisütő tranzisztor (DISCHARGE) kinyit és az R_A , R_B ellenállás találkozási pontját összeköti a 0 V-tal, a kondenzátor az R_B ellenálláson keresztül kisül, potenciálja csökken. Amikor eléri az alsó komparálási szintet, a TRIGGER feszültséget, a flip-flop ismét l-be billen, a kisütő tranzisztor "elenged" a kondenzátor ismét $R_A + R_B$ ellenálláson keresztül töltődik a tápfeszültség felé egészen addig, amíg a THRESHOLD feszültségét el nem éri és így tovább. A kondenzátoron tehát exponenciális töltődési és kisülési szakaszokból álló feszültség van, a két feszültséghatár: $1/3U_T$ és $2/3U_T$ között. Az oszcillációs frekvencia (amely nagymértékben független a tápfeszültségtől, mivel a referencia feszültségek és töltő feszültség együtt változik):

$$f = \frac{1,44}{(R_A + 2R_B)C}$$

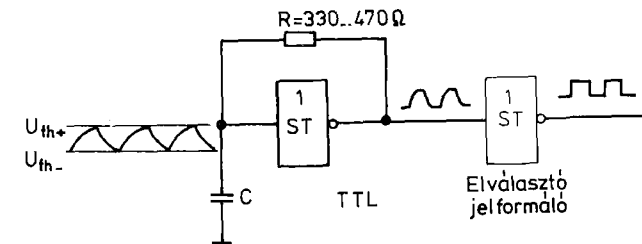
A kitöltési tényező (a kimeneti jel H-ban tartózkodásának ideje a teljes periódusidőhöz viszonyítva), ha az exponenciális töltődés-kisülés időfüggvényét háromszöggel helyettesítjük:

$$D = \frac{R_A + R_B}{(R_A + R_B) + R_B} = \frac{R_A + R_B}{R_A + 2R_B} \approx \frac{R_B}{R_A + 2R_B}$$

Az R_A értékét legtöbbször sokkal kisebbre választjuk, mint R_B értékét (R_A -ra körülbelül $1 \text{ k}\Omega \dots 100 \text{ k}\Omega$ érték kívánatos, $R_A + R_B$ megengedett maximális értéke $20 \text{ M}\Omega$, ilyenkor a kitöltés 50%-hoz közelít (de mindig nagyobb ennél). A biztonságosan beállítható frekvenciatartomány ezzel a típussal körülbelül 0,1 Hz és 200 kHz közötti. A Q kimenet jele TTL kompatibilis és a legnagyobb áram-terhelhetőség 200 mA (!). Az 555-ös IC-t általában 8 lábú mini-DIP (V-package) tokban árusítják, a "dupla" 556-ost 14 lábú "normál" (DIL) tokban. Ha változtatható frekvenciára van szükségünk, R_A helyére kis ellenállást célszerű helyezni ($1 \text{ k}\Omega$), R_B helyére pedig a változ-

tatható ellenállást ($2 \text{ k}\Omega \dots 1 \text{ M}\Omega$) az átfogás igen nagy lehet.

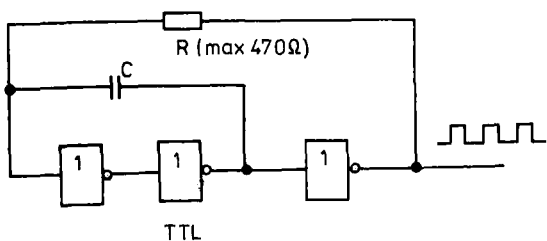
Hasonló elven működik, csak éppen sokkal pontosabb frekvenciával a Schmitt-triggerből felépülő egészen egyszerű kapcsolás (5.54. ábra).



5.54. ábra

Amikor a bemeneten a feszültség kisebb, mint az U_{th+} a kimenet l-ben van, így a visszacsatoló R-en keresztül töltődik a C kondenzátor. Amikor feszültsége eléri a felső küszöböt, az U_{th+} feszültséget, a kimenet 0-ba billen, a kondenzátort az ellenállás 0 V felé kisüti egészen addig, amíg feszültsége el nem éri az alsó küszöböt, az U_{th-} feszültséget. Ekkor a kimeneten ismét 1 lesz, C töltődik és így tovább. A kondenzátor feszültsége (a Schmitt trigger bemeneti feszültsége) a két küszöbszint között ingadozik, periódikusan. A frekvenciára ebben az esetben sem lehet felelősséggel formulát adni. Mivel a javasolt visszacsatoló ellenállás érték 330 ohm körüli, tehát jóval kisebb, mint a TTL-bemeneti impedancia, a periódusidőre az RC időállandó értéke ad nagyságrendi közelítést. Létezik a TEXAS típusválasztékában nagy bemeneti ellenállású Schmitt-triggeres NAND kapu (49713, 49813, bemeneti áramuk max $10 \mu\text{A}$!), ezek felhasználásával R értéke $30 \dots \dots 50 \text{ k}\Omega$ -ig növelhető, ill. változtatható ellenállás helyezhető a kapcsolásba, így széles határok között változtatható frekvenciájú oszcillátor készíthető.

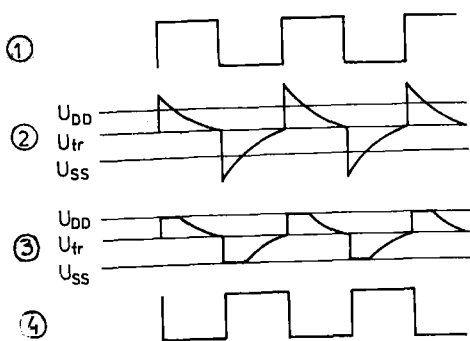
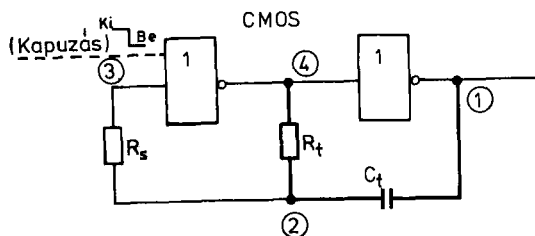
TTL inverterekből, kapukból is készíthető "igénytelen" felhasználásra egy-időállandós oszcillátor pl. az 5.55. ábra szerint.



5.55. ábra

Működése hasonló a Schmitt-triggeres változatéhoz, csak itt a "billenés" a kondenzátor által létesített pozitív visszacsatolás hatására jön létre.

A CMOS inverterekből, kapukból felépíthető egyszerű univerzális astabil kapcsolás szokásos változatát mutatja be az 5.56. ábra.



5.56. ábra

A kapcsolás működése a lerajzolt hullámalakokból jól követhető. R_S csak azért van az első kapu bemenete elé sorosan beiktatva, hogy a CMOS bemeneti védelem ne határolja az időzítő tagokon létrejövő feszültséget, és ezzel a frekvenciát lehető legnagyobb mértékben függetlenné tegye a tápfeszültségtől és a transzfer karakterisztika billenési, transzfer-feszültségétől. Mivel a CMOS bemeneti ellenállása gyakorlatilag végtelen, R_S a működésre egyéb, kártékony hatást nem gyakorol (R_S lehetőleg legyen 2-szerese R_t -nek, és mindkettő lehet 100 kΩ, MΩ nagyságrendben is!). A periódusidőre a katalógus a következő formulát adja:

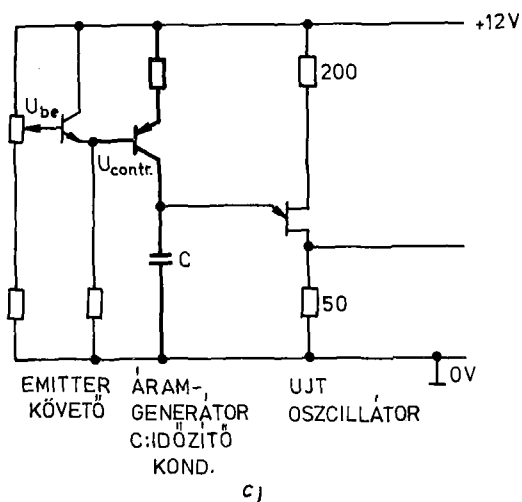
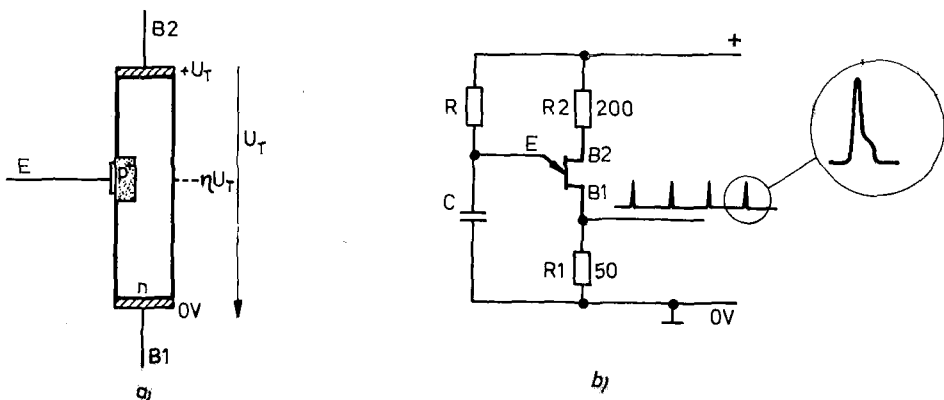
$$T = -RC \left(\ln \frac{U_{tr}}{U_{DD} + U_{tr}} + \ln \frac{U_{DD} - U_{tr}}{2U_{DD} - U_{tr}} \right) \approx 2 RC.$$

A kapcsolásban elektrolit kondenzátor felhasználása kerülendő, "normál" kondenzátorral az oszcillátor jól, univerzálisan használható kb. 0,1 Hz-től néhány 100 kHz-ig.

Igen alacsony frekvenciás periódikus jel előállítására az előbbi áramkörökkel nehézkes a nagy kondenzátor-kapacitás igény miatt (ami csak elektrolit kondenzátorral lenne megvalósítható, de ezek sok kapcsolásban eleve nem alkalmazhatók, ott ahol az időzítő ellenállásérték nagy és összemérhető a veszteségi ellenállásával, ahol megfelelne, ott az okoz gondot, hogy az elektrolit kondenzátorok kapacitása pontatlan, példányonként és időben nagyok az eltérések).

Az UJT (UniJunction Transistor: "kétbázisú dióda") alkalmazásával felépülő oszcillátort 0,01 Hz...100 kHz tartományban használják esetenként digitális berendezésekben. Az UJT eredetileg tirisztor, triac gyújtásához készült eszköz, egyszerűsített szerkezeti rajzát az 5.57a ábra mutatja N-típus esetére (pl. TIS 43). Alapja egy N-típusú gyengén szennyezett rudacska, végein B1 és B2 "bázis" kivezetésekkel. Ebbe a rudacskába egy erősen szennyezett P tartományt diffundálnak, ez az Emitter (lásd az ábrát!). Ha a rudra feszültséget adunk (B1 a 0 V-os pont, ehhez képest B2 pozitív), akkor - a gyenge szennyezés miatt - áram gyakorlatilag nem folyik B2-től B1 felé. A félvezetőben, mint egy "tömör ellenállásban" B1-től B2

felé haladva a potenciál 0 V -tól $+U_T$ -ig lineárisan változik. A bediffundált emitter helyén egy leosztott potenciál van jelen (mint egy potenciométer leszedő érintkezőjén):

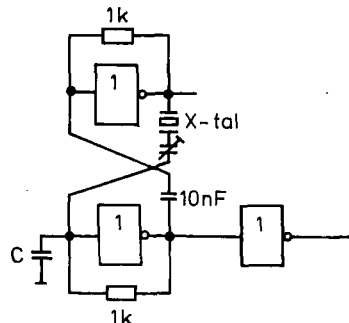
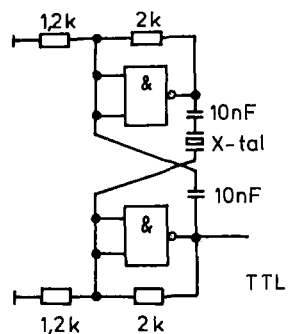


5.57. ábra

$$U_E = \eta \cdot U_T,$$

ahol η az illető UJT példányra jellemző érték, szokásosan $0,5 \dots 0,8$ körül van, nevezhetjük "leosztási tényezőnek". Helyezzük az UJT-t az 5.57b ábrán látható kapcsolásba! A tápfe-

szültség bekapcsolásakor C töltetlen, B1-B2 között a rudacska nagy ellenállása miatt gyakorlatilag nem folyik áram (így a külső, $n \cdot 10$ Ohmos ellenállások most nem játszanak szerepet), a bediffundált emitternél, belül a félvezetőben beáll a $+\eta U_T$ feszültség. A kondenzátor R-en keresztül töltődik a tápfeszültség felé, potenciálja és ezzel az UJT emitter potenciálja is emelkedik. Áram az emitterbe nem folyik, mivel a P típusú emitter (még) kevésbé pozitív potenciálú, mint ami a "csatornában" jelen van az emitter helyén, a PN átmenet záróirányú előfeszítést kap (az UJT-kat szándékosan úgy gyártják, hogy a záróirányú maradékáram extrém kicsi - pA - legyen, azért, hogy ne befolyásolja a kondenzátor töltését). Amikor a kondenzátor és ezzel együtt az emitter feszültsége eléri a félvezető rudban lévő ηU_T leosztott feszültséget, és még kb. $0,5 \dots 0,6\text{ V}$ -tal meg is haladja, akkor kinyit a PN átmenet, aminek az a következménye, hogy az emitter töltéshordozók tömegét "injektálja" az eddig gyakorlatilag töltéshordozómentes félvezetőbe (a pozitív töltéshordozók a B1 felé törekednek), ezért a "csatorna" emitter-B1 közötti szakaszának ellenállása lecsökken (több töltéshordozó = kisebb ellenállás), emiatt a belső leosztott potenciál lecsökken, emiatt mégjobban kinyit az emitter-csatorna átmenet, stb. a folyamat hirtelen, lavinaszerűen játszódik le. Az áram az emittertől a B1 felé nagymértékben megnövekszik (gyakorlatilag R1 korlátozza), a kondenzátor igen rövid idő alatt kisül. Ezután "elfogy" az emitter árama, a félvezető csatorna ismét nagy ellenállásúvá válik, besejében ugrás-szerűen beáll a "belső" ηU_T potenciál a PN átmenet lezár, és minden kezdődik előlről C töltődésével. Az oszcillátor kimeneti feszültségének az R1, vagyis a B1 pont feszültségét tekintjük: ezen nagyon rövid, a kondenzátor kisütő árama által R1-en létrehozott feszültség "tüskék" jelennek meg (sokszor oszcilloszkópon nehéz regisztrálni), ezek alkalmasak a legtöbb digitális áramkör meghajtására (célszerű az UJT oszcillátor tápfeszültségét $8 \dots 12\text{ V}$ -ra választani, ha erre van lehetőség, 5.57c ábra). Az E, emitter ponton lévő "fűrészfeszültséget" nehezebb hasznosítani, ui. ezt a pontot nem szabad terhelnünk ("lehuzzuk" az emitter-potenciált, lehet, hogy be sem indul az oszcilláció, mert sohasem éri el az $\eta U_T + 0,6\text{ V}$ -ot).



TEXAS ajánlás: $f=200\text{kHz}$ 500kHz 1MHz 2MHz 5MHz
 $C=3,3\text{nF}$ $1,2\text{nF}$ 680p 330p 120p

5.58. ábra

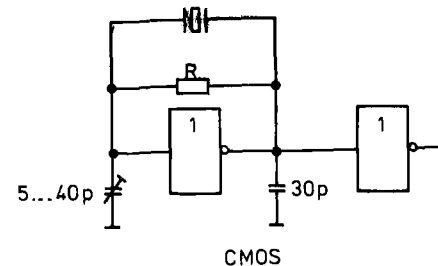
Ha nagy impedanciás leválasztásról gondoskodunk, akkor ez a korlátozás nem áll fenn. Mivel az UJT-k nagyon kis maradékáramuak, R nagyon nagy is lehet ($M\Omega$), de a kondenzátornak - az előbb említett okból - igen jóminőségűnek (kis veszteségűnek, nagy szigetelési ellenállásúnak) kell lennie (elektrolit típus szóba sem jöhet). A periódusidőre RC ad közelítő értéket, de mivel a tényleges idő függ a komparálási szinttől, azaz ζ -tól, a pontos beállítást mindig R finomszabályozásával kell elvégeznünk (az R_2 szerepe a kapcsolásban a hőkompenzálás: a frekvencia a hőmérséklet változásakor is stabil marad). Változtatható frekvenciájú oszcillátor céljára nagyon alkalmas az UJT-s oszcillátor: az R helyébe változtatható ellenállást téve sávváltás nélkül akár 4-5 nagyságrenden keresztül beállítható a periódus idő (időkapcsolás, készülékek indítása stb. célra). Az R helyébe feszültséggel vezérelhető áramgenerátort is tehetünk pl. VCO (Voltage Controlled Oscillator: feszültségvezérelt oszcillátor) céljára pl. az 5.57c ábra szerint.

5.4.4. Pontos oszcillátorok

Ebbe a kategóriába soroljuk az "igényes" célra felhasználható precíziós vagy bármely más szempontból többlet szolgáltatást nyújtó digitális típusokat. A továbbiakban néhány jellemző típust ismertetünk.

Nagy frekvencia pontosságú oszcillátorok: kvarc oszcillátorok

Az inverterekből, kapukból felépített, egymást "vezérlő" astabil kapcsolások frekvencia stabilitása, pontossága - ahogy láttuk - nem kedvező. Ugyanezeket a változatokat vagy hasonlókat kvarckristállyal felépítve "kvarc pontosságú" oszcillátorokat kapunk. Az 5.58. ábrán látható módon az egyik csatoló ágba iktatjuk be a kristályt, amelyet "soros rezonancián" működtetünk. Ez azt jelenti, hogy egy nagyon pontosan adott frekvencián (amely rendszerint ezrelék pontosan a házára van írva) a kvarc, mint egy soros, igen nagy jóságú rezgőkör, rövidzárként viselkedik. Ez csak a "sajátfrekvencián" (esetleg harmonikusán) teljesül, legkisebb mértékben eltérő frekvencián már ismét nagyvá válik a soros impedancia. Az oszcillátorban a jel visszacsatolása tehát csak ezen a specifikus frekvencián tud létrejönni, más frekvencián nem teljesül a rezgés feltétele. (A kvarckristályt lehet párhuzamos rezonancián is üzemeltetni, de ez kevésbé szokásos, különösen a digitális áramkörökben. A párhuzamos rezonancia frekvencia nem olyan stabil, mint a soros, mert a párhuzamos terhelések kapacitások is befolyásolják a sajátfrekvenciát). Előfordulhat, hogy a kvarc frekvenciától egy egészen kismértékben eltérő frekvenciára van szükségünk, ilyenkor párhuzamosan kapcsolt kondenzátorral próbálkozhatunk. A jó kvarckristály "nem hagyja magát elhangolni". Be kell látnunk, hogy ha egy kristály frekvenciáját külső elemekkel könnyen befolyásolhatjuk, akkor ez nem is lesz stabil, hiszen a külső terhelések időbeni, hőmérséklet okozta megváltozásai is könnyen elhangolhatják. Általában ezreléknél kisebb "finomhangolást" tűzhetünk ki célul. Az 5.58. ábrán lerajzolt egyszerű oszcillátorok frekvencia pontossága $10^{-4} \dots 10^{-5}$ körüli lehet (kvarc típustól függően).



5.59. ábra

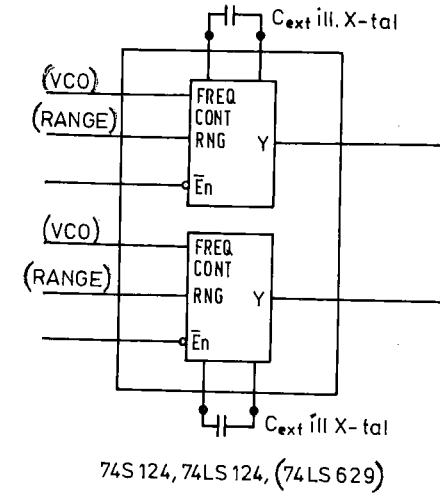
CMOS kapukból - nagyobb bemeneti impedanciájuknak köszönhetően - precízebb kvarc oszcillátorokat készíthetünk, pl. az 5.59. ábrán látható kapcsolásban. A C_0 és C_1 valamint R értékét a felhasznált kristály jellemzői határozzák meg, erre rendszerint táblázatokat közölnek.

Ugyancsak CMOS technológiával előállított kristályvezérelt TIMER-COUNTER a CD 4045A, amely oszcillátort és 21 fokozatu számlálóból álló frekvenciaosztót tartalmaz egyetlen tokban (2,097152 MHz-es kristály rákapcsolásával a kimenetén pontosan 1 Hz-es jelet szolgáltat). Hasonló, univerzálisan a használható OSZCILLÁTOR-COUNTER IC pl. az MC 14451-es. Kívülről RC hálózat vagy kvarc kapcsolható az oszcillátorához, és ennek frekvenciájától, valamint a $2^{11} \dots 2^{19}$ -es frekvenciaosztó megfelelő kimenetéhez való csatlakozástól függően egészen kisfrekvenciás jel vehető le róla (1 Hz...0,125 Hz).

VCO-k és egyéb szolgáltatásokat nyújtó oszcillátorok

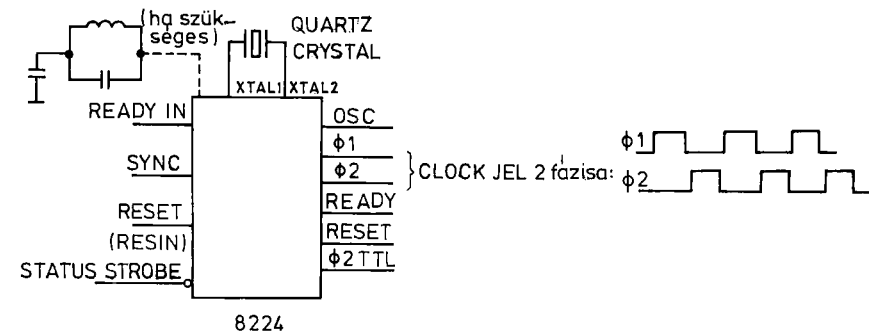
TTL VCO (Voltage Controlled Oscillator: feszültségvezérelt oszcillátor) vagy kvarc oszcillátor céljára készült IC a 74S124, 74LS124 (egy tokban két oszcillátor van, 5.60. ábra). Működési frekvenciatartományuk 0,12 Hz...85 MHz (!), ill. 0,12...30 MHz. VCO üzemmódban az időzítéshez csak egy külső elemre (kondenzátorra) van szükség (8,3 pF...500 μ F). A frekvencia finombeállítás (főleg PLL: Phase Locked Loop = fáziszárt hurrok VCO elemként való felhasználásához) a RANGE, ill. a FREQUENCY CONTROL bemenetekre adott digitális, ill. analóg jellel történik. A kimeneti jel 50%-os kitöltési tényezőjű. Az Enable bemenet 1-es vezérlésével tiltható, 0-val engedélyezhető az oszcilláció, az S változatnál az Enable = 0-ra fázishelyesen indul az oszcillátor.

A TTL oszcillátorok közül érdemes még megemlíteni a 74LS629 típust, amely funkciójában hasonló a 74LS124-hez, csak még ehhez képest is "javított" jellemzőkkel, kisebb fogyasztással.



74S124, 74LS124, (74LS629)

5.60. ábra



8224

5.61. ábra

A mikroprocesszor rendszerek órajelét (ütemjelét, amihez valamennyi időzítés igazodik) szinte minden esetben kvarc oszcillátor szolgáltatja. Általában kétfázisú órajelre van szükség, amelyek nem "lapolódnak át", elkülönülnek. Több, ma elterjedt rendszerhez gyártanak "célra orientált" CLOCK GENERATOR IC-eket. Példa erre az INTEL 8224 óragenerátora, amely a 8080 mikroprocesszorhoz készült és ahhoz illeszkedő jeleket ad ki, ill. fogad (5.61. ábra, a TANK feliratu kivezetést akkor használják, ha a kristályt "felharmonikus üzemben" működ-

tetik). Hasonló funkciójú a SIGNETICS 5009 típusa vagy a MOTOROLA 6800-hoz, hibrid technológiával készült 6871 típusa, amely tartalmazza a kvarc kristályt is, és közvetlenül a szükséges kétfázisú órajelet szolgáltatja, "fogadja" a rendszerhez tartozó vezérlőjeleket (indításhoz), ill. kiadja a rendszer működtetéséhez még szükséges egyéb jeleket.

Az astabil áramkörökről, digitális oszcillátorokról szóló ismertetésünk nem tekinthető teljesnek, csak az alap-típusokkal foglalkoztunk és az elvek bemutatása volt a cél. További eszközökkel a gyakorlatban találkozunk majd, ezek megismerése, működési elvük megértése remélhetőleg nem okoz majd komolyabb nehézséget.

IRODALOMJEGYZÉK

1. WICKES: INTEGRÁLT ÁRAMKÖRÖK HÁLÓZATOK LOGIKAI TERVEZÉSE
Budapest. Műszaki Könyvkiadó 1973.
2. TEXAS: TTL RECEPTEK
Budapest. Műszaki Könyvkiadó 1978.
3. TEXAS: ANALÓG ÉS ILLESZTŐ INTEGRÁLT ÁRAMKÖRÖK
Budapest. Műszaki Könyvkiadó 1979.
4. TIETZE-SCHENK: ANALÓG ÉS DIGITÁLIS ÁRAMKÖRÖK
Budapest. Műszaki Könyvkiadó 1973.
5. CSÁKÁNY-Dr.VAJDA: MIKROSZÁMITÓGÉPEK
Budapest. Műszaki Könyvkiadó 1977.
6. JANOVICS-TÓTH: A LOGIKAI TERVEZÉS MÓDSZEREI
Budapest. Műszaki Könyvkiadó 1973.
7. TEXAS: DESIGNING WITH TTL INTERGRATED CIRCUITS
Mc Graw-Hill
8. TEXAS: MOS/LSI DESIGN AND APPLICATIONS
Mc Graw-Hill
9. THE TTL DATA BOOK FOR DESIGN ENGINEERS
10. FAIRCHILD: BOOK TWO (Advanced Logic Book)
11. RCA: COS-MOS INTEGRATED CIRCUITS MANUAL
12. NATIONAL: CMOS INTEGRATED CIRCUITS (katalógus)
13. SIGNETICS: INTEGRATED CIRCUITS (katalógus)
14. MOTOROLA SEMICONDUCTORS: THE EUROPEAN CMOS SELECTION
15. FAIRCHILD: 34000 ISOPLANAR CMOS DATA BOOK
16. ELEKTRONIKUS LABORATÓRIUM MÉRÉSI UTASÍTÁSOK I. és II.
(KKVMF-1070, 1071)
17. MBL-SIGNETICS:FIELD PROGRAMMABLE DEVICES
Design Manual
18. ZSOM: LOGIKAI ÁRAMKÖRÖK 1. (KKVMF-1004)