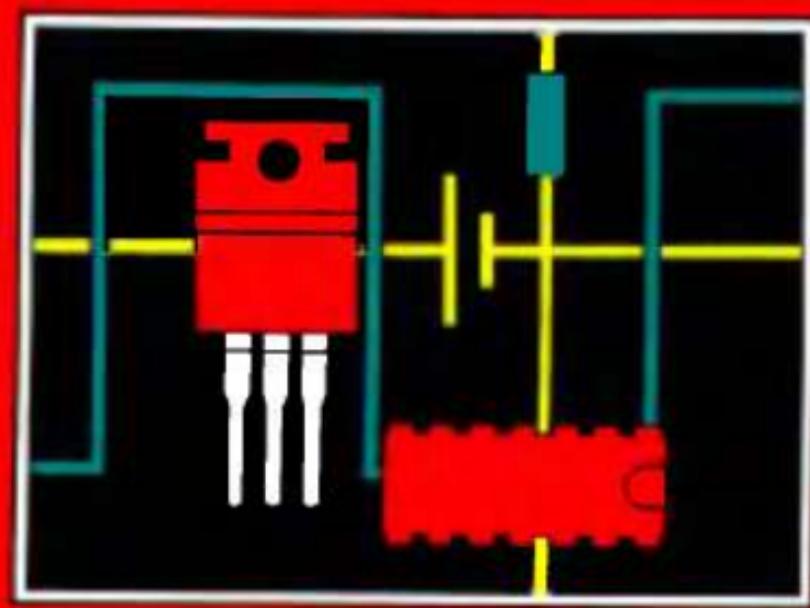


KLAUS BEUTH - OLAF BEUTH

AZ ELEKTRONIKA ALAPJAI



III. DIGITÁLIS ÁRAMKÖRÖK

MŰSZAKI KÖNYVKIADÓ

KLAUS BEUTH – OLAF BEUTH

AZ ELEKTRONIKA
ALAPJAI

III.

KLAUS BEUTH – OLAF BEUTH

**AZ ELEKTRONIKA
ALAPJAI**

**III. DIGITÁLIS
ÁRAMKÖRÖK**

MŰSZAKI KÖNYVKIADÓ
BUDAPEST, 1990

Az eredeti mű:

Beuth, K. — Beuth, O.: Elementare Elektronik by Vogel-Verlag, Würzburg
Copyright 1987 by Vogel-Verlag, Würzburg (BRD)

Lektorálta: Keresztély József okl. mérnök

© Hungarian translation Maklárné Fülöp Judit, 1990

ETO: 621. 3.049.77

ISBN 963 10 9622 4 (összkiadás)

ISBN 963 10 9625 9

Kiadja a Műszaki Könyvkiadó

Felelős kiadó: Szűcs Péter igazgató



90/4370 Franklin Nyomda, Budapest

Felelős vezető: Mátyás Miklós igazgató

Felelős szerkesztő: Molnár Ervin

Műszaki vezető: Bereczki Gábor

Műszaki szerkesztő: Bán Ferenc

A borítót tervezte: Székely Edit

A könyv ábráit rajzolta: Csermely Józsefné

A könyv formátuma: B/5

Ívterjedelme: 9,25 (A5)

Azonossági szám: 61601

MÜ: 4441-i-9093

Készült az MSZ 5601 és 5602 szerint

A kézirat lezárva 1989. december

Tartalom

27. Digitális alapkapsolások	8
27.1. Alapfogalmak	8
27.2. Logikai kapcsolatok	10
27.2.1. ÉS kapcsolat	10
27.2.2. VAGY kapcsolat	13
27.2.3. Tagadás	15
27.2.4. NAND (ÉS-NEM) kapcsolat	15
27.2.5. NOR (VAGY-NEM) kapcsolat	17
28. Digitális kódok	19
28.1. Számjegyek és számok ábrázolása	19
28.1.1. Kettes számrendszer	19
28.1.2. BCD kód (8-4-2-1 kód)	21
28.2. Egyéb bináris kódok	22
29. Kapcsolásanalízis	24
29.1. Általános ismeretek	24
29.2. Névleges kapcsolat	24
29.3. Tényleges kapcsolat	27
30. Logikai algebra	30
30.1. Alapfogalmak	30
30.2. A kapcsolások logikai függvényének meghatározása	31
30.3. A kapcsolás ábrázolása a függvényegyenlet alapján	32
30.4. Függvényegyenlet és kapcsolóérintkezők leírás	33
30.5. A logikai algebra felhasználási lehetőségei	34
31. A logikai hálózatok szintézise	35
32. Áramkörcsaládok	40
32.1. Relés kapcsolások	40
32.2. DTL áramkörök	40
32.3. TTL áramkörök	42
32.4. MOS áramkörök	43
32.5. ECL áramkörök	45
32.6. Az alacsony (Low) és magas (High) logikai szintek	45
32.7. Pozitív és negatív logika	47

33. Flip-flopok	50
33.1. A flip-flopok tulajdonságai	50
33.2. SR flip-flopok	52
33.3. T flip-flopok	54
33.4. JK flip-flopok	54
33.5. Master-slave flip-flopok	55
33.6. Alkalmazások	55
34. A digitális jelek szétválasztása és egyesítése	56
34.1. Adatszelektor, multiplexer, demultiplexer	56
34.1.1. 1 a 4-ből adatszelektor	56
34.1.2. 4 a 2×4-ből típusú adatszelektor	57
34.1.3. 8 a 4×8-ból típusú adatszelektor	58
34.1.4. 1-ből 4-et típusú demultiplexer	58
34.2. Címdekódoló	59
34.3. Sínrendszerek	60
35. Regiszterek és tárolók	62
35.1. Léptetőregiszter	62
35.2. Flip-flopos tároló	66
35.3. Írható-olvasható (RAM) tárolók	67
35.3.1. A tárolók felépítése	68
35.3.2. A tárolók jellemző adatai	70
35.4. Csak olvasható (ROM) tárolók	71
35.5. Programozható, csak olvasható (PROM) tároló	73
35.6. Törölhető, újraprogramozható, csak olvasható (EPROM) tároló	73
35.7. Ferritmagos tárolók	75
36. Számítási áramkörök	77
36.1. Frekvenciaosztó	77
36.2. Előreszámító	79
36.3. Visszaszámító	80
36.4. Dekadikus számítók	81
37. D/A és A/D átalakítók	83
37.1. Digitális-analóg (D/A) átalakító	83
37.1.1. A digitális-analóg átalakítás elve	83
37.1.2. Ellenállás-létrahálózatos D/A átalakító	84
37.2. Analóg-diigális (A/D) átalakító	86
37.2.1. Az analóg-digitális átalakítás elve	86
37.2.2. Az A/D átalakítók tulajdonságai	86
38. Aritmetikai áramkörök	88
38.1. Félösszeadó	88
38.2. Teljes összeadó	89
38.3. Párhuzamos összeadó	90
38.4. Összeadó-kivonó áramkör	91

39. Mikroprocesszorok és mikroszámítógépek	93
39.1. A mikroprocesszor mint univerzális áramkör	93
39.2. Aritmetikai-logikai egység (ALU)	93
39.3. Az ALU-tól a mikroprocesszorig	96
39.4. A mikroprocesszor építőelemei	97
39.4.1. Mikroprocesszortípusok	97
39.4.2. SAB 8080A típusú mikroprocesszor	100
39.5. A mikroprocesszorok kiegészítő egységei	101
39.6. Mikroszámítógép	102

27. Digitális alapkapcsolások

27.1. Alapfogalmak

Az analóg számítógépeken a számértékek ábrázolásához analóg mennyiségeket használnak. Analóg mennyiségek körül a legtöbbször az elektromos feszültséget használják úgy, hogy alkalmas megfeleltetést, vagyis léptéket választanak.

1. példa

Választott lépték: $1 \rightarrow 1 \text{ V}$.

Az 1,36 szám 1,36 V feszültségnek felel meg. Nagyobb számokhoz más léptéket kell választani, ha a „nagyfeszültségű” számítógép használatát el akarjuk kerülni.

2. példa

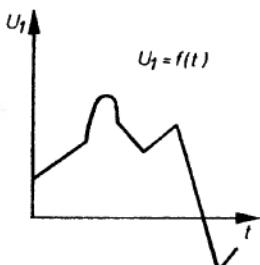
Választott lépték: $1 \rightarrow 10 \mu\text{V}$.

Az 10 530-hoz ekkor $105\,300 \mu\text{V} = 0,1053 \text{ V}$ feszültség tartozik.

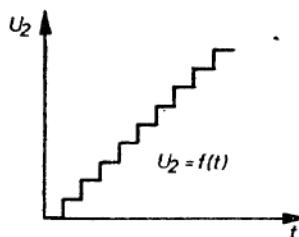
A jól bevált logarléc elvileg szintén analóg számológép, de az analóg mennyiség itt a hosszúság. Ha analóg számítógépről beszélünk, majdnem minden elektronikus analóg számítógépre gondolunk. Az ezekkel feldolgozott jeleket analóg jeleknek nevezzük.

Az analóg jelek — legtöbbször feszültségek — értékei egy megengedett tartományon belül tetszőleges értéket felvehetnek.

A 27.1. és 27.2. ábra az analóg feszültség időbeli lesolyását mutatja. Az analóg ábrázolás fogalmát a méréstechnikában is megtaláljuk. A mutatók mérőszerek (pl. feszültségmérő, árammérő) analóg mennyiség formájában mutatják a mért értéket. Az analóg mennyiség a skálán mért szögelfordulás. A skálán alapvetően tetszőleges érték előfordulhat, más kérdés, hogy ezt az értéket milyen pontosan tudjuk leolvasni. A szokásos mutatók órákat alapjában véve „analóg óráknak” is nevezhetnék. Térjünk rá a digitális jelekre! A „digitális” kifejezés a latin *digitus* (ujj) szóból származik. Az ujjal való számolás viszont azt feltételezi, hogy csak egész számmal dolgozunk,



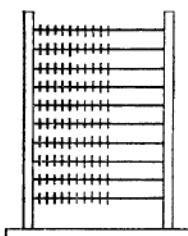
27.1. ábra. Példa az analóg jelre



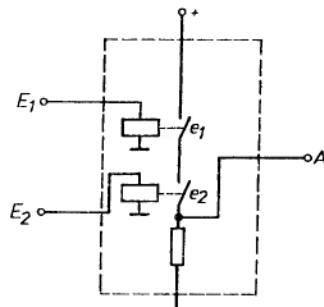
27.2. ábra. Lépcsős feszültség (analóg jel)

nincs semmiféle közbenső érték. Ilyen elven működő egyszerű számológép a régóta ismert abakusz (27.3. ábra).

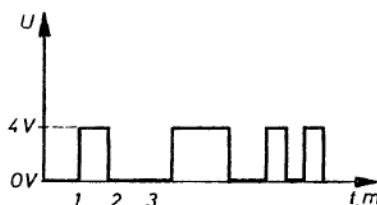
Ha digitális számítógépről beszélünk, akkor elektronikus számítógépre gondolunk, amely csak két állapotot „ismer”, pl. azt, hogy „van feszültség” vagy „nincs feszültség”. Valamely digitális jel alapvetően meghatározott számú (gyakran két), egymástól különböző állapotból áll. Általában két feszültségállapotot használnak, mivel több állapot megvalósítása nehézségeket okoz. Ezek az állapotok tetszőleges ritmusban követhetik egymást. A 27.4. ábra egy kétállapotú digitális jel időbeli változását mutatja. Az ábra alapján a digitális jelek egyik állapota 0 V, a másik +4 V.



27.3. ábra. Abakusz mint egyszerű digitális számológép



27.5. ábra. ÉS kapu



27.4. ábra. A digitális jel időbeli lefolyása

A feszültségek helyett alkalmazhatunk áramértekeket vagy mágneses állapotokat, sőt teljesen általánosan két tetszőleges fizikai állapotot.

A szokásos digitális jelek két különböző állapotból, pl. két különböző feszültségekből állnak. Egyéb állapot nincs megengedve.

A méréstechnikában ismertek a digitális mérőműszerek. Pontosabban azt kellene mondanunk, hogy „digitális kijelzésű” mérőműszerek. Ezek a mérőműszerek az eredményt számjegyekkel, vagyis decimális számokként jelzik ki. Közbenső érték itt sem lehetséges. Ha a tizedesvessző után egy számjegy van a kijelzőn, akkor a tizedesvessző utáni második helyről semmit nem mondhatunk, ill. felfelé vagy lefelé kell kerekíteni. A digitális kijelzésű mérőműszerek legfontosabb része a számláló. A kijelzett szám többsnyire valamely mért digitális jel impulzusainak száma.

Az időt számjegyekkel kijelző órát digitális órának nevezzük. Ha számokat digitális jelekkel ábrázolunk, akkor bizonyos megállapodásokra, ún. kódokra van szükség. A digitális jelek két állapotához két logikai állapot rendelhető hozzá. A logikai állapotokat a következőképpen jelöljük:

első logikai állapot: 0,
második logikai állapot: 1.

A logikai állapotok fizikai állapotokhoz való hozzárendelése alapjában véve tetszőleges lehet. Pl. a logikai 0 hozzárendelhető a 0 V-hoz és a logikai 1 a +5 V feszültséghöz, de fordított hozzárendelés is lehetséges. Magától értetődő, hogy áraméktékekhez is hozzárendelhetők a logikai állapotok.

A további vizsgálódások során a következő hozzárendelést választjuk:

Logikai 0 \rightarrow 0 V (föld),
Logikai 1 \rightarrow +5 V.

27.2. Logikai kapcsolatok

27.2.1. ÉS kapcsolat

A 27.5. ábra szerinti kapcsolás bemenetén csak az előzőekben megadott hozzárendelés szerinti jelek, ill. logikai állapotok léphetnek fel.

A bemeneti kombinációk – esetek – a 27.1. táblázat szerintiek lehetnek. Alapjában véve négy különböző eset lehetséges:

1. eset: minden bemenet állapota 0;
2. eset: az E_2 bemenet állapota 0,
az E_1 bemenet állapota 1;
3. eset: az E_2 bemenet állapota 1,
az E_1 bemenet állapota 0;
4. eset: minden bemenet állapota 1.

27.1. táblázat. Egy logikai áramkör igazság-táblázatának esetei

Eset	E_1	E_2	A
1	0	0	
2	0	1	
3	1	0	
4	1	1	

Gondoljuk most meg, hogy milyen feltételek mellett lép fel 1 állapot az A kimeneten! Az e_1 és e_2 érintkezőknek zártaknak kell lenniük. Tehát minden relének behúzott állapotban kell lennie. Ez azonban csak akkor lehetséges, ha minden bemeneten egyidejűleg 1 állapot (+5 V) van. Tehát csak a 4. esetben van az A kimeneten 1 állapot. A 27.2. táblázat az áramkör logikai viselkedését adja meg. Az ilyen típusú táblázatot igazságtáblázatnak nevezzük.

Az ÉS kapu kimenetén csak akkor jön létre 1 állapot, ha minden E_2 , minden E_1 bemeneten 1 van.

27.2. táblázat. Az ÉS kapu igazságtáblázata

Eset	E_1	E_2	A
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

Ezt a logikai kapcsolatot ÉS kapcsolatnak nevezzük. minden áramkör, amely megfelel a 27.2. táblázat szerinti igazságtáblázatnak, ÉS kapcsolatot állít elő, és ÉS kapunak nevezzük.

Az ÉS kapuk igen eltérő kivitelűek lehetnek. Ma többnyire integrált félvezető áramkörök által állítják elő őket. Az ÉS kapcsolat matematikailag a logikai algebra segítségével fejezhető ki. Érvényes a következő kifejezés:

$$A = E_1 \cdot E_2$$

az ÉS kapcsolat jele (szokásos még a \wedge jel is).

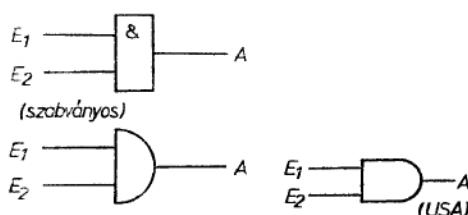
Az irodalomban egyéb jelöléseket is találunk az ÉS kapcsolat kifejezésére.

Az iménti egyenletet a következőképpen is szokták írni:

$$A = E_1 \wedge E_2, \quad A = E_1 \& E_2.$$

Az ÉS kapu ábrázolásához nem célszerű a teljes kapcsolást felrajzolni. A szabványos (szimbolikus) kapcsolási jeleket a 27.6. ábrán látjuk.

Az ÉS kapuk három vagy több bemenettel is készülhetnek. minden további bemenet megduplázza az igazságtáblázatban levő lehetőségek számát, mivel a bemenet 0 vagy 1 lehet (27.3. táblázat).



27.6. ábra. A kétbemenetű ÉS kapu rajzjelei

Pi. három bemenet esetén 8, négy bemenet esetén 16 és 5 bemenetnél 32 eset adódik. A 27.4. táblázat a 4 bemenetű ÉS kapu igazságtáblázatát mutatja be. Az igazságtáblázat kitöltésének sorrendje – az esetek felsorolása – alapjában véve tetszőleges. minden egyes esetet meg kell azonban vizsgálni, és egyiknek sem szabad kétszer szerepelnie.

27.3. táblázat. A hárombemenetű ÉS kapu igazságtáblázata

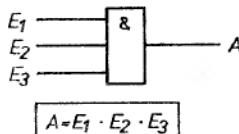
Eset	E ₃	E ₂	E ₁	A
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	0
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

27.4. táblázat. A négybemenetű ÉS kapu igazságtáblázata

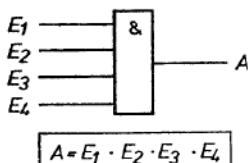
Eset	E ₄	E ₃	E ₂	E ₁	A
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	0
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	0
11	1	0	1	0	0
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	1

Hogy ne nehezítsük a munkát feleslegesen, ajánlatos követni a következő sémát: Az egyik bemenet (pl. E₁) egyik esetről a másikra változtatja az állapotát. A következő bemenet (pl. E₂) állapotát mindenketesével, a harmadik bemenetét (pl. E₃) minden négyesével változtatjuk. A negyedik bemenet (pl. E₄) állapotát minden nyolcasával kell változtatni.

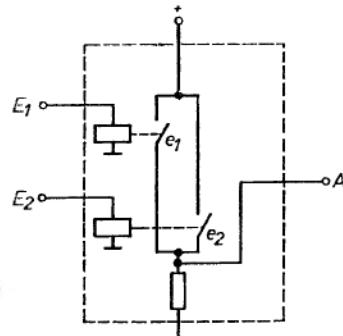
A 27.4. táblázatban is ezt az eljárást követtük. A 3 és 4 bemenetű ÉS kapuk szimbolikus rajzát és a hozzá tartozó logikai algebrai egyenleteket (kapcsolási egyenleteket) a 27.7. és 27.8. ábrán mutatjuk be. A magyar ÉS kapu kifejezés helyett esetenként használják az angol AND gate, vagy a „kevert” AND kapu kifejezést is.



27.7. ábra. A hárombemenetű ÉS kapu rajzjele és logikai egyenlete



27.8. ábra. A négybemenetű ÉS kapu rajzjele és logikai egyenlete



27.9. ábra. VAGY kapu

27.2.2. VAGY kapcsolat

Készítsük el az igazságáblázatot a 27.9. ábra szerinti kapcsoláshoz! Mivel az áramkörnek 2 bemenete van, 4 eset fordul elő (27.5. táblázat). Mely esetben lesz most 1 állapot az A kimeneten? Elegendő, hogyha az e₁ vagy e₂ érintkező közül az egyik zárt, akkor a kimeneten 1 (+5 V) lesz. Ez azt jelenti, hogy vagy az E₁, vagy az E₂, vagy mindenből 1 állapotnak kell lennie. Ekkor a 27.6. táblázat szerinti igazságáblázat adódik.

27.5. táblázat. A VAGY kapu bemeneti kombinációi

Eset	E ₂	E ₁	A
1	0	0	
2	0	1	
3	1	0	
4	1	1	

27.6. táblázat. A VAGY kapu igazságtáblázata

Eset	E_2	E_1	A
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

Ezt a logikai kapcsolatot **VAGY** kapcsolatnak nevezzük. minden áramkör, amely a 27.6. táblázat szerinti igazságtáblázatot teljesíti, VAGY kapcsolatot állít elő, tehát **VAGY** kapu.

Az ÉS kapukhoz hasonlóan a VAGY kapuk is igen sokféleképpen alakíthatók ki, de többnyire integrált áramkörként készítik.

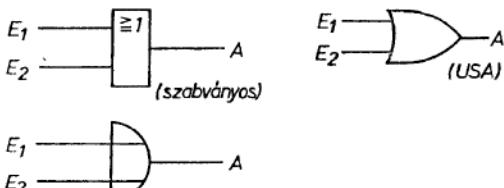
A VAGY kapcsolat matematikailag a logikai algebra segítségével fejezhető ki.

$$A = E_1 + E_2.$$

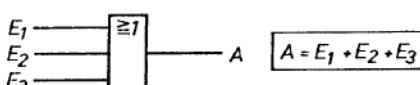
+ a VAGY kapcsolat jele (szokásos még a \vee jel is).

Az irodalomban a szabványos jelölésen kívül másképpen is jelölik a VAGY kapcsolatot. A VAGY kapuk ábrázolására a szabványos rajzjelet (27.10. ábra) alkalmazzák. A VAGY kapukat három vagy több bemenettel is készítik. A 3 és 4 bemenetű VAGY kapuk szimbolikus rajzát és a hozzájuk tartozó logikai algebrai egyenleteket (kapcsolásegyenleteket) a 27.11. és 27.12. ábra szemlélteti.

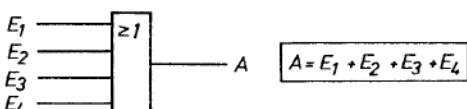
A VAGY kapu kimenetén mindenkor van 1 állapot, ha legalább egy bemenetén 1 állapot van.



27.10. ábra. A kétbemenetű VAGY kapu rajzjele



27.11. ábra. A hárombemenetű VAGY kapu rajzjele és logikai egyenlete



27.12. ábra. A négybemenetű VAGY kapu rajzjele és logikai egyenlete

27.2.3. Tagadás

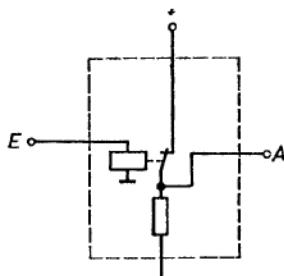
A 27.13. ábrán látható kapcsolásnak csak egyetlen bemenete van. Ha ezen a bemeneten a 0 állapotnak megfelelően 0 V van, akkor a relé nem húz be, és a kimeneten az 1 állapotnak megfelelő +5 V lesz. Ha a bemenetre az 1 állapotnak megfelelően +5 V-ot kapcsolunk, akkor a relé behúz, a bontóérintkező nyit, és a kimeneten 0 állapot lesz. Csak ez a két eset lehetséges. Ezzel a 27.7. táblázatban látható igazságtáblázat adódik. A kimeneten mindenkor a bemenettel ellentétes állapot jelenik meg. Ezt a logikai függvényt tagadásnak, negációnak vagy inverzióknak nevezzük. Azt a kapcsolást, amely ezt a logikai kapcsolatot előállítja, NEM kapunak, negációs kapunak vagy invertereknek nevezzük. A NEM kapu rajzjelei a 27.14. ábrán láthatók.

A NEM kapu kimenetén mindenkor a bemenet ellentétes állapotára reagál.

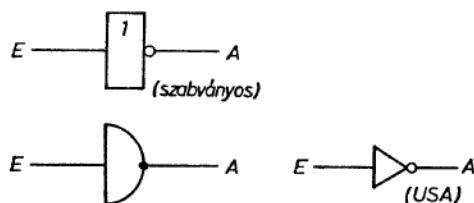
A NEM kapu angol neve NOT gate vagy inverter.

27.7. táblázat. Az inverter igazságtáblázata

Eset	E	A
1	0	1
2	1	0



27.13. ábra. NEM kapu



27.14. ábra. A NEM kapu rajzjele

Az ÉS kapcsolat, a VAGY kapcsolat és a NEM kapcsolat a logikai algebra három alapművelete. Az ezeket megvalósító áramköröket együttesen alapkupuknak nevezzük. Az alapkupuk segítségével az összes elképzelhető logikai kapcsolat létrehozható (l. még a 30. fejezetet).

27.2.4. NAND (ÉS-NEM) kapcsolat

Gyakran alkalmazott művelet az ÉS kapu és a NEM kapu összekapcsolása. Vajon hogy néz ki az összekapcsolás igazságtáblája?

A 27.8. táblázat az ÉS kapu igazságtáblázatát mutatja. Az ÉS kapu A kimenete azonban a NEM kapu bemenete is. A NEM kapu Z kimenetén mindenkor a bemeneti

27.8. táblázat. Az alapkapukból felépített NAND kapu igazságtáblázata

Eset	E_2	E_1	A	Z
1	0	0	0	?
2	0	1	0	
3	1	0	0	
4	1	1	1	

állapot ellentettje jelenik meg. Tehát a Z oszloban minden az A oszloban található érték ellentettjének kell szerepelnie.

Az 1. esetben $A = 0$, tehát ekkor $Z = 1$. A Z-nek a 2. 3. esetben szintén 1-nek kell lennie. A 4. esetben az $A = 1$. Ekkor $Z = 0$ -t kapunk (27.9. táblázat).

27.9. táblázat. A NAND kapu igazságtáblázata

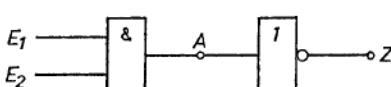
Eset	E_2	E_1	A	Z
1	0	0	0	1
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

Az eredő kapcsolat tehát az ÉS kapcsolat negáltja, amit ÉS-NEM kapcsolatnak nevezünk. Angolul a kapcsolatot NOT-AND-nek hívják. A két szó összevonásából alakult ki a rövidsége miatt nálunk is használt NAND kifejezés. A 27.15. ábra szerinti kapcsolás a NAND kapcsolat ÉS és NEM kapuból felépített változatát mutatja. minden áramkört, amely a 27.9. táblázat szerinti igazságtáblázatot teljesíti, NAND kapunak nevezzük. A 2 és 3 bemenetű NAND kapuk kapcsolási rajzát a 27.16. ábra adjá meg.

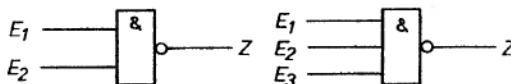
A NAND kapcsolat szintén kifejezhető matematikailag:

$$Z = \overline{E_1 \cdot E_2}$$

Az $E_1 \cdot E_2$ ÉS kapcsolat fölött húzott negációs vonal azt jelöli, hogy az egész ÉS kapcsolat ellentettjét kell venni.



27.15. ábra. Az ÉS kapu és a NEM kapu összekapcsolása



27.16. ábra. A 2 és 3 bemenetű NAND kapu rajzjele

A NAND kapu kimenetén akkor van 1 állapot, ha nem minden egyik bemenete 1 állapotú.

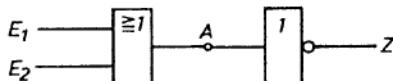
27.2.5. NOR (VAGY-NEM) kapcsolat

A VAGY kapu és a NEM kapu összekapcsolásával új logikai kapcsolat alakul ki (27.17. ábra). Az igazságátbla a NAND kapcsolatéhoz hasonló.

Az A kimeneten a VAGY kapcsolat eredménye adódik. Az A kimenet állapotát most negáljuk (ellentettjét képezzük). A 0-ból 1 lesz, az 1-ből 0. A Z oszlopban most az A oszlop negált állapotai állnak (27.10. táblázat). Az eredő kapcsolat a VAGY kapcsolat tagadása. Nevezhetnénk VAGY-NEM kapcsolatnak is. A gyakorlatban azonban az angol megfelelője (NOT-OR), ill. annak összevont alakja, a NOR használatos. A 27.17. ábra szerinti kapcsolás NOR kapcsolatot ad.

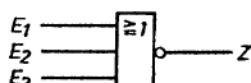
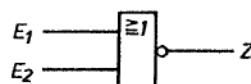
27.10. táblázat. A NOR kapu igazságátblázata

Eset	E ₂	E ₁	A	Z
1	0	0	0	1
2	0	1	1	0
3	1	0	1	0
4	1	1	1	0



27.17. ábra. A VAGY kapu és a NEM kapu összekapcsolása

Azokat az áramköröket, amelyek NOR kapcsolatot állítanak elő, vagyis teljesítik a 27.10. táblázat szerinti igazságátblázatot, NOR kapuknak nevezzük. A 27.18. ábra



27.18. ábra. A két- és hárombemenetű NOR kapuk rajzjele

2 és 3 bemenetű NOR kapuk szimbolikus rajzát mutatja. A NOR kapcsolat matematikai írásmódja a következő:

$$Z = \overline{E_1 + E_2}$$

A negációt jelölő felülvonás az egész $E_1 + E_2$ kifejezésre vonatkozik. Ezáltal az egész VAGY kapcsolat ellentettjét képezzük. ($Z = \overline{E_1} + \overline{E_2}$ ugyanis valami másat jelent.)

A NOR kapu kimenetén csak akkor jelenik meg 1, ha egyik bemeneten sincs 1 állapot.

28. Digitális kódok

28.1. Számjegyek és számok ábrázolása

Ha a digitális technikában számlálni vagy számolni akarunk, akkor meg kell találnunk a módját annak, hogy a számjegyeket és a számokat 2 jellel fejezzük ki. A digitális technika csak két állapotot ismer. Ezen állapotok mindenekéhez egy-egy jelet rendelünk hozzá. A számjegyek, ill. a számok ábrázolására a 0 és 1 jelet szokás használni.

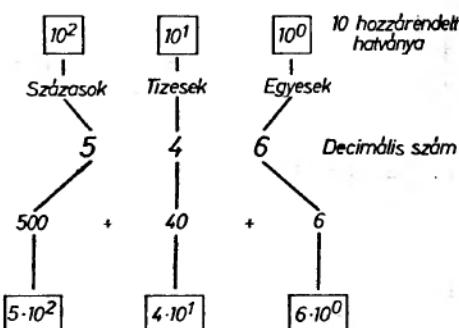
A két jellel való ábrázolást bináris ábrázolásnak nevezik.

A bináris ábrázolásra nagyon sok lehetőség van. A következőkben a legfontosabbakat tekintjük át.

28.1.1. Kettes számrendszer

A kettes számrendszer megértéséhez először a tízes számrendszert kell megvizsgálni. Válasszuk ki pl. az 546-ot! Kisiskolás korunk óta tudjuk, hogy az ez esetben — a helyi értéket tekintve — az 5 százasokat, a 4 tízeseket, a 6 pedig egyeseket jelent. Az 546 helyett tulajdonképpen $500 + 40 + 6$ -ot kellene írnunk. Tudjuk azonban, hogy az 5 jobbról a 3. helyen áll, akkor 500-at jelent. Ha az $500 + 40 + 6$ kifejezést a tíz hatványaival kifejezve írjuk fel (28.1. ábra), akkor megállapíthatjuk, hogy a számon belül minden egyes helyi értékhez a tíz valamelyik hatványa van hozzárendelve. Ez alapjában véve minden tízes számrendszerbeli számra igaz.

A decimális számon belül minden egyes helyi értékhez a tíz valamelyik hatványa van hozzárendelve.



28.1. ábra. A tízes számrendszer felépítése

Feladat

Írjuk fel a 25 648-at tíz hatványaival kifejezve!

Megoldás

$$2 \cdot 10^4 + 5 \cdot 10^3 + 6 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$$

Térünk most rá a kettes számrendszer felépítésére! A kettes számrendszer csak a 0 és az 1 számjegyeket „ismeri”. Ez tehát bináris (kétértékű) számrendszer.

A kettes számrendszer számjegyei: a 0 (nulla) és az 1.

Egy kettes számrendszerbeli szám tehát minden részben csak nullákból és egyesekből áll. Egy kettes számrendszerben felírt szám a következő alakú:

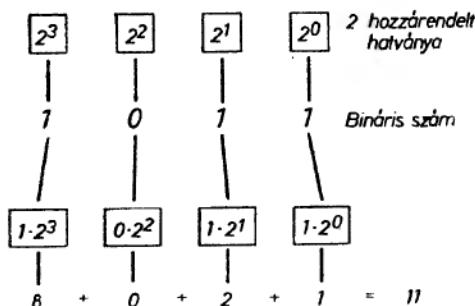
1011

Mekkora a szám decimális értéke?

Akárcsak a tízes számrendszerben, itt is jelentősége van annak, hogy a számon belül milyen számjegy melyik helyen áll, mivel az egyes helyekhez itt is hatvánnyal van hozzárendelve.

A kettes számrendszerbeli szám minden helyéhez a kettőnek valamelyen hatványa van hozzárendelve.

A 28.2. ábra mutatja a kettes számrendszer felépítését. Természetesen a kettes számrendszerbeli számok tízes számrendszerben is értelmesek. A bináris 110 (decimális hat) pl. összecserélhető a decimális 110-zel (száz tíz). Ezért a bináris számoknál minden meg kell adni, hogy kettes számrendszerbeli számról van szó.



28.2. ábra. A kettes számrendszer felépítése

Feladat

A bináris 11011-nek mekkora a decimális értéke?

Megoldás

$$\begin{array}{cccccc}
 & 1 & 1 & 0 & 1 & 1 \\
 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 & & & & & \\
 16 & 8 & 0 & 2 & 1 & = 27
 \end{array}$$

A bináris szám decimális értéke 27.

28.1.2. BCD kód (8-4-2-1 kód)

A BCD a *Binary Coded Decimal* angol kifejezés rövidítése, ami magyarul binárisan kódolt decimális számot jelent.

A BCD kód megnevezés nem egészen egyértelmű, mert többféle BCD kód lehetséges. Helyesebb a 8-4-2-1 kód elnevezés.

Ebben a rendszerben a decimális számokat 0-kkal és 1-esekkel, vagyis a kettes számrendszerrel ábrázoljuk. A legnagyobb értékű decimális számjegy a 9. Hogyan néz ki a 9 kettes számrendszerben?

Az ábrázoláshoz szükséges legmagasabb értékű kettes alapú hatvány 2^3 , ehhez összesen 4 jegy kell.

2^3	2^2	2^1	2^0	
8	4	2	1	1 bináris hely = 1 bit
1	0	0	1 = 9	

Mivel a legnagyobb decimális számjegy ábrázolásához 4 bináris jegy szükséges, minden decimális számjegyet egy-egy számjegynégyessel, ún. tetráddal fejezzük ki.

Egy bináris hely neve 1 bit (bit – *binary digit*). Egy decimális számjegy ábrázolására 4 bitet használunk. A BCD-kód (8-4-2-1 kód) tehát 4 bites kód.

A 28.1. táblázatban megadjuk a teljes BCD kódrendszerét. Meg kell most vizsgálni, hogy BCD kód segítségével mekkora decimális számok adhatók meg. A decimális számokat nagyon könnyű BCD kódban megadni. A decimális szám számjegyeit egyenként kell kódolni. Ezt egy példán keresztül mutatjuk meg.

28.1. táblázat. A BCD kód (8-4-2-1 kód)

Decimális szám	Bináris szám			
	2^3	2^2	2^1	2^0
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

A 375 három decimális számjegyből áll. Mindegyik számjegyet egy négyjegyű egységgel (tetráddal) adjuk meg.

3	7	5
0011	0111	0101

28.2. Egyéb bináris kódok

A két számjegyen alapuló kódok a bináris kódok. A decimális számok kifejezésére nagyon sokféle bináris kódot használnak.

A kódrendszer kialakításakor persze nem kell kötnünk magunkat a kettes számrendszer hatványkitevős rendszeréhez. Ez csak egy a sok hozzárendelési lehetőség közül, bár kétségtelenül gyakran alkalmazzák. Alapjában véve teljesen tetszőleges hozzárendelést választhatunk, és a decimális szám ábrázolására szolgáló bináris helyek számát is szabadon megválaszthatjuk. A 4 bites egységeken alapuló bináris kódrendsereket tetrádkódoknak nevezzük. A 28.2., 28.3. és 28.4. táblázatban három gyakran előforduló tetrádos kódrendszeret adtunk meg. Az 5 bites egységeken alapuló kódok lehetővé teszik a hibák felismerését. Pl. minden 5 bites egység lehet olyan, hogy az 1 kétszer, a 0 háromszor szerepeljen benne. Ilyen kódot ábrázol a 28.5. táblázat.

Ha hiba lép fel, azaz az egyik bináris hely megváltozik (pl. 0-ról 1-re, vagy megfordítva), akkor az 5 bites egységen már nem kétszer fordul elő az 1. Alkalmas digitális áramkör segítségével az 1 állapotok száma ellenőrizhető, és hibás szám esetén hiba-jelzés adható ki. Az információfeldolgozás megbízhatósága ezáltal jelentősen növelhető.

28.2. táblázat. 3-többletes kód

Decimális szám	Helyi érték			
	4	3	2	1
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

28.3. táblázat. Aiken-kód

Decimális szám	Helyi érték			
	4	3	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

28.4. táblázat. Gray-kód

Decimális szám	Helyi érték			
	4	3	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1

28.5. táblázat. 2 az 5-ből kód

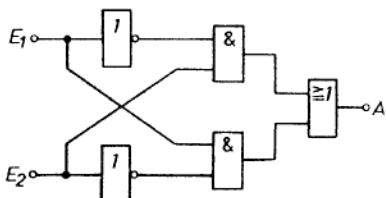
Decimális szám	Helyi érték				
	5	4	3	2	1
0	0	0	0	1	1
1	0	0	1	0	1
2	0	0	1	1	0
3	0	1	0	1	0
4	0	1	1	0	0
5	1	0	1	0	0
6	1	1	0	0	0
7	0	1	0	0	1
8	1	0	0	0	1
9	1	0	0	1	0

29. Kapcsolásanalízis

29.1. Általános ismeretek

A logikai kapukat egymagukban ritkán alkalmazzák. A digitális áramkör többnyire meglehetősen sok logikai kapuból áll, amelyek együttesen állítják elő a kívánt kapcsolatot. A gyakorlat számára tehát rendkívül fontos, hogy a logikai kapuk kapcsolásait analizálni tudjuk; vagyis meg kell tudnunk állapítani, hogy az áramkör egyes részei milyen kapcsolatot valósítanak meg, és milyen kapcsolatot állít elő a teljes áramkör. A kapcsolatok elemzését nevezzük kapcsolásanalízisnek.

Milyen kapcsolatot valósít meg pl. a 29.1. ábrán látható kapcsolás?



29.1. ábra. Logikai áramkör

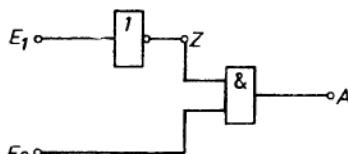
Először is azt a kérdést kell megválaszolni, hogy milyen kapcsolatot kell az áramkörnek előállítania. Azt a kapcsolatot, amelyet létre kell hozni, névleges kapcsolatnak nevezzük. A névleges kapcsolat rögzítéskor abból indulunk ki, hogy minden egyes elem kifogástalanul működik.

Azt a kapcsolatot, amit az áramkör ténylegesen megvalósít, ami tehát a tényleges állapotot adja meg, tényleges kapcsolatnak nevezzük.

A tényleges kapcsolat minden akkor tér el a névlegestől, ha egy vagy több kapcsolási elem nem működik tökéletesen. Ha az áramkör hibamentesen működik, akkor a névleges és tényleges kapcsolat megegyezik.

29.2. Névleges kapcsolat

A névleges kapcsolatot legegyszerűbben az igazságtablázattal határozzuk meg. Ezt egy példán keresztül mutatjuk be. A 29.2. ábrán két logikai elemből álló, egyszerű digitális kapcsolás látható. Mivel két bemenet van, az igazságtablázat 4 esetet tartalmaz (29.1. táblázat).



29.2. ábra. Logikai áramkör

29.1. táblázat. A 29.2. ábra szerinti kapcsolás igazságtáblázata

Eset	E_2	E_1	$Z = E_1$	A
1	0	0	1	0
2	0	1	0	0
3	1	0	1	1
4	1	1	0	0

A bemeneti állapotokat az előzőekben ismertetett séma szerint (l. a 27.2.1. pontot) írjuk be.

Először a NEM kapu Z kimenetének állapotait kell megállapítani. A NEM kapu kimenetén mindenig a bemeneten levővel ellentétes állapot van.

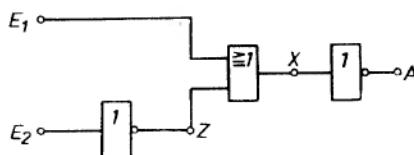
Az 1. esetben az E_1 állapota 0. A Z-nek tehát 1-nek kell lennie. A 2. esetben az E_1 1 állapotú, Z-nek tehát ebben az esetben 0-nak kell lennie. A 3. esetben Z ismét 1, a 4. esetben pedig 0.

Az ÉS kapcsolat most természetesen nem az E_1 és E_2 között jön létre, hanem a Z és E_2 között, mert az ÉS kapu egyik bemenetét itt most Z-nek nevezzük. Az ÉS kapcsolatra a 29.1. táblázatban a két aláhúzott oszlop a mérvadó. A kétbemenetű ÉS kapcsolat kimenetén akkor lesz 1 állapot, ha mindenki bemenetén 1 van.

Az 1. esetben az egyik bemenet (E_2) állapota 0 és a másik (Z) 1. A kimeneten tehát 0 állapotnak kell megjelennie. Csak a 3. esetben van mindenki bemenetén 1 állapot, így a kimeneten is csak ekkor van 1. A 2. és 3. esetben a kimenet állapota 0. Ezkel a kapcsolás névleges kapcsolatát meghatároztuk.

Feladat

Határozzuk meg a 29.3. ábra szerinti digitális kapcsolás névleges kapcsolatát az igazságtáblázat segítségével!



29.3. ábra. Logikai áramkör

Megoldás

Az igazságtáblázatban ismét 4 eset tüntethető fel, mivel a kapcsolásnak 2 bemenete van. A bemeneti állapotokat az ismert sémának megfelelően vesszük fel (29.2. táblázat).

29.2. táblázat. A 29.3. ábra szerinti kapcsolás igazságtáblázata

Eset	E_2	E_1	$Z = \bar{E}_2$	X	A
1	0	0	1	1	0
2	0	1	1	1	0
3	1	0	0	0	1
4	1	1	0	1	0

Most az E_2 állapotainak ellentettjét képezzük. Ahol E_2 nulla, a Z állapota 1, és megfordítva.

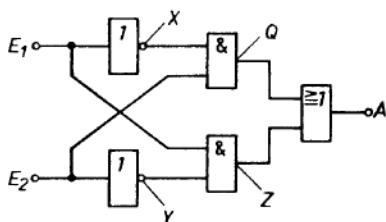
A VAGY kapcsolat az E_1 és Z (aláhúzott oszlopok) között hajtandó végre. A VAGY kapu kimenetén kapjuk az X értékeit. Ha a VAGY kapu valamelyik bemenetén 1 állapot van, akkor az X kimeneten is 1 van. Ez a helyzet érvényes az 1., 2. és 4. esetben. X azonban a NEM kapu bemenete. A NEM kapu kimenetén a bemeneti állapot negálta jelenik meg. Az A kimeneten tehát az X ellentettjének kell megjelennie. A 29.2. táblázat a névleges kapcsolat adatait mutatja. Érdekes, hogy a 29.2. és 29.3. ábra szerinti kapcsolások névleges kapcsolata azonos. Tehát a kívánt logikai kapcsolatot különböző kapcsolásokkal lehet megvalósítani.

Feladat

Állapítsuk meg a 29.4. ábrán látható digitális kapcsolás névleges kapcsolását az igazságtáblázat segítségével!

Megoldás

Az E_1 állapotát negálni kell, így az $X = \bar{E}_1$ -et kapjuk. Az E_2 állapotait is negálni kell. $Y = \bar{E}_2$ -t kapunk. Most képezzük a $Q = X \cdot E_2$ ÉS kapcsolatot, majd képuzzuk a $Z = E \cdot Y$ ÉS műveletet! A Q és Z oszlopok tartalma most VAGY kapura kerül, amelynek kimenete az A oszlopan van feltüntetve.



29.4. ábra. Logikai áramkör

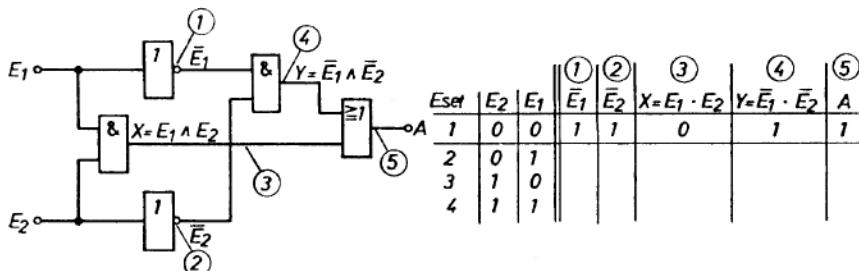
A 29.3. táblázat szerinti igazságtáblázat mutatja a névleges kapcsolatot. Ezt a kapcsolatot kizáró VAGY kapcsolatnak nevezzük. Az igazságtáblázaton alapuló analízis nagyobb digitális kapcsolások esetén is minden biztonsággal elvégezhető. Csak az a fontos, hogy nagy gondossággal, lépésről lépéstre haladjunk előre.

29.3. táblázat. A 29.4. ábra szerinti kapcsolás igazságáblázata

Eset	E_2	E_1	$X = \bar{E}_1$	$Y = \bar{E}_2$	$Q = X \cdot E_2$	$Z = E_1 \cdot Y$	A
1	0	0	1	1	0	0	0
2	0	1	0	1	0	1	1
3	1	0	1	0	1	0	1
4	1	1	0	0	0	0	0

29.3. Tényleges kapcsolat

A tényleges kapcsolatot többnyire méréssel állapítjuk meg, feltéve, hogy még nem tudjuk, melyik logikai elem hibás, és a hiba miben nyilvánul meg. Tekintsük a 29.5. ábrát! Ha meg kell határozni erre a kapcsolásra a tényleges kapcsolatot, akkor elkészítjük az ábra szerinti igazságáblázatot. Kapcsoljuk most rá a bemenetekre az 1. esetnek megfelelő logikai állapotot (vagyis mindenre a 0 állapotot, azaz 0 V-ot)! Mérjük meg a kapcsolás 1, 2, 3, 4 és 5 pontján a logikai állapotokat, és írjuk be az igazságáblázat megfelelő oszlopaiba! Ezután a 2. esetnek megfelelő állapotokat hozunk létre a bemeneten, és jegyezzük fel a vizsgált pontok logikai állapotát! Ugyanúgy járunk el a 3. és 4. esetben is! Ezzel megállapítottuk a tényleges kapcsolatot.



29.5. ábra. Logikai áramkör

A hibásan működő logikai elemeket a névleges és a tényleges kapcsolat egyszerű összehasonlításával szűrjük ki.

A hiba meghatározása a névleges és a tényleges kapcsolat összehasonlításával történik.

Ha a névleges és tényleges kapcsolat egyezik, akkor nincs hiba. Először az eredő kapcsolás kimeneti állapotait vetjük össze. Ha itt egyeznek az értékek, nem kell tovább vizsgálni, a kapcsolás rendben van. Ha a kimeneti állapotok nem egyeznek, a bemenettől kezdve lépésről lépésre kell az összehasonlítást elvégezni.

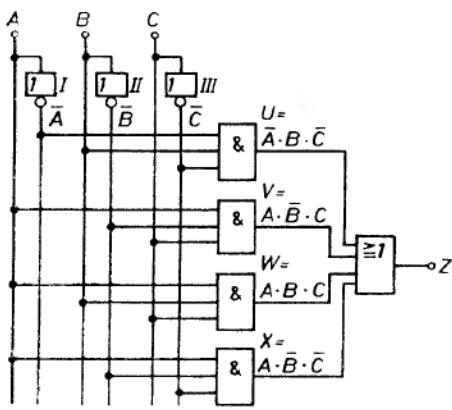
A 29.4. táblázat a 29.6. ábra szerinti kapcsolás igazságáblázatát és a mérés alapján összeállított igazságáblázatot adja meg. Mely kapuk működnek hibásan?

29.4. táblázat. A 29.6. ábra szerinti kapcsolás igazságtablázata (felül a névleges függvénykapcsolat, alul a tényleges függvénykapcsolat)

Eset	C	B	A	\bar{C}	B	\bar{A}	$U = \bar{A} \cdot B \cdot C$	$V = A \cdot \bar{B} \cdot C$	$W = A \cdot B \cdot \bar{C}$	$X = A \cdot \bar{B} \cdot \bar{C}$	Z
1	0	0	0	1	1	1	0	0	0	0	0
2	0	0	1	1	1	0	0	0	0	1	1
3	0	1	0	1	0	1	1	0	0	0	1
4	0	1	1	1	0	0	0	0	0	0	0
5	1	0	0	0	1	1	0	0	0	0	0
6	1	0	1	0	1	0	0	1	0	0	1
7	1	1	0	0	0	1	0	0	0	0	0
8	1	1	1	0	0	0	0	0	1	0	1

Eset	C	B	A	\bar{C}	\bar{B}	\bar{A}	$U = \bar{A} \cdot B \cdot \bar{C}$	$V = A \cdot \bar{B} \cdot \bar{C}$	$W = A \cdot B \cdot C$	$X = A \cdot \bar{B} \cdot C$	Z
1	0	0	0	1	1	1	0	0	0	0	0
2	0	0	1	1	1	0	0	0	0	1	1
3	0	1	0	1	1	1	1	0	0	0	1
4	0	1	1	1	1	0	0	0	1	1	1
5	1	0	0	0	1	1	0	0	0	0	0
6	1	0	1	0	1	0	0	1	0	0	1
7	1	1	0	0	1	1	0	0	0	0	0
8	1	1	1	0	1	0	0	1	1	0	1

Ha balról kezdve összehasonlítjuk az egyes oszlopokat, akkor a \bar{B} -nál hibát látunk. A NEM kapu, amelynek a \bar{B} -ot elő kellene állítania (II. elem), mindig 1 kimeneti állapotban van, tehát hibás. A II. elem hibája kihat a V és X kimenetekre, mert csak ezek a kapuk állnak kapcsolatban a \bar{B} -tal is. A mérési eredmények szerint a V és X értékei jók, figyelembe véve, hogy \bar{B} minden esetben 1. A V és X kapu tehát rendben van. További hiba mutatkozik a W-nél. A W kapu szintén hibás. A hibás kapcsolat nem vezethető vissza a II. kapu hibájára, mert a $W = A \cdot B \cdot \bar{C}$ nem tartalmazza a \bar{B} -ot. A II. és W kaput tehát ki kell cserélni.



29.6. ábra. Logikai áramkör

30. Logikai algebra

30.1. Alapfogalmak

A logikai kapcsolatokat a matematika speciális területe tárgyalja, amelyet logikai algebrának nevezünk.

A logikai algebra elméleti alapjait az ún. Boole-algebra alkotja (George Boole, 1815 – 1864), amely a halmazelmélet alapjául is szolgál. Emiatt a logikai algebrát Boole-algebrának is nevezik, bár a teljes Boole-algebrának ez csak egy részét teszi ki. A logikai algebra a normális algebrához hasonlóan változókat és állandókat ismer. Mindazonáltal csak két lehetséges állandó, nevezetesen a 0 és az 1 létezik benne. A változó vagy 0, vagy 1 értéket vehet fel.

A logikai algebra csak kétféle állandót ismer: a 0-t és 1-et.

Azok a mennyiségek, amelyek vagy a 0, vagy az 1 értéket vehetik fel, a változók. A logikai kapuk bemenetei és kimenetei tehát változók. A változókat többnyire nagybetűkkel jelöljük.

A logikai algebra változói olyan mennyiségek, amelyek a 0 és az 1 állapotokat vagy értékeket vehetik fel.

A változók között, ill. a változók és az állandók között összefüggéseket adhatunk meg logikai algebrai egyenletek formájában. A lehetséges műveletek: az ÉS kapcsolat, a VAGY kapcsolat és a negáció.

Minden logikai kapcsolat kifejezhető logikai algebrai egyenlet alakjában.

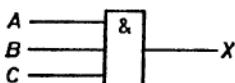
A következő műveleti jeleket szokás használni:

- az ÉS kapcsolat jele (használják a \wedge jelet is).
- + a VAGY kapcsolat jele (használják a \vee jelet is).

A negációt felülvonással jelöljük, pl. E_1 .

A 30.1. ábra szerinti ÉS kapcsolatra az alábbi egyenlet érvényes:

$$X = A \cdot B \cdot C.$$

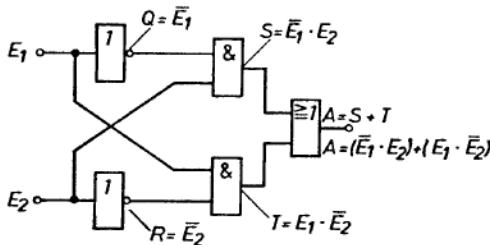


30.1. ábra. ÉS kapu

30.2. A kapcsolások logikai függvényének meghatározása

A kapcsolat, amelyet egy vagy több logikai kapuból álló áramkör létrehoz, logikai algebrai egyenettel is kifejezhető.

Ezt a 30.2. ábrán látható kapcsolás példáján keresztül mutatjuk be.



30.2. ábra. Összetett logikai hálózat

A változók: E_1 , E_2 , Q , R , S , T és A . Az E_1 és Q változó között a következő összefüggés áll fenn:

$$Q = \bar{E}_1.$$

Hasonló összefüggés áll fenn az E_2 és R változók között is:

$$R = \bar{E}_2.$$

Az S értéke egyenlő a Q és az E_2 között végrehajtott ÉS művelet eredményével:

$$S = Q \cdot E_2 = \bar{E}_1 \cdot E_2,$$

A T változóra hasonlóképp:

$$T = E_1 \cdot R = E_1 \cdot \bar{E}_2.$$

Az A értéke egyenlő az S és T közötti VAGY művelet eredményével:

$$A = S + T.$$

Ha az S -re és T -re kapott kifejezéseket behelyettesítjük, akkor

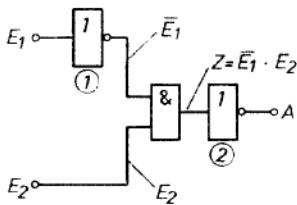
$$A = (\bar{E}_1 \cdot E_2) + (E_1 \cdot \bar{E}_2).$$

Ez az egyenlet a bemenet és a kimenet közötti kapcsolatot fejezi ki. Pontosan ugyanazt adja meg, mint az igazságtablázat.

Valamely kapcsolás logikai függvényét kifejező egyenletet függvényegyenletnek nevezik.

Feladat

Keressük meg azt a logikai algebrai egyenletet, amely megadja a 30.3. ábra szerinti kapcsolás bemenete és kimenete közti kapcsolatot!



30.3. ábra. Összetett logikai hálózat

Megoldás

Az 1. NEM kapu kimenetén E_1 van. Az ÉS kapu az \bar{E}_1 és E_2 közötti ÉS műveletet végzi el. Érvényes, hogy

$$Z = \bar{E}_1 \cdot E_2.$$

A 2. NEM kapu bemenetén a $Z = \bar{E}_1 \cdot E_2$ van. A kimeneten tehát a Z ellentettjének kell megjelennie.

$$A = \bar{Z} = \bar{\bar{E}}_1 \cdot \bar{E}_2.$$

30.3. A kapcsolás ábrázolása a függvényegyenlet alapján

Ha egy kapcsolat logikai algebrai egyenletét ismerjük, akkor ezen egyenlet alapján a kapcsolás felrajzolható.

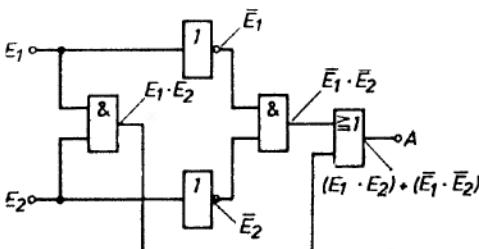
Példa

$$A = (E_1 \cdot E_2) + (\bar{E}_1 \cdot \bar{E}_2);$$

Megoldás

Először ÉS kapcsolatot kell képezni az E_1 és E_2 között. Ehhez egy ÉS kapura van szükség.

Az E_1 -et és E_2 -t negálni kell. Ez két NEM kapuval történik. Eredményül az \bar{E}_1 -et és \bar{E}_2 -t kapjuk. Az \bar{E}_1 és \bar{E}_2 ÉS kapcsolatát egy ÉS kapuval állítjuk elő. Utolsóként a két ÉS kapu kimenetét kell egy VAGY kapuval egyesíteni. A keresett kapcsolást a 30.4. ábra mutatja.



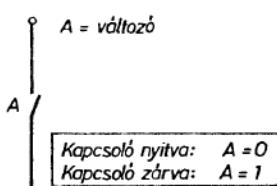
30.4. ábra. Összetett logikai hálózat

30.4. Függvényegyenlet és kapcsolóérintkező leírás

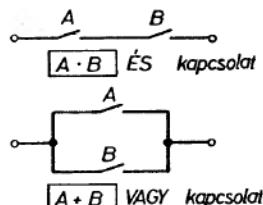
A logikai algebrában a változók kapcsoló (záróérintkező) segítségével is ábrázolhatók. A kapcsolónak két állapota van: nyitott és zárt.

A nyitott kapcsolóálláshoz a 0 állapotot, a zárt kapcsolóálláshoz az 1 állapotot rendeljük (30.5. ábra).

Két sorba kötött kapcsoló ÉS kapcsolatot, a párhuzamosan kapcsolt kapcsolók VAGY kapcsolatot reprezentálnak (30.6. ábra).



30.5. ábra. Változó ábrázolása kapcsolóval



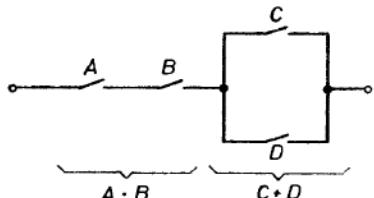
30.6. ábra. Érintkezőséma

Hogyan ábrázolható a változó ellentettje, pl. \bar{A} ? Az \bar{A} mindenkor ellentétes állapotú, mint az A . Ha az A -t most záróérintkezővel ábrázoljuk, akkor az \bar{A} -at bontóérintkezőként kell ábrázolnunk (30.7. ábra).

Az érintkezőséma tehát függvényegyenletet ábrázol. Milyen függvényegyenlet tartozik a 30.8. ábrán látható vázlathoz?



30.7. ábra. A negált változó ábrázolása bontóérintkezővel



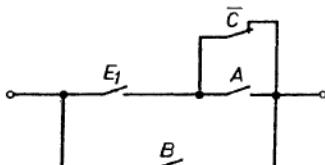
30.8. ábra. Érintkezőséma

Az A és B változók ÉS kapcsolatban vannak ($A \cdot B$). A C és D változók között VAGY kapcsolat ($C + D$) áll fenn. Az ($A \cdot B$) és ($C + D$) kapcsolat egymással ÉS kapcsolatban (soros kapcsolás) van. Végül a következő kifejezést kapjuk:

$$(A \cdot B) \cdot (C + D).$$

Feladat

Adjuk meg a 30.9. ábrán látható érintkezőséma függvényegyenletét!



30.9. ábra. Érintkezőséma

Megoldás

$$[(\bar{C} + A) \cdot E_1] + B.$$

30.5. A logikai algebra felhasználási lehetőségei

A logikai algebra segítségével logikai kapcsolatok ábrázolhatók, így a kapcsolásanalízist az igazságtablázat helyett a logikai algebra segítségével végezzük el.

A logikai algebra lehetővé teszi a kapcsolás átalakítását és egyszerűsítését. Adott kapcsolásra megadjuk a függvényegyenletet, és ez a függvényegyenlet meghatározott szabályok alapján matematikailag átalakítható és egyszerűsíthető. E könyv keretében a logikai algebra ezen szabályait nem vezethettük le, mert túl nagy terjedelmet igényelt volna. Az igazságtablázat logikai tartalma logikai algebrai egyenletekkel is leírható. Ezáltal lehetővé válik, hogy valamely digitális kapcsolást előre megadott feltételeknek megfelelően állítsunk elő (1. még a 31. fejezetet). Ez a digitális kapcsolások tervezése során igen fontos lehet.

A logikai algebra magas szintű ismerete inkább a tervezőmérnök számára fontos. A gyakorlati szakembernek elsősorban az itt közölt alapismeretek szükségesek.

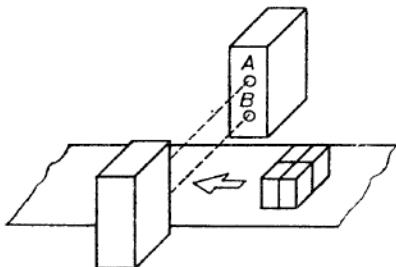
31. A logikai hálózatok szintézise

A hálózatszintézis célja az, hogy igényeinknek megfelelő digitális kapcsolást fejlesszünk ki. Pontosan adott, hogy ennek a digitális kapcsolásnak mit kell tudnia, azaz a követelmények ismertek. A szintézis összeállítást jelent. A feladat abból áll, hogy ismert kapcsolási elemeket úgy állítsunk össze, hogy eredményül a kívánt digitális kapcsolást kapjuk. Nézzük meg a kapcsolásszintézist egy konkrét példán!

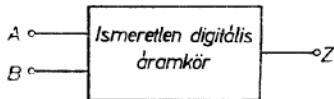
Egy csomagrendező berendezés két fotocellás fénysugárral működik (A és B), a 31.1. ábrának megfelelően. A csomagot ki kell dobni:

1. ha minden fotocella megszakítja a csomag (a csomag túl nagy),
2. ha az A fénysugár megszakad, a B viszont nem (torlódás, egymásra került csomagok),
3. egyik fénysugár sem szakad meg (túl kicsi a csomag).

A megszakított fénysugár jele: A, ill. B (mindkettő 0). A csomagkidobás jele: Z = 1. Keressük az A és B bemenetű, Z kimenetű kapcsolást, amely a felsorolt vezérlési feltételeket kielégíti (31.2. ábra).



31.1. ábra. Csomagrendező berendezés



31.2. ábra. Az A és B bemenettel és Z kimenettel rendelkező ismeretlen digitális áramkör

Hogyan jutunk most el a kívánt digitális kapcsoláshoz? A vezérlés feltételeit először igazságírásban kell kifejezni. Mivel a kívánt kapcsolásnak két bemenete van (A és B), ezért 4 eset létezik. A lehetséges bemeneti állapotokat az ismert séma szerint töltjük ki.

Mikor kell a Z-nek 1 állapotúnak lennie?

Az 1. feltétel (mindkét fénysugár megszakad $A = 0, B = 0$) az 1. esetnek felel meg. Itt $Z = 1$ -nek kell lennie. A 3. feltétel (egyik fénysugár sincs megszakítva, $A = 1, B = 1$) a 4. esetnek felel meg. Ekkor is $Z = 1$ kell, hogy legyen. Ha az A fénysugár megszakad, a B viszont nem, az azt jelenti, hogy az $A = 0, B = 1$, vagyis a 2. eset áll fenn.

Ezekben az esetekben ugyancsak $Z = 1$ érvényes. Z kívánt állapotait a 31.1. táblázatban tüntettük fel.

31.1. táblázat. A keresett vezérlő áramkör igazságtáblázata

Eset	A	B	Z
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

Milyen logikai kapcsolattal érjük el Z kívánt állapotait?

Most egy kicsit próbálkozni kell. Általában segít, ha a bemeneti változókat és negáltjaikat (A , B , \bar{A} és \bar{B}) egymással kapcsolatba hozzuk. A 31.2. táblázat a bemeneti változók és ellentettjeik igazságtáblázata. Ha a B -t és \bar{A} -at VAGY kapcsolatba hozzuk, akkor a Z értéke a kívánt állapotot veszi fel.

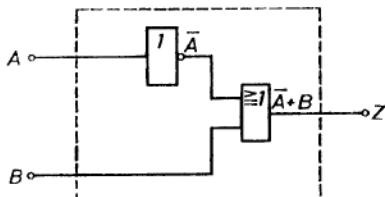
31.2. táblázat. A 31.3. ábra szerint megvalósított vezérlő áramkör igazságtáblázata

Eset	A	B	\bar{A}	\bar{B}	$Z = \bar{A} + B$
1	0	0	1	1	1
2	0	1	1	0	1
3	1	0	0	1	0
4	1	1	0	0	1

A logikai egyenlet tehát a következő:

$$Z = A + B.$$

A keresett vezérlőkapcsolást a 31.3. ábra mutatja.



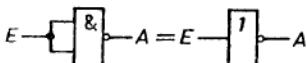
31.3. ábra. A keresett vezérlőkapcsolás

Feladat

Tételezzük fel, hogy csak NAND kapuk állnak rendelkezésünkre! Ezekkel a NAND kapukkal állítsuk elő az ÉS, a VAGY és a NEM kapcsolatot!

Megoldás

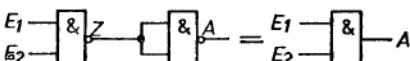
A legegyszerűbb a negációt megvalósítani. A NAND kapu két bemenetét összekötjük egymással a 31.4. ábrán bemutatott módon. A 31.3. igazságtablázat mutatja, hogy most csak az 1. és 4. esetek lehetségesek. Az ÉS kapcsolatot a NAND kapcsolat negációjával nyerjük úgy, hogy a NAND kapu után kapcsolunk egy NEM kapuként működő másik NAND kaput (31.5. ábra). Az igazságtablázatból kivehető, hogy tényleg ÉS kapu jön létre (31.4. táblázat).



31.4. ábra. Negáció megvalósítása NAND kapuval

31.3. táblázat. A NAND kapu igazságtablázata

Eset	E_2	E_1	$A = \overline{E_2 \cdot E_1}$
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0

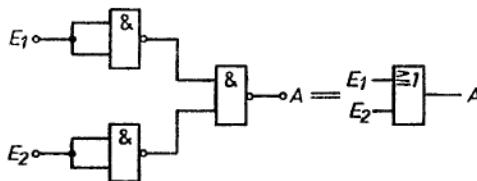


31.5. ábra. ÉS kapcsolat megvalósítása két NAND kapuval

31.4. táblázat. A 31.5. ábrán látható NAND kapus áramkör igazságtablázata

Eset	E_2	E_1	$Z = \overline{E_2 \cdot E_1}$	$A = Z$
1	0	0	1	0
2	0	1	1	0
3	1	0	1	0
4	1	1	0	1

Valamivel nehezebb a VAGY kaput előállítani. Az E_1 és E_2 bemeneti változókat először negálni kell. Az \bar{E}_1 és \bar{E}_2 között azután NAND műveletet végezünk (31.6. ábra). Az igazságtablázat (31.5. táblázat) alapján ez a kapcsolás VAGY kapcsolatot alkot.



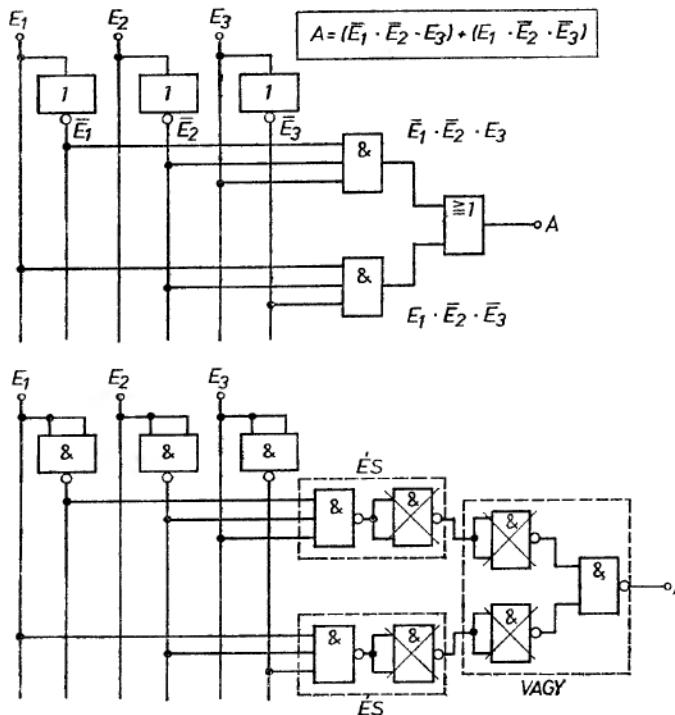
31.6. ábra. VAGY kapcsolat előállítása három NAND kapuval

31.5. táblázat. A NAND kapukkal előállított
VAGY kapcsolat
igazságitablázata

Eset	E_2	E_1	\bar{E}_2	\bar{E}_1	$E_2 \cdot E_1$	$\bar{E}_2 \cdot \bar{E}_1 = A$
1	0	0	1	1	1	0
2	0	1	1	0	0	1
3	1	0	0	1	0	1
4	1	1	0	0	0	1

A bemutatott megoldást többszöri próbálkozás után lehet megtalálni.

Ha bonyolult digitális kapcsolatot keresünk, nem sokra megyünk a próbálgatással. A bonyolult kapcsolásokat a logikai algebra és diagramok segítségével állíthatjuk elő.



31.7. ábra. A logikai hálózat és az alapelemek megfelelő NAND kapus helyettesítése

Így gazdasági szempontból optimális megoldáshoz is eljuthatunk, amelyben a legkisebb számú logikai kapu fordul elő.

A VAGY kapcsolat kialakításánál két NAND kapu szükséges a bemeneti változók negálásához, és további NAND kapu kell a kapcsolat létrehozásához (31.6. ábra). Ha tehát az ÉS kapcsolatok, a VAGY kapcsolatok és a NEM kapcsolatok NAND kapukkal megvalósíthatók, akkor tetszőleges kapcsolatot létrehozó kapcsolásokat is felépíthetünk kizárálag NAND kapukból.

Csak NAND kapuk felhasználásával minden kívánt kapcsolat létrehozható.

A NAND kapuk tehát — a NOR kapukhoz hasonlóan — univerzális kapukként alkalmazhatók.

Ha a digitális kapcsolásokat csak NAND kapukkal vagy csak NOR kapukkal akarjuk létrehozni, akkor az adott logikai algebrai egyenleteket sok esetben megfelelően át kell alakítanunk. Ezek az átalakítások különböző módokon hajthatók végre. Hogy ezt hogyan kell elvégezni, azt a 31.7. ábrán mutatjuk meg.

Az ábra felső részén az $A = (\bar{E}_1 \cdot \bar{E}_2 \cdot E_3) + (E_1 \cdot \bar{E}_2 \cdot \bar{E}_3)$ egyenletnek megfelelő kapcsolást alapelemekkel ábrázoltuk. Ezt a kapcsolást NAND kapukkal kell megvalósítani. minden egyes alapkaput a neki megfelelő NAND kapcsolással helyettesítünk. Az egymás után következő NEM kapcsolatokat elhagyhatjuk.

Ezt az eljárást minden alkalmazhatjuk.

32. Áramkörcsaládok

32.1. Relés kapcsolások

Relék segítségével minden logikai kapcsolat előállítható, a relék viszont lassan kapcsolnak, nagy a helyigényük, viszonylag nagy elektromos teljesítményt vesznek fel és aránylag könnyen meghibásodnak. Ezért ma már csak viszonylag ritkán alkalmaznak relés logikai áramköröket. Korábban főként az ilyen típusú kapukat használták, és az első számítógépek egyike kizárolag relés elemekkel működött. Az erősáramú technikában a logikai kapcsolatokat védelemmel látták el. Az ilyen kapcsolásokban relés logikai áramköröket alkalmaznak.

32.2. DTL áramkörök

A DTL a *dióda-tranzisztor-logika* kifejezés rövidítése. A logikai kapukat félvezetős kapcsolásként építik fel, túlnyomórészt diódákból és bipoláris tranzisztorokból. Az áramkörök egyedi alkatrészektől (diszkrét elemektől) épülnek fel, és nem integrált áramkör formájában.

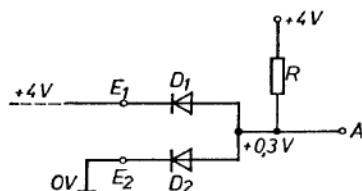
A diódákon és tranzisztorokon kívül többnyire még ellenállásokat is alkalmaznak. A 32.1. ábra egy ÉS kapu áramköri kialakítását mutatja. Érvényes az eddig alkalmazott megállapodás:

- logikai $0 \rightarrow 0$ V (föld);
- logikai $1 \rightarrow +5$ V.

Csak ha minden bemenetben 1 (+5 V) állapot van, akkor lehet az A kimeneten 1 állapot.

Ha bármelyik bemenetben 0 (0 V) állapot van, akkor a kimeneti feszültség kb. 0 V-ra csökken.

A 32.1. ábrán pl. $E_1 = 1$ és $E_2 = 0$.

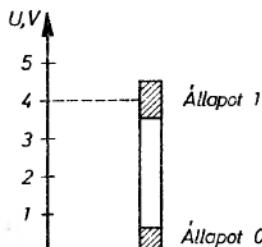


32.1. ábra. Diódás ÉS kapu

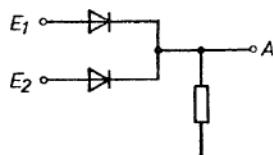
A kimeneti feszültségéről nem pontosan, hanem csak megközelítőleg 0 V. Az eltérés tulajdonképpen a dióda küszöbfeszültsége. Az 1 állapotban sem lesz a kimeneti feszültség pontosan +5 V, mivel a következő kapu vezérléséhez megfelelő áram szükséges.

Ha a kimeneten áram van, akkor az R ellenálláson feszültségesés lép fel, így a kimeneti feszültség csökken. Emiatt tehát a feszültségértékeket, amelyekhez a logikai állapotokat hozzárendelik, megfelelő tűréssel kell értelmezniünk. A 32.2. ábra egy ilyen tűréstartományt személlet. A 0 és 0,5 V közé eső feszültségek tartoznak a logikai 0 állapothoz, a +4,5 és 5,5 V közé eső feszültségek pedig a logikai 1 állapothoz.

A VAGY kapu működését (32.3. ábra) viszonylag könnyű megmagyarázni. Ha az E_1 vagy E_2 bemeneten (vagy minden bemenetén) 1 állapot ($\approx +5$ V) van, akkor a dióda vezetővé válik, és a kimeneten is 1 állapot lesz.



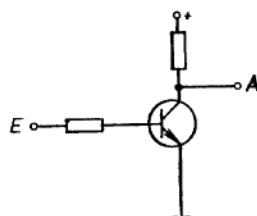
32.2. ábra. A logikai szintek tűrési sávjai



32.3. ábra. DTL áramkörös VAGY kapu

A NEM kaput tranzisztoros fokozatként alakítjuk ki (32.4. ábra). Ha az E bemeneten 0 állapot van (≈ 0 V), akkor a tranzisztor zárva van. A kimeneten kb. +5 V feszültség, tehát 1 állapot van. Ha most a bemenetre az 1 állapotot kapcsoljuk, akkor a tranzisztor nyit, és a kimeneten kb. 0,3 V feszültség lesz, tehát hozzávetőleg 0 V. Ez a 0 állapotnak felel meg.

Az áramkör kimenetén tehát mindenkorában a bemenettel ellentétes állapot lép fel.



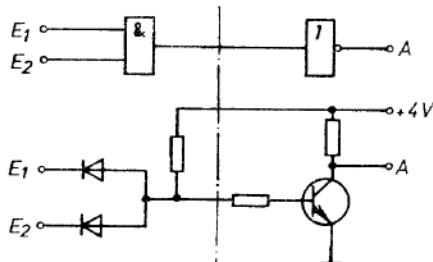
32.4. ábra. DTL áramkörös NEM kapu

Feladat

Állítsuk elő a NAND kapu DTL áramkörös változatát!

Megoldás

A keresett kapcsolás egy ÉS kapu és egy NEM kapu összekapcsolásából adódik (32.5. ábra).

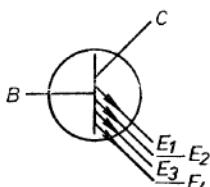


32.5. ábra. DTL áramkörös NAND kapu felépítése

32.3. TTL áramkörök

A TTL a *tranzisztor-tranzisztor-logika* elnevezés rövidítése. A logikai kapukat főleg a tranzisztorokkal építik fel. Kiviteli formájuk kizárolag integrált áramköri (IC). A TTL IC-kben a tranzisztorokat bipoláris tranzisztorokként – tehát nem FET-ként – állítják elő.

A leggyakrabban alkalmazott TTL kapuáramkörök speciális jellemzője a többemitteres tranzisztor (32.6. ábra).

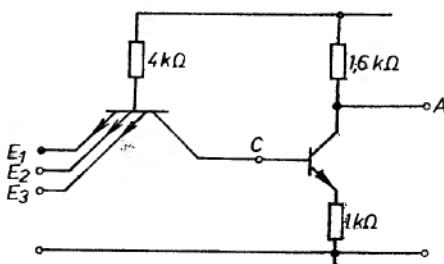


32.6. ábra. Többemitteres tranzisztor

A TTL áramkörök alapeleme a NAND kapu, amely viszonylag egyszerűen és nagyon jó minőséggel gyártható. Az ÉS, VAGY és NEM kapukat mint logikai alapelemeket NAND kapukból is fel lehet építeni. Ez viszont azt jelenti, hogy NAND kapukkal minden elképzelhető kapcsolat előállítható.

A gyártók katalógusaiba vetett egyetlen pillantás elárulja azonban, hogy a NAND kapun kívül ÉS, VAGY, NEM és NOR kapukat is kínálnak.

Az integrált áramkörök gyakran igen bonyolultak, úgyhogy a belső működésmódjukat többnyire nem könnyű megérteni. Ennek ellenére egy jellemző példán keresztül megpróbáljuk a TTL áramkör (32.7. ábra) működését megmagyarázni.

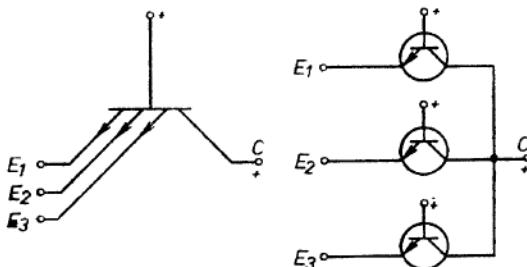


32.7. ábra. TTL áramkörös NAND kapu kapcsolása

Valamely többemitteres tranzisztor – pl. egy 3 emitterű tranzisztor – úgy működik, mint 3 párhuzamosan kapcsolt tranzisztor (32.8. ábra).

Ha az egyik bemeneten 0 állapot van (vagyis 0 V), akkor a hozzá tartozó tranzisztor kis ellenállású (vezet).

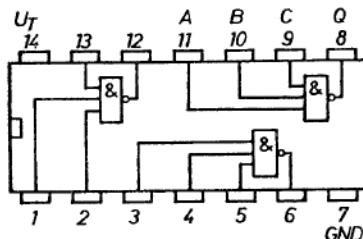
A feszültségszint a C pontban tehát csökken. Csak akkor nem csökken a C pont feszültsége, ha minden bemenet állapota 1 (tehát +5 V). Azok a tranzisztorok,



32.8. ábra. Többemitteres tranzisztor három párhuzamosan kapcsolt tranzisztorról ábrázolva

amelyeknek a bemenetén +5 V van, inverz üzemmódban működnek. (A kollektor és az emitter szerepét cserél.) A többemitteres tranzisztor tehát ÉS kapcsolatot hoz létre. Ha a C pontban pozitív feszültségszintet alakít ki, akkor elláthatja a következő tranzisztoros fokozat vezérlését. Ez a fokozat NEM kapuként működik, így a 32.7. ábra szerinti kapcsolás összességében NAND kapcsolatot eredményez.

A TTL áramköröket többnyire mint dual-in-line tokban elhelyezett integrált áramköröket kínálják. A 32.9. ábra egy integrált áramkör lábkiosztását ábrázolja. Az áramkör három hárombemenetű NAND kaput tartalmaz.



32.9. ábra. Integrált TTL áramkör három hárombemenetű NAND kapuval

32.4. MOS áramkörök

Egészen különleges technológiát alkalmaznak a MOS áramkörök esetében. A logikai elemeket MOS térvízelésű tranzisztorokból, integrált áramkörként alakítják ki. Ezzel az eljárással igen nagy alkatrészsűrűségű integrált áramkörök állíthatók elő.

Pl. egyetlen áramköri lapon egy egész számítógép összes áramköre létrehozható. A MOS áramkörök teljesítményfelvétele csekély, a vezérlés gyakorlatilag teljesítmény nélkül megvalósítható, mivel mint ismeretes, a MOSFET tranzisztoroknak extrém nagy a bemeneti ellenállásuk.

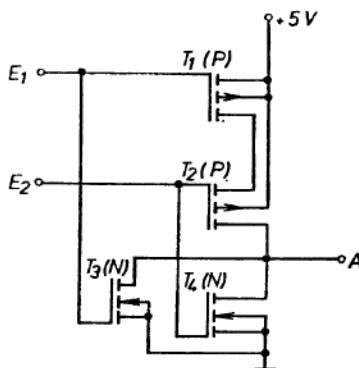
A MOS áramkörök igen gazdaságosak, vagyis viszonylag olcsón előállíthatók.

A 32.10. ábrán ún. CMOS technológiával készült NOR kapu áramköre látható. A CMOS technológia olyan MOS technológia, amely az áramkörökhez egyaránt használ p és n csatornás MOSFET-eket. Ez tehát szimmetrikus komplementer technológiá.

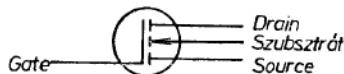
A COS/MOS technika vagy CMOS technológia szimmetrikus komplementer, szimmetrikus fém – oxid – félvezető-technológiát jelent.

A 32.10. ábra szerinti kapcsolás elemzéséhez ismerni kell a MOSFET működési elvét. A T_1 és T_2 tranzisztor önzáró, p csatornás MOSFET. A tranzisztor rajzjele és a csatlakozások megnevezése a 32.11. ábrán látható.

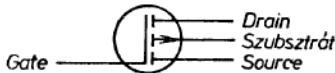
A drain – source-szakasz igen kis ellenállású lesz, ha a kapuelektródára a szubsztrát-hoz, ill. a source-höz képest elegendően nagy negatív feszültséget kapcsolunk.



32.10. ábra. CMOS áramkörös NOR kapu



32.11. ábra. Önzáró, p csatornás MOSFET rajzjele és csatlakozásainak elnevezése



32.12. ábra. Önzáró, n csatornás MOSFET rajzjele és csatlakozásainak elnevezése

A T_3 és T_4 tranzisztor önzáró, n csatornás MOSFET. Rajzjelét és a csatlakozások megnevezését a 32.12. ábra mutatja. Az n csatornás MOSFET drain – source-szakasza akkor lesz kis ellenállású, ha a kapura a szubsztráthoz, ill. a source-höz képest elegendően nagy pozitív feszültséget kapcsolunk.

Ha a 32.10. ábra szerinti kapcsolásban az E_1 bemeneten 1 állapotot (vagyis +4 V) van, akkor T_3 nyit, a kimenet 0 állapotba megy át (feszültségszintje ≈ 0 V). Ha az E_2 bemenet állapota 1, akkor az A kimenet ugyancsak a 0 (vagyis 0 V) állapotot veszi fel. Csak akkor zár T_3 és T_4 , ha minden bemenet állapota 0.

A T_1 és T_2 tranzisztor ennek ellenére nyitva van, mivel a 0 V számukra negatív kapufeszültséget jelent (a source-höz, ill. a szubsztráthoz képest). Vegyük figyelembe,

hogy a T_1 source-pontja és minden két szubsztrát +5 V feszültségen van! A kimenet most 1 állapotba kerül. Ezzel az áramkörre a 32.1. táblázat szerinti igazságigénylátható adódik, ami a NOR kapcsolat igazságigényláthatója. A MOS technológiájú áramkörököt minden szélesebb körben alkalmazzák a digitális vezérléstechnikában és a számítógépekben. Mivel kis helyen sok FET elfér, ezért nemcsak a logikai kapukat, hanem teljes funkcionális egységeket állítanak elő integrált áramkörként.

Ma már a zsebszámolók teljes aritmetikai és vezérlőrendszerét egyetlen MOS integrált áramkörként építik fel.

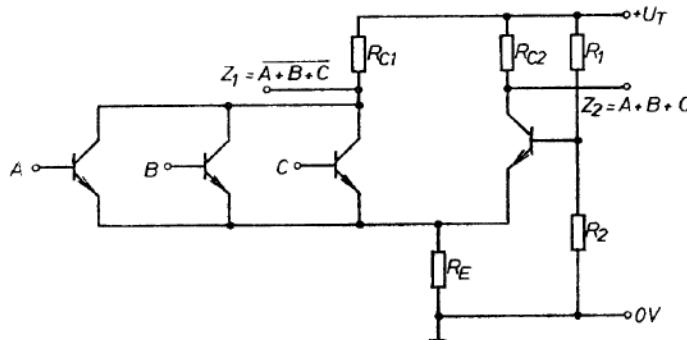
32.1. táblázat. A CMOS NOR kapu igazságigényláthatója

Eset	E_2	E_1	A
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	0

32.5. ECL áramkörök

Az ECL az angol Emitter Coupled Logic kifejezés rövidítése, ami emittercsatolt logikát jelent. Az ECL áramköröket bipoláris tranzisztorokból felépített integrált áramkörként állítják elő (32.13. ábra).

Az ECL áramkörök kifejlesztésének célja egy lehetőleg gyors, nagyon rövid kapcsolási idejű áramkörcsalád létrehozása volt. A rövid kapcsolási idő azonban csak úgy érhető el, ha a tranzisztorok nyitóirányban nincsenek teljes telítésig túlvezérelve.



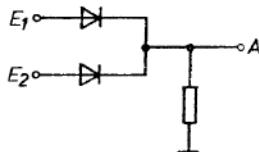
32.13. ábra. ECL kapu kapcsolása

32.6. Az alacsony (Low) és magas (High) logikai szintek

Egy dolgot világosan kell látni: az áramkörök, amelyek relékből, diódákból, bipoláris tranzisztorokból vagy FET-ekből épülnek fel, nem „érlik” a digitális logikát. Feszültségekre és áramerősségekre reagálnak.

Ezen a gondolaton alapul az az ötlet, hogy a digitális kapcsolások működését elektromos úton, azaz bármiféle logikai hozzárendelés nélkül írjuk le. Ennek egyszerű módja az, hogy az igazságátblázathoz hasonló táblázatot készítünk, és abba a 0 és 1 logikai állapotok helyett a tényleges feszültségeket írjuk be.

Végezzük el ezt a 32.14. ábra szerinti kapcsolásra! Eredményül a 32.2. táblázatot kapjuk. Az ilyen táblázatot működési tábláznak nevezzük. Csak az elektromos viszonyokat írja le, és csak a 0 V és +4 V feszültségértékekre adja meg az adatokat.



32.14. ábra. Egyszerű diódás kapuáramkör

Ha 0 V és +6 V feszültséget alkalmaznánk, akkor a táblázatnak más értéket kellene tartalmaznia. Megint más lenne a táblázat tartalma, ha 0 V és +8 V feszültséget használnánk (32.3. táblázat).

32.2. táblázat. Működési táblázat feszültségértékekkel

Eset	E_2 , V	E_1 , V	A, V
1	0	0	0
2	0	4	4
3	4	0	4
4	4	4	4

32.3. táblázat. Működési táblázat más feszültségértékekkel

Eset	E_2 , V	E_1 , V	A, V
1	0	0	0
2	0	8	8
3	8	0	8
4	8	8	8

Ha azonban összehasonlítjuk a 32.2. és 32.3. táblázatokat, megállapíthatjuk, hogy a magasabb, ill. az alacsonyabb feszültségszintek minden ugyanazonon a helyeken szerepelnek.

Ha a magasabb szintértéket H-val jelöljük (az angol High – magas alapján), az alacsonyat pedig L-lel (Low – alacsony), akkor olyan működési táblázatot kapunk, amelyik bármelyik konkrét jelszintre érvényes (32.4. táblázat).

32.4. táblázat. Működési táblázat

Eset	E_2	E_1	A
1	L	L	L
2	L	H	H
3	H	L	H
4	H	H	H

A következő megállapodás érvényes:

L — Low — alacsony szint.

A mínusz végtelenhez ($-\infty$) közelebb eső szint.

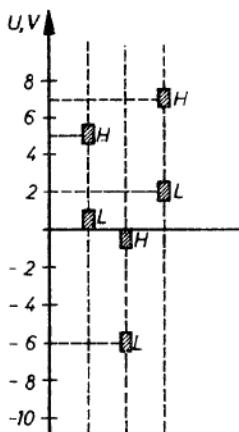
H — High — magas szint.

A plusz végtelenhez ($+\infty$) közelebb eső szint.

A digitális áramkörök különböző feszültségszinteken működtethetők.

Hogy a H és az L érték mekkora feszültségnek felel meg, arra nézve a 32.15. ábra három különböző esetet mutat be.

Igen gondosan kell ügyelni arra, hogy az L és H szinteket sohase cseréljük fel a logikai állapotokkal!



32.15. ábra. A logikai szintek lehetséges értékei

32.7. Pozitív és negatív logika

Az L és a H bináris szintek a 0 és az 1 logikai állapotokhoz kétféleképpen rendelhetők hozzá:

$L \rightarrow 0$	$L \rightarrow 1$
$H \rightarrow 1$	$H \rightarrow 0$

(pozitív logika) (negatív logika)

Pozitív logikáról beszélünk, ha az alacsony jelszint a 0 állapothoz, a magas jelszint pedig az 1 állapothoz tartozik.

A digitális technikában ma túlnyomó többségben pozitív logikával dolgoznak. Ha valamely áramkörről semmi közelebbet sem tudunk, abból indulhatunk ki, hogy a pozitív logika érvényes.

A negatív logikában az alacsony szint az 1, a magas szint a 0 állapothoz van hozzárendelve.

A negatív logikának akkor volt nagyobb jelentősége, amikor még csupán pnp tranzisztorok álltak rendelkezésre. Negatív U_{CE} feszültségek esetén a tranzisztoros kapcsolófokozatok kimenetein negatív feszültségek adódnak.

Példa

$$0 \rightarrow -0,3 \text{ V} = \text{H}$$

$$1 \rightarrow -6 \text{ V} = \text{L}$$

Napjainkban a negatív logikát egyes vezérlőkapcsolásokban — zavarvédelmi okokból — alkalmazzák.

Milyen kapcsolatot valósít meg a 32.14. ábra szerinti kapcsolás pozitív logika, ill. milyent negatív logika esetén? A megfelelő működési adtokat a 32.4. táblázat szemlélteti. A működési táblázatból az igazságátáblázat származtattható. Pozitív logika esetén a H helyébe a logikai 1, az L helyébe a logikai 0 állapotot kell beírni (32.5. táblázat). Pozitív logika esetén az áramkör VAGY kapcsolatot állít elő.

32.5. táblázat. A 32.14. ábra kapuáramkörének igazságátáblázata pozitív logika esetén (VAGY kapu)

Eset	E_2	E_1	A
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

Negatív logika esetén az L-ből logikai 1, a H-ból logikai 0 állapot lesz (32.6. táblázat). A kapcsolás ÉS kapcsolatot állít elő.

Az igazságátáblázatban az esetek sorrendje sajnos egy kicsit más.

Pozitív logikáról negatív logikára való áttérés esetén, vagy megfordítva, a kapcsolás megváltoztatja logikai függvényét.

32.6. táblázat. A 32.14. ábra kapuáramkörének
igazságtablázata negativ logika
esetén (ÉS kapu)

Eset	E_2	E_1	A
1	1	1	1
2	1	0	0
3	0	1	0
4	0	0	0

33. Flip-flopok

33.1. A flip-flopok tulajdonságai

Minden elektronikai kapcsolás, amelynek két stabil állapota van és amely megfelelő bemenőjel segítségével egyik állapotból a másikba átkapcsolható, flip-flopként, azaz bistabil billenőfokozatként működik.

Igen sok olyan áramkör van, amely az iménti megállapítás szerint bistabil billenőfokozatnak számít. Az egyes kapcsolások különbözőképpen vezérelhetők és eltérően működő bemenetekkel rendelkezhetnek. Néhány kapcsolás csak meghatározott feltételek mellett billen át, pl. különleges ütem- vagy parancsjelek egyidejű jelenléte esetén.

A legtöbb flip-flopnak rögzített alapállapota van, vagyis a feszültség bekapcsolásakor ezt az alapállapotát veszi fel. Gyakran külön bemenet, az ún. törlőbemenet (reset) segítségével kapcsolható vissza az alapállapotba az áramkör.

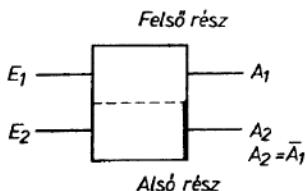
A flip-flopokat ma többnyire integrált áramkör formájában állítják elő. Az ilyen IC belső felépítése nem túlságosan érdekel bennünket, mivel a hibás integrált áramkör megjavítására úgysincs lehetőségünk. Ezért sokkal inkább érdekes lenne azt megtudnunk, hogy az áramkör egészében mint „fekete doboz”, hogyan működik. A következő vizsgálódások ezért nem a belső felépítésre irányulnak – mint ahogy azt a 19. fejezetben tettük –, hanem a teljes kapcsolás viselkedésére vonatkoznak. A „fekete doboz” szemléletmód miatt a rajzokon nem kell a flip-flopokat tranzisztorrendszerek, diódák, ellenállások segítségével ábrázolni.

Az egész flip-flop egyetlen kapcsolási jellel megadható.

Nézzünk meg először egy egyszerű flip-flopot, amelynek E_1 , E_2 bemenete és A_1 , A_2 kimenete van! Az áramkör rajzjelét a 33.1. ábra mutatja.

A tápfeszültség csatlakozásait a rajzjelen nem tüntetik fel.

Az áramkör működését az L és a H feszültségszintekkel írhatjuk le. Gyakoribb azonban, hogy a feszültségszintekhez hozzárendelik a 0 és az 1 logikai állapotokat, és a működést a logikai állapotokkal ábrázolják. A következőkben is ezt fogjuk tenni. A 33.1. ábrán látható flip-flopnak A_1 és A_2 kimenete van. Ha A_1 1 állapotú, akkor A_2 -nek 0 állapotban kell lennie és megfordítva. Ez általánosságban érvényes minden flip-flopra. Ez a törvényszerűség a következő logikai algebrai egyenlettel fejezhető ki:



33.1. ábra. Egyszerű flip-flop

$$A_2 = \bar{A}_1.$$

A tápfeszültség bekapcsolása után a flip-flop alaphelyzetbe kerül, amit nyugalmi állapotnak is nevezünk. Azt a bemenetet, amely nyugalmi állapotban 1 értékű, fekete vastagítással jelöltük. Mivel ez gyakorlatilag mindenkor az A_2 kimenet, ez a jelölés el is hagyható, ha nem okoz tévedést.

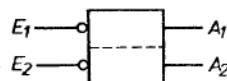
$$\text{Nyugalmi állapot: } A_1 = 0, \quad A_2 = 1.$$

$$\text{Működési állapot: } A_1 = 1, \quad A_2 = 0.$$

A következő kikötések érvényesek:

1. Az E_1 bemeneten levő 1 állapotot a flip-flopot $A_1 = 1$ állapotba kapcsolja. Ezt az eseményt beállításnak nevezzük. Ha a flip-flop már $A_1 = 1$ állapotban van, akkor a bemenetre adott 1 állapot hatástalan. A flip-flop ilyenkor nem kapcsol át.
2. Az E_2 bemenet 1 állapota a flip-flopot $A_2 = 1$ állapotba kapcsolja. Ezt az eseményt visszaállításnak nevezzük. Ha a flip-flop már $A_2 = 1$ állapotban van, akkor az E_2 bemenetre adott 1 állapot nem befolyásolja a flip-flop állapotát.
3. A 0 állapotoknak semmiféle vezérlő hatása nincs.
4. Az A_1 kimenet állapota megadja a flip-flop tárolási állapotát.

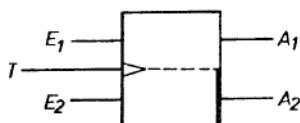
Ha $A_1 = 1$, akkor a flip-flop az 1 értéket tárolja. Magától értetődő, hogy olyan flip-flopok is készíthetők, amelyeket a 0 állapottal lehet vezérelni. Ezeknek a flip-flopoknak különleges, negált bemeneti vannak (33.2. ábra), és csak ritkán alkalmazzák őket.



33.2. ábra. 0 állapottal vezérelt flip-flop rajzjele

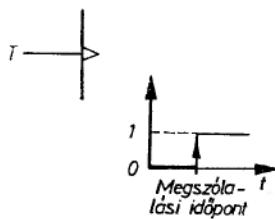
A 33.3. ábra szerinti flip-flopnak az E_1 és E_2 bemeneten kívül még egy T órabemenete is van. Az ilyen ütemezőbemenettel nagyobb áramkörök esetén több flip-flop egyidejű kapcsolása váltható ki.

Az órabemenetek a bemeneti állapotok változásaival együtt hatnak, ezért dinamikus bemeneteknek nevezik őket.



33.3. ábra. Órabemenettel rendelkező flip-flop (kapcsolási időpont: felfutó élnél)

Kétfélé dinamikus bemenet van. Az egyik akkor hatásos, ha a bemeneti állapot 0-ról 1-re változik. Az ilyen bemenetet a felfutó élhez tartozó dinamikus bemenetnek (33.4. ábra) nevezzük. A másik fajta dinamikus bemenet akkor hatásos, amikor a bemeneti



33.4. ábra. Felfutó ére reagáló dinamikus bemenet ábrázolása

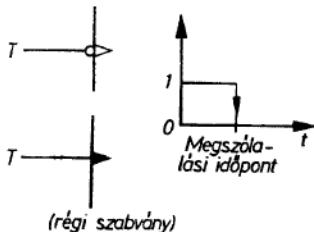
állapot 1-ről 0-ra változik. Ezt a lefutó élhez tartozó dinamikus bemenetnek nevezzük (33.5. ábra).

A 33.3. ábrán látható flip-flop órabemenete felsutó élnél hatásos. A dinamikus bemenetre érvényes a következő:

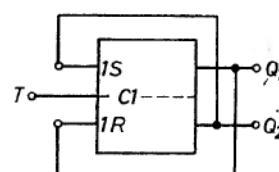
Ha az E_1 -en 1 van, az órajel felfutó éle az A_1 -et 1-be állítja. Ez a beállítás művelete (működési állapotba kapcsolás).

Ha az E_2 -n 1 állapot van, az órajel felfutó éle az A_2 -t 1-re állítja, vagyis az A_1 -et 0-ra. Ez a visszaállítás művelete (nyugalmi állapotba való kapcsolás).

A 33.6. ábra szerinti flip-flop az órajel lefutó élének hatására kapcsol. Egyébként úgy működik, mint a 33.3. ábrán bemutatott flip-flop. A sok lehetséges flip-flop típusból csak háromnak van különleges jelentősége. Ezek az SR flip-flop, a T flip-flop és a JK flip-flop.



33.5. ábra. Lefutó ére reagáló dinamikus bemenet ábrázolása



33.6. ábra. Flip-flop lefutó ére reagáló órabemenettel

33.2. SR flip-flopok

Az S és az R a beállítás (set) és a visszaállítás (reset) műveletét jelenti.

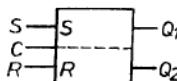
Az SR flip-flop rajzjelét a 33.7. ábra mutatja. A beállításra szolgáló bemenetet S betűvel jelöljük, a visszaállítás bemenetét pedig R-rel. Az órabemenet jele T vagy C (clock). A kimeneteket többnyire Q_1 és Q_2 jelöli.

Az újabb szabványok az óraütemet számjegyekkel jelölik. Az 1. ütemhez tartozó bemenetet C1-nek nevezzük.

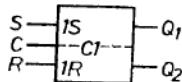
Nagyobb áramkörök esetén szükség lehet a 2. ütemre is, amelyet C2-nek nevezünk.

Az ütembemenetekhez tartozó S és R bemeneteket a betűjel elő helyezett számjeggyel azonosítják. Az 1S bemenet az 1. óraütemhez tartozó beállítóbemenet. Az 1R bemenet az 1. ütemhez tartozó visszaállító bemenet (33.8. ábra).

Az SR flip-flop működése a 33.1. igazságtablázatból olvasható le. Az igazságtablázat t_n és t_{n+1} jelölésű oszlopokat tartalmaz.



33.7. ábra. SR flip-flop rajzjele



33.8. ábra. Ütemfüggő SR flip-flop rajzjele

33.1. táblázat. Az SR flip-flop igazságtablázata

Eset	t_n		t_{n+1} Q_1	Funkció
	S	R		
1	0	0	Q_{1n}	tárolás
2	0	1	1	beállítás (Set)
3	1	0	0	visszaállítás (Reset)
4	1	1	?	tiltott kombináció

t_n -nel jelöljük az órajel vizsgált éle előtti időt és t_{n+1} -gyel a vizsgált utáni időpillanatot.

1. eset: R = 0, S = 0:

Az órajel megfelelő éle semmit sem változtat meg, a Q₁ ütem utáni állapota ugyanaz, mint az ütem előtt volt.

Ez a tárolás esete.

2. eset: R = 0, S = 1:

Az ütem után a flip-flop beállított helyzetbe kerül, Q₁ állapota 1, Q₂-é 0. Ha a flip-flop már az ütem előtt is be volt állítva, akkor úgy is marad.

Ez a beállítás esete.

3. eset: R = 1, S = 0:

Az ütem után a flip-flop visszaállítódik. Q₁ állapota 0, Q₂-é 1 lesz. Ha a flip-flop az ütem előtt be volt állítva, akkor az ütem visszaállítja. Ha visszaállított helyzetben volt, akkor abban is marad.

Ez a visszaállítás esete.

4. eset: R = 1, S = 1:

SR flip-flop esetén ennek a kombinációjának nem szabad előfordulnia. Meghatározatlan kimeneti állapotok adódnak.

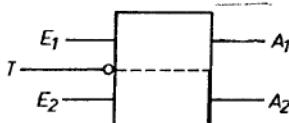
A 4. eset tiltott lehetőség.

33.3. T flip-flopok

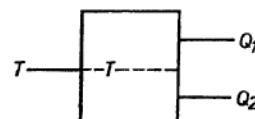
A T flip-flop az órajel minden vezérlőévének hatására ellentétes állapotba billen át. Ha pl. be van állítva ($Q_1 = 1$), akkor a vezérléskor visszaállítódik ($Q_1 = 0$). Ha a flip-flop visszaállított állapotban van, akkor a vezérlés beállítja. A T betű a trigger elnevezésre utal.

Ha egy SR flip-flopot a 33.9. ábra szerint visszacsatolunk, T flip-flop keletkezik. A 33.10. ábra egy T flip-flop rajzjelét szemlélteti.

A T flip-flopot trigger flip-flopnak vagy számláló flip-flopnak is szokták nevezni, mivel gyakran alkalmazzák számláló áramkörökben és frekvenciaosztókban.



33.9. ábra. SR flip-flop kiegészítő átkötésekkel



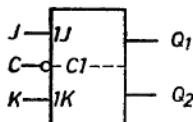
33.10. ábra. T flip-flop rajzjele

33.4. JK flip-flopok

A J és a K betűnek különösebb jelentősége nincsen, csak jelölésként használjuk.

A JK flip-flop bővített SR flip-flop. Az előzőekben, az SR flip-flopoknál említett 4. eset a JK flip-flopánál nem tiltott.

$J = 1$ és $K = 1$ esetén a JK flip-flop úgy működik, mint egy T flip-flop, azaz minden egyes vezérlő órajelre a meglevővel ellentétes állapotba vált át. Egyébként úgy működik, mint az SR flip-flop. A 33.11. ábrán a JK flip-flop rajzjelét látjuk. Az igazságítáblázatot a 33.2. táblázat közli.



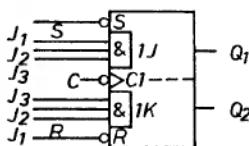
33.11. ábra. JK flip-flop rajzjele

33.2. táblázat. JK flip-flop igazságítáblázata

Eset	t_n		t_{n+1}		Funkció
	K	J	Q_1	Q_1	
1	0	0	Q_{1n}		tárolás
2	0	1		1	beállítás (Set)
3	1	0		0	visszaállítás (Reset)
4	1	1	Q_{1n}		átbillenés

A gyakorlatban univerzálisan beépíthető flip-flopokra van szükség. Ilyen flip-flopot mutat be a 33.12. ábra. Három, ÉS kapcsolatban álló J bemenet és három, ugyan-

csak ÉS kapcsolatban álló K bemenetet tartalmaz. A J bemeneteket és a K bemeneteket az órajel vezérli. Az S bemenet ütemfüggetlen beállítóbemenet, az R bemenet pedig ütemfüggetlen visszaállító bemenet. Mindkét bemenetet 0 szintű jel vezérli. Ezt fejezi ki az S és R feletti negációjel.



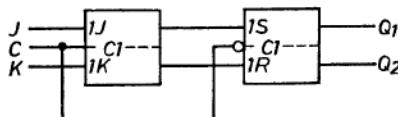
33.12. ábra. Kiegészítő lehetőségekkel ellátott JK flip-flop rajzjele

Az S-re adott 0 jel a flip-flopot beállítja ($Q_1 = 1$). Az R-re adott 0 jel a flip-flopot visszaállítja ($Q_1 = 0$).

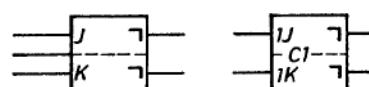
Mindez az órajel hatása nélkül, tehát ütemtől függetlenül történik. Ezáltal további lehetőségek adódnak az áramkörök felépítése terén.

33.5. Master-slave flip-flopok

A master-slave flip-flopok kéttárolós, tehát kettős flip-flopok. Az első flip-flop a „mester” (master). Ez veszi fel a bemenőjelet pl. felfutó ér segítségével. A master után van kötve a slave („szolga”) flip-flop. Ez lefutó ér hatására átveszi a jelet a master flip-floptól, és saját Q_1 kimenetén jeleníti meg. A 33.13. ábra egy JK master-slave flip-flop felépítését mutatja. A megfelelő kapcsolási jelet a 33.14. ábrán láthatjuk. A master-slave flip-flopok különösen megbízhatóak, mivel az információ felvételére és továbbítására különböző üteméleknél következik be.



33.13. ábra. JK master-slave flip-flop szerkezete



33.14. ábra. JK master-slave flip-flop rajzjele

33.6. Alkalmazások

Minden egyes flip-flop egy bináris jel tárolására alkalmas, tehát tartalma 0 vagy 1 lehet. A tartalom szempontjából a Q_1 kimenet a mértékadó.

Flip-flopok segítségével sokbites tárolók vagy regiszterek alakíthatók ki. A regiszterekbe jelek vihetők be, azok eltolhatók, majd ismét kiolvashatók (l. a 35. fejezetet). A flip-flopok ezenkívül számlálók és frekvenciaosztók létrehozásához is szükségesek (l. a 36. fejezetet). minden vezérlőkapcsolás, aminek a működésére valamiféle időbeli egymásutániság a jellemző, tartalmaz flip-flopot.

34. A digitális jelek szétválasztása és egyesítése

34.1. Adatszelektor, multiplexer, demultiplexer

Az adatszelektor feladata, hogy a különböző adatok sorozatából a kívánt adatokat kiválassza és a kimeneten keresztül továbbítsa.

A bemeneti adatok pl. időben egymás után, az ún. időmultiplex átvételi eljárással továbbíthatók. Azt az áramkört, amely a független bemenőjeleket időben egymás után a kimenetre továbbítja, multiplexernek nevezzük.

A multiplexer időtől függően vezérelt adatszelektor.

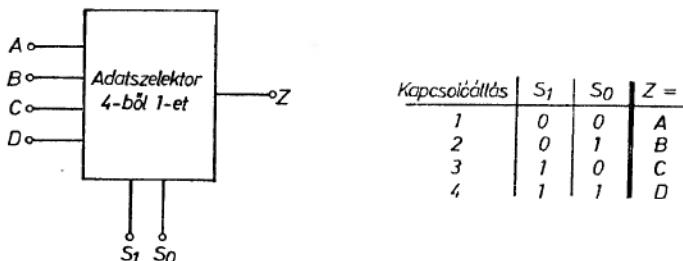
Az érkező adatok időben egymás után, különböző kimenetekre szét is oszthatók.

Azt az áramkört, amely a bemenetén megjelenő adatokat adott utasítással az egyik meghatározott kimenetre kapcsolja, demultiplexernek nevezzük.

34.1.1. 1 a 4-ből adatszelektor

Az adatszelektort működését egyszerű áramkörön keresztül mutatjuk be. Az 1 a 4-ből adatszelektornak négy bemenete van, és a négy bemenet bármelyikét össze kell tudnunk kapcsolni a Z kimenettel (34.1. ábra).

Az adatszelektor tehát úgy működik, mint egy 4 állású fokozatkapsoló. Az 1. fokozatban az A és a Z-vel van összekötve, a 2. fokozatban a B és a Z, és így tovább. A kapsolófokozat beállítása a vezérlőbemenetekkel történik. A 4 különböző kapsolófokozat digitális vezérléséhez 2 vezérlőbemenet szükséges. Két bittel négy különböző utasítás állítható elő, amelyek segítségével a négy kapsolási fokozat beállítható (l. a 34.1. ábra igazságtablázatát). Az 1 a 4-ből adatszelekció könnyen megvalósítható. Az S_1 és S_0 változóknak negált és nem negált alakban kell rendelkezésre állniuk.

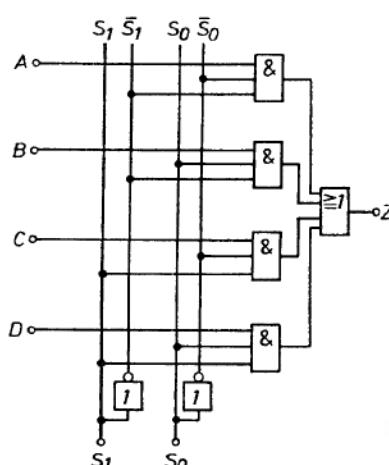


34.1. ábra. 1 a 4-ből típusú adatszelektor és igazságtablázata

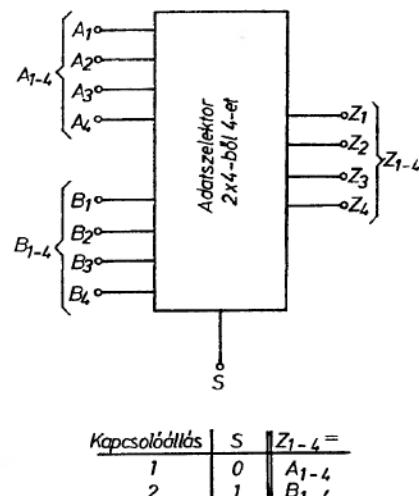
A bemeneteket ÉS kapun keresztül vezetjük tovább, ha a vezérlőbemeneteken levő jel ezt engedélyezi (34.2. ábra).

34.1.2. 4 a 2×4-ből típusú adatszelektor

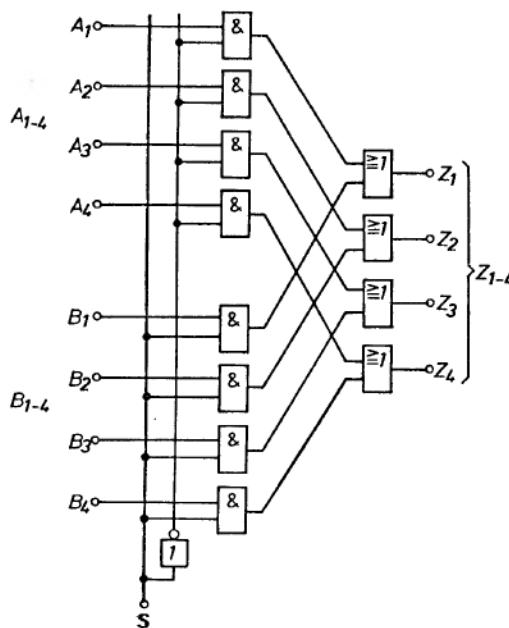
A 4 a 2×4-ből típusú adatszelektorban két 4 bites, párhuzamos bemenetet és egy 4 bites kimenetet találunk (34.3. ábra). A szelektor vagy a négy A bemenetet, vagy a négy B bemenetet kapcsolja a négy bites Z kimenetre. Mivel csak két kapcsolóállás lehetséges, ezért egyetlen vezérlőjel (S) is elegendő. Az adatszelektor kapcsolását a 34.4. ábrán láthatjuk.



34.2. ábra. 1 a 4-ből típusú adatszelektor kapcsolása



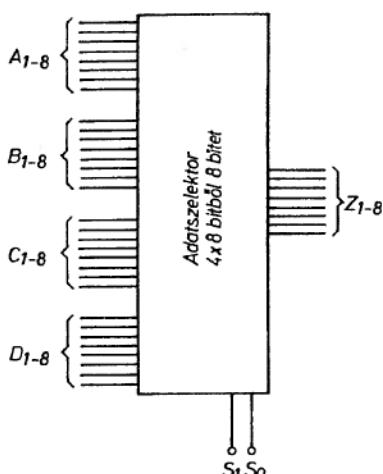
34.3. ábra. 4 a 2×4-ből típusú adatszelektor



34.4. ábra. 4 a 2×4-ből típusú adatszelektor kapcsolása

34.1.3. 8 a 4×8-ból típusú adatszelektor

További adatszelektorként érdemes bemutatni a mikroprocesszoros technikában nagy jelentőségű 8 a 4×8-ból típusú adatszelektort (34.5. ábra). Ez az adatszelektor négy 8 bites szó bármelyikét a 8 bites kimenetre tudja adni. Négy kapcsolási fokozat szükséges hozzá. A kapcsolási utasítás az S_0 és az S_1 vezérlőbemeneteken keresztül adható ki (2 bites kódok).



34.5. ábra. 8 a 4×8-ból típusú adatszelektor

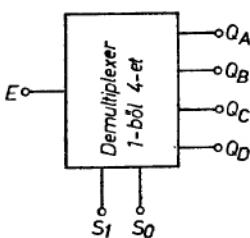
Az $S_0 = 0$, $S_1 = 0$, utasítás a nyolc A bemenetet kapcsolja össze a nyolc Z kimenettel ($Z_1 = A_1$, $Z_2 = A_2 \dots Z_8 = A_8$). Ha a B bemeneteket kívánjuk a Z-vel összekapcsolni, akkor az $S_0 = 1$, $S_1 = 0$ vezérlőkódot kell kiadnunk. A C, ill. D bemenetek kiválasztásához értelemszerűen az $S_0 = 0$, $S_1 = 1$, ill. az $S_0 = 1$, $S_1 = 1$ kódok tartoznak.

34.1.4. 1-ből 4-et típusú demultiplexer

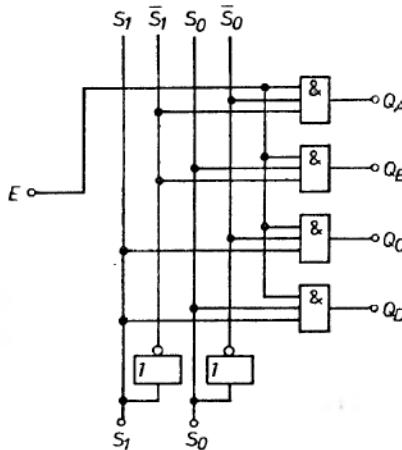
A demultiplexer fordítva működik, mint az adatszelektor vagy a multiplexer. A bemeneten levő jel választhatóan több kimenetre kapcsolható. A vezérlés utasításkódokkal történik.

A vizsgált demultiplexernek egy bemenete és négy kimenete van (34.6. ábra). Négy kapcsolási fokozatra, s ezzel négy különböző vezérlőkódra van szükség. A négy különböző utasításkódhoz két vezérlőbemenet kell (S_0 és S_1).

A demultiplexer felépítését a 34.7. ábrán láthatjuk. Az ÉS kapu csak azt a bemenő jelet engedi át, amelyre nézve a megfelelő utasításkód engedélyt ad.



34.6. ábra. 1-ből 4-et típusú demultiplexer



Kapcsolóállás	S_1	S_0	$E =$
1	0	0	Q_A
2	0	1	Q_B
3	1	0	Q_C
4	1	1	Q_D

34.7. ábra. Az 1-ből 4-et típusú demultiplexer kapcsolása

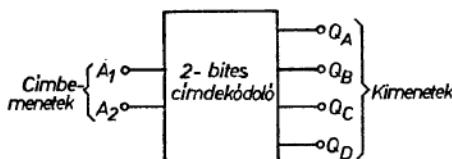
34.2. Címekódoló

A különböző elemek vezérelt eléréséhez ún. címre van szükség. A digitális technikában címen meghatározott hosszúságú bináris jelsorozatot, tehát rögzített számú bitből álló bináris szót értünk. Vannak pl. 2 bites, 4 bites stb. címek.

A címekódoló több kimenetű áramkör. A címbemenet alapján kiválasztandó kimeneten logikai 1 szint jelenik meg.

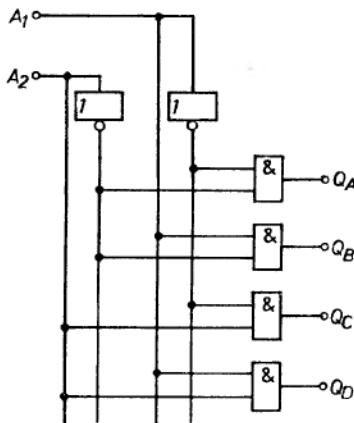
2 bites címekódoló

Ha egy címekódolónak négy kimenete van, akkor két címbemenetre van szükség. A vezérlés tehát 2 bites címmel történik. Két bittel négy különböző cím állítható elő (34.8. ábra). Egy 2 bites címekódoló kapcsolását láthatjuk a 34.9. ábrán.



Cím (szám)	A_2	A_1	Q_A	Q_B	Q_C	Q_D
1	0	0	1	0	0	0
2	0	1	0	1	0	0
3	1	0	0	0	1	0
4	1	1	0	0	0	1

34.8. ábra. 2 bites címekódoló és igazságtáblázata



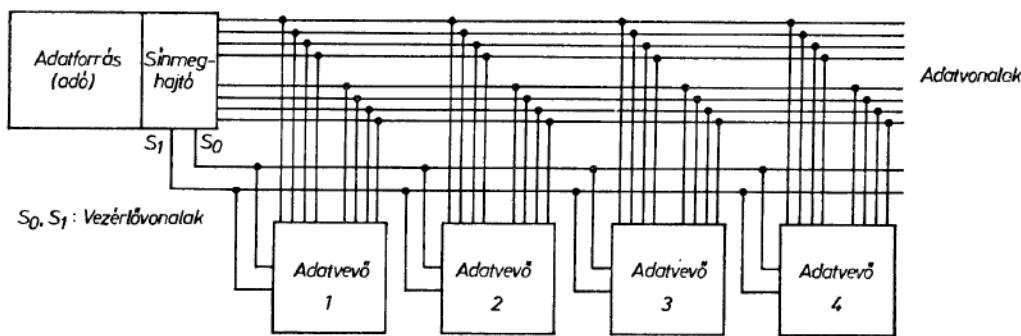
34.9. ábra. A 2 bites címdekódoló kapcsolása

34.3. Sínrendszerök

A bináris információkat átvivő és elosztó rendszert sínnek nevezünk.

Mindazokat az egységeket, amelyek bináris információkat küldenek vagy fogadnak, sínrendszeren keresztül kapcsolják össze. Ha a sínrendszer csak egyirányú információátvitelre alkalmas, egyutas vagy egyirányú sínről beszélünk. Ha az információ mindkét irányban átvihető, akkor ezt a sínt kétutas vagy kétirányú sinnek nevezzük.

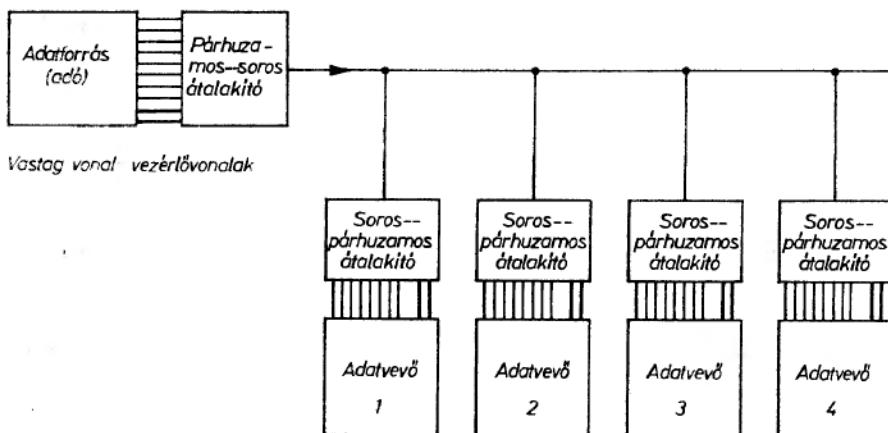
A sínrendszerök az információátvitelt párhuzamosan vagy sorosan végzik. Ennek folytán párhuzamos és soros sínrendszereket különböztetünk meg. (A soros sínrendszeret általában csak soros vonalnak nevezik.) Párhuzamos sínrendszerben az átvienő bináris szó valamennyi bitjének átviteléhez külön vonal áll rendelkezésre. Egy 8 bites szó átviteléhez tehát 8 vezeték szükséges. Ezeket a vezetékeket adatvonalaknak nevezzük. A vezérlési feladatokhoz további vezérlővonalakra van szükség (34.10. ábra).



Eset	S ₁	S ₀	Adatvevők sorszáma
1	0	0	1
2	0	1	2
3	1	0	3
4	1	1	4

34.10. ábra. Párhuzamos, egyirányú sínrendszer több adóval és vevővel

Soros sínrendszerben egyetlen vonal elegendő. Az egyes biteket egymás után továbbítják a vonalon, és a fogadóoldalon állítják össze az eredeti bináris szót (34.11. ábra). A soros sínrendszerek lassúbbak, mint a párhuzamos sínrendszerek, és áramköri költségeik is magasabbak a szükséges párhuzamos – soros – és soros – párhuzamos-átalakítók miatt. Ezért a soros sínrendszereket csak ott alkalmazzák, ahol a vezetékek költsége lényeges szerepet játszik, tehát ahol az adó és a vevő közötti távolság nagy. A legtöbb alkalmazásban az adó és a vevő közötti távolság kicsi, így a párhuzamos sínrendszer az előnyösebb megoldás.



34.11. ábra. Soros, egyirányú sínrendszer több adóval és vevővel

35. Regiszterek és tárolók

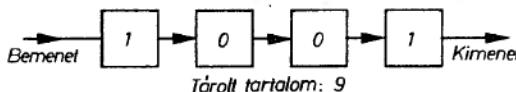
35.1. Léptetőregiszter

A léptetőregiszter bináris jelek tárolására szolgál és több flip-flop ból áll. minden egyes flip-flopnak két különböző állapota lehet. Az A_1 kimenet 0 vagy 1 lehet. Ez a két állapot megfelel a logikai állapotnak. Tehát minden egyes flip-flop egy bináris helyi értéket ábrázol, egy bit információ tárolására alkalmas.

Minden egyes flip-flop 1 bit tárolására alkalmas.

A flip-flop alapállapota (nyugalmi állapota) minden 0 tárolótartalmat jelent. Működési állapotban a tárolótartalom 1. A decimális számjegyeket pl. 4 bináris helyen, egy ún. tetráddal adjuk meg (l. a 28. fejezetet).

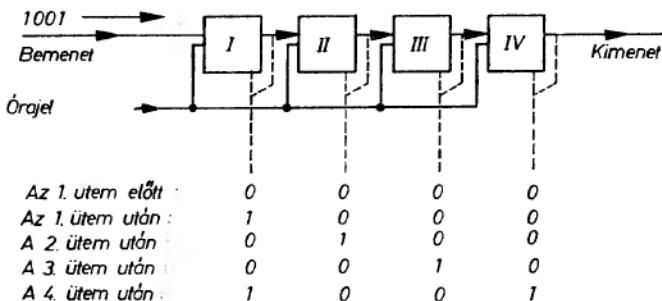
Tehát a léptetőregiszterek, amely egy ilyen tetrádot tárolni tud, négy flip-flop ból kell állnia. A 35.1. ábra a léptetőregiszter egyszerű vázlatos rajza. Mindegyik négyzetben egy flip-flopot jelképez. Az egyes flip-flopok A_1 kimeneti állapotát beírtuk a négyzetbe. A léptetőregiszterben most a decimális 9 számjegyet tároljuk.



35.1. ábra. Négy flip-flop ból álló léptetőregiszter sematikus rajza

Hogyan megy végbe most az információbevitel? Nézzük meg a 35.2. ábrát! Először mindegyik flip-flop alapállapotban ($A_1 = 0$, $A_2 = 1$), tehát 0 állásban van. A léptetőregiszter bemenetén 1 állapotnak kell lennie. Az I. flip-flopnak azonban először akkor szabad 1 állapotba kapcsolnia, amikor óraimpulzus érkezik, pontosabban amikor ez az óraimpulzus 1-ről 0-ra vált.

Az 1. óraimpulzus után az I. flip-flop 1 állapotban van, azaz az A_1 kimenetén 1 van. Az összes flip-flop órabemenete egy közös bemenetben van összefogva. A 2. óraimpulzus (röviden ütem) előtt az I. flip-flop bemenetén 0, a II. flip-flop bemenetén 1 áll-



35.2. ábra. Soros léptetőregiszterbe való információbevitel (soros adatbevitel)

pot van. A 2. ütem után az I. flip-flop 0, a II. flip-flop 1 állapotba kapcsol. A 3. ütem előtt az I. és II. flip-flop bemenete 0 állapotban van, a III. flip-flop bemenete 1 állapotú.

A 3. ütem után a flip-flopok felveszik a bemeneti állapotokat. Az I. és II. flip-flop 0 állapotú, a III. flip-flop 1 állapotú lesz.

A 4. ütem előtt a következő bemeneti állapotok állnak fenn:

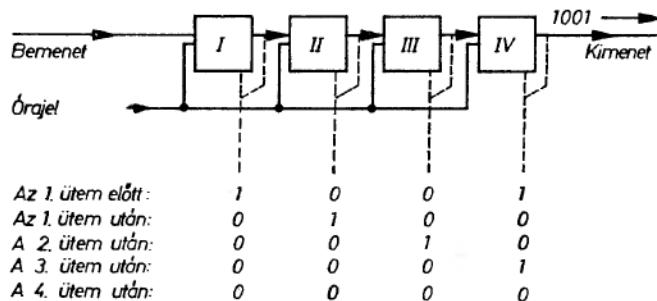
- I. flip-flop 1,
- II. flip-flop 0,
- III. flip-flop 0,
- IV. flip-flop 1.

A 4. ütem után a flip-flopok abban az állapotban lesznek, amelyek a 4. ütem előtt a bemeneteiken voltak. A léptetőregiszter most az 1001 információt tárolja.

A tárolandó tartalmat balról kezdve, bináris helyről bináris helyre léptetjük, ezáltal minden egyes ütemnél valamennyi flip-flop állapota egy hellyel jobbra tolódik.

A tárolt információt most el kell érnünk, ami azt jelenti, hogy olvashatónak kell lennie. Erre két lehetőség van:

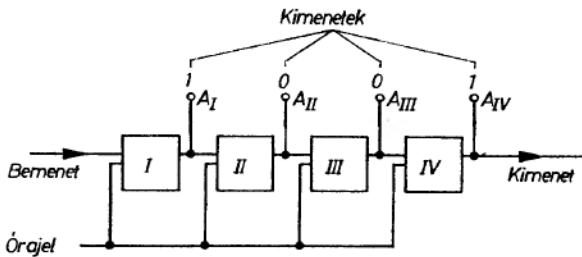
A tárolótartalmat bitenként egyesével jobbra kiléptetjük. Ez további órajelekkel történik. Az egyes állapotok a léptetőregiszter kimenetén egymás után megjelennek és onnan levezethetők (35.3. ábra). Ezt nevezzük soros kivitelnek. A kiolvasás 4. ütem után a léptetőregiszter üres lesz. Az információ a léptetőregiszter kimenetén rendelkezésre áll, de a tároló egyúttal ki is ürül. A tárolt adat törlődése viszont nem minden kívánatos.



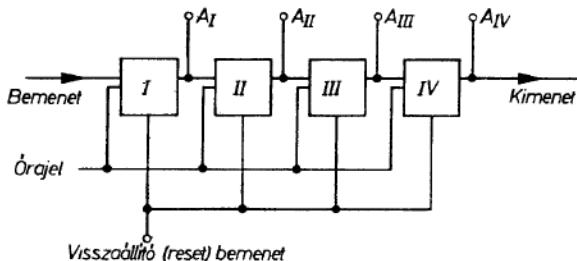
35.3. ábra. Információ kivitele léptetőregiszterből (soros kivitel)

Az információkivitel másik lehetősége az egyes flip-flopok kimeneti állapotának leolvasása. Ebből a célból a kimeneteket külön kivezetik (35.4. ábra). Az információkivitelnek ezt a fajtáját párhuzamos kivitelnek nevezik. Párhuzamos kivitel esetén az információtartalom megmarad a léptetőregiszterben. Ha az információt törölni kell, akkor vagy megfelelő számú jobbra léptetést hajtunk végre, vagy visszaállító (reset) bemenettel lájtuk el a léptetőregisztert (35.5. ábra).

Ez a külön bemenet az információ gyors törlését teszi lehetővé. A visszaállító bemenetre adott 1 állppittal az összes flip-flop alapállásba, vagyis 0 állapotba kerül.



35.4. ábra. Párhuzamos kimenetű léptetőregiszter

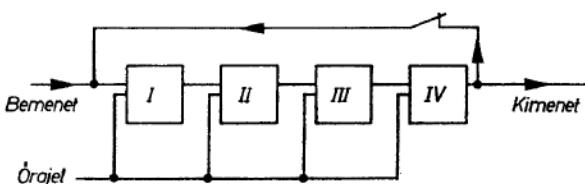


35.5. ábra. Párhuzamos kimenettel és visszaállító bemenettel rendelkező léptetőregiszter

Ha soros információkivitel esetén el akarjuk kerülni a léptetőregiszter kiürülését, akkor a kimeneten megjelenő állapotot visszacsatoljuk a léptetőregiszter bemenetére (35.6. ábra).

Ezt a fajta áramkört hurokregiszternek nevezzük.

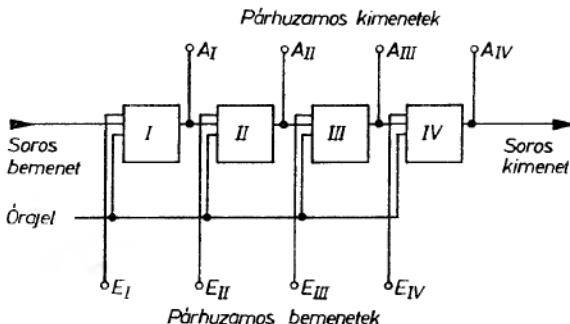
A hurokregiszter olyan léptetőregiszter, amelynek kimenete össze van kötve a bemenettel, így a tárolótartalom ciklusban körbe-körbe léptethető.



35.6. ábra. Hurokregiszterként kapcsolt léptetőregiszter

A 35.7. ábra szerinti kapcsolás egy különleges típusú léptetőregiszter, amely már tulajdonképpen nem is igazi léptetőregiszter, hanem egy léptetőregiszter és egy flip-flopos tároló (l. a 35.2. szakasz) kombinációja. Mint minden léptetőregiszter esetén, úgy itt is, az információ sorasan, az órajellel vezérelve vihető be és törölhető. Ezenkívül minden egyes flip-flop egy párhuzamos bemeneten át is vezérelhető. Az információ tehát párhuzamosan is bevihető az E_I...E_IV bemeneteken keresztül. Ugyancsak megvalósítható az A_I...A_IV kimeneteken keresztüli párhuzamos információkivitel.

Az eddigi vizsgálódások során a flip-flopokat dobozoknak tekintettük, amelyek minden úgy működtek, ahogy kívántuk. Ezeket a dobozokat most pontosan adott tulajdonságú flip-flopokkal kell helyettesíteni.



35.7. ábra. Soros és párhuzamos adatbemenettel és -kimenettel rendelkező léptetőregiszter

Nézzünk meg egy soros információbevitelre és soros, ill. párhuzamos információkivitelre alkalmas 4 bites léptetőregisztert!

Milyen típusúnak kell lenniük az áramkörhöz alkalmazott flip-flopoknak?

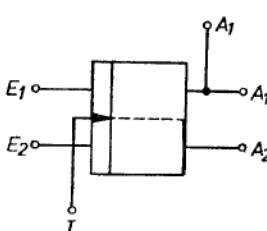
A flip-flopoknak akkor kell mintavételezni a bemenetet, amikor az órajel 1 állapotból 0 állapotba megy át. Szükség van tehát a dinamikus bemenetre mint órabemenetre, amely mindegyik flip-flopra nézve közös vezérlőjelet (órajelet) kap.

A flip-flop beállításához — így nevezik a működési állapotba való kapcsolást — a felső (A_1 -hez tartozó) részen a statikus bemenetre van szükség. A flip-flop visszaállításához — így nevezik az alaphelyzetbe vagy nyugalmi helyzetbe való kapcsolást — az alsó (A_2 -höz tartozó) részen van szükség statikus bemenetre. A statikus bemeneteknek és a közös dinamikus bemenetnek egy bemeneti áramkörön keresztül előzőleg együtt kell működniük. Magától értehető, hogy a flip-flopnak rögzített alaphelyzettel kell rendelkeznie, és külön ki kell vezetni az A_1 kimenetet. A kapcsolást a 35.8. ábra szerinti RS flip-flopok alkotják.

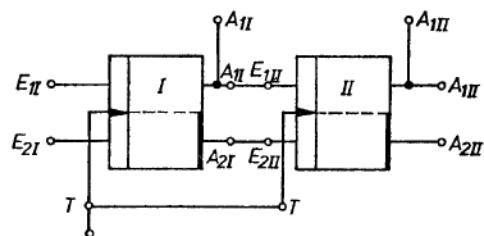
Hogyan kell a flip-flopokat összekapcsolni?

Először a bemeneteket kapcsoljuk össze közös órabemenetté.

A beállítás során az E_1 bemeneten keresztül 1 állapotot viszünk be. Az 1 állapotot az I. flip-flop A_{1I} kimenetén csak akkor adódhat tovább, ha megvan az $A_{1I} - E_{1II}$ összekötöttség (35.9. ábra).



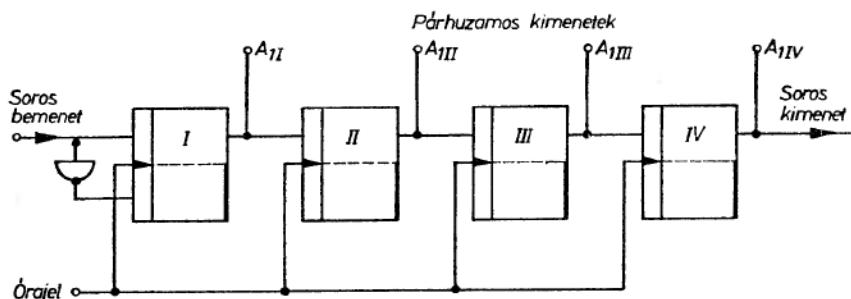
35.8. ábra. Léptetőregiszter flip-flopja (RS flip-flop)



35.9. ábra. A flip-flopok összekapcsolása

A visszaállítás az E_2 bemenetre adott 1 állappal történik. Az I. flip-flop A_{1I} kimenetén levő 0 állapotot tehát az A_{2I} kimenettel hozzuk létre. Ha az $A_{1I} = 0$, akkor $A_{2I} = 1$. Csak akkor lehet a II. flip-flopot visszaállítani, ha ez az 1 állapotot az E_{2II} bemenetére hat, ezért az A_{2I} és E_{2II} közti összekötöttsére szintén szükség van (35.9. ábra). Tehát minden A_1 kimenetet összekötjük az E_1 bemenetekkel, és minden A_2

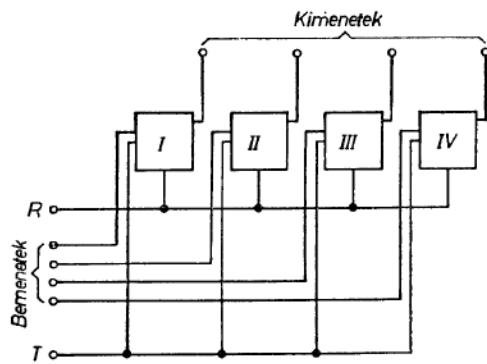
kimenet össze van kötve mindegyik E_2 bemenettel. Így a 35.10. ábrán látható kapcsolás adódik. A soros bemeneten beadott 0 állapot még nehézségeket okoz. Ahhoz, hogy az I. flip-flopot 0 állapotba állítsuk, az E_{2I} bemeneten 1 állapotnak kell lennie. Ezt az állapotot az E_{1I} bemeneten levő 0 állapot felhasználásával hozzuk létre úgy, hogy az E_{1I} bemenetet egy inverteren keresztül kötjük össze az E_{2I} bemenettel (35.10. ábra).



35.10. ábra. Soros be- és kimenettel és párhuzamos kimenettel rendelkező 4 bites léptetőregiszter

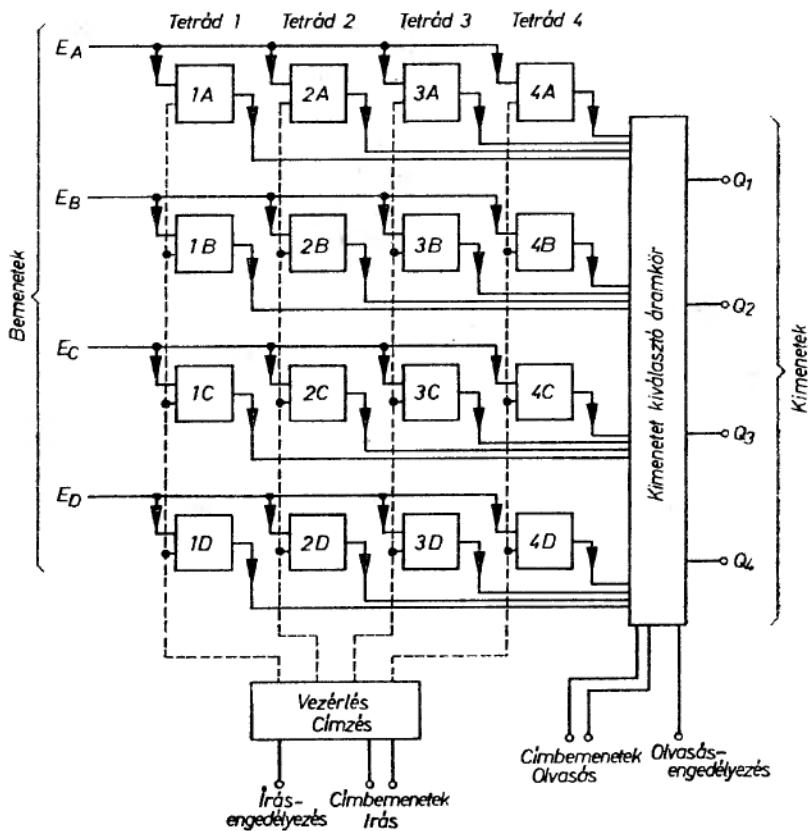
35.2. Flip-flopos tároló

Minden egyes flip-flop 1 bit információ tárolására képes. Ezért tárolóeszközöként flip-flopot alkalmazhatunk. Az ilyen típusú tárolókat flip-flopos tárolóknak, félvezető tárolóknak vagy szilárdtest-tárolóknak nevezünk. A flip-flopos tároló abban különbözik a léptetőregisztertől, hogy nincs lehetőség soros adatbevitelre és -kivitelre. Az egyik flip-flop ról a másikra eltolással nem továbbítható információ. Mindegyik flip-flop külön bemenettel vezérlik, és minden egyes flip-flopnak saját kimeneti vonala is van. Az információt párhuzamosan adjuk be és párhuzamosan olvassuk ki. A 35.11. ábra egyszerű 4 bites, flip-flopos tároló kapcsolását mutatja be. Az információ bevitelre órajellel vezérelt. A 4 bites, flip-flopos tárolót ritkán alkalmazzák, mert túl kicsi a tárolókapacitása.



35.11. ábra. Egyszerű 4 bites flip-flopos tároló

A 35.12. ábra egy 4 tetrádos (16 bites) írható-olvasható tárolót ábrázol. Mindegyik tetrádnak saját címe van, az információ beírásakor a címnek a címbemeneten kell lennie.



35.12. ábra. 4 tetrádos, 16 bites írható-olvasható tároló

A tároló működéséhez írásengedélyező jelnek kell lennie a megfelelő bemeneten. Az információ olvasásakor ugyancsak egy címnek és engedélyezőjelnek kell lennie az adott bemeneteken. A flip-flopos tárolókat az utóbbi időben minden szélesebb körben alkalmazzák. Integrált áramkör formájában kínálják őket. Nagyobb alkatrészsűrűség esetén az egy bitre eső költségek kisebbek. Olyan integrált áramköröket is készítenek, amelyeknek a tárolókapacitása 256 000 bit, vagy még több.

35.3. Írható-olvasható (RAM) tárolók

RAM-nak nevezük az általában félvezetőkkel (integrált áramköri formában) kialakított, írásra-olvasásra egyaránt alkalmas tárolót. A RAM meghatározott számú tárolóhellyel rendelkezik, és minden egyes tárolóhelynek rögzített (tipikusan 1, 4 vagy 8 bites) kapacitása van, tehát adott hosszúságú információt tud felvenni. Az egyes tárolóhelyekre címekkel lehet hivatkozni. A RAM tárolóhelyei tehát szabadon választhatók. A RAM elnevezés az angol *Random Access Memory* (véletlen elérésű tároló) rövidítése. A tárolóhelyet (rekeszt) a címével választjuk ki, majd beleírjuk az információt. Az információkivitel szintén a cím alapján történik. Kiolvashatjuk az információt anélkül, hogy a tárolórekesz tartalmát törölünk. Ha már nincs szükség az

adatra, akkor a RAM törölhető, vagy a pedig a tárolórekesz új információval tölthető fel. A RAM-okat újabban kizárálag integrált áramkörök kent alakítják ki. Különbösséget teszünk statikus és dinamikus RAM között. Statikus RAM-nál a tárolórekeszek flip-flopokból állnak, és az egyes biteket egy-egy flip-flopban tároljuk. A dinamikus RAM-okban a tároláshoz belső kapacitásokat alkalmaznak. minden egyes bitet egy kis kondenzátorban tárolják. Mivel a szivárgási áram nem végtelenül kicsi, töltési veszteségek lépnek fel, amit rövid időközönként frissítéssel kell pótolni.

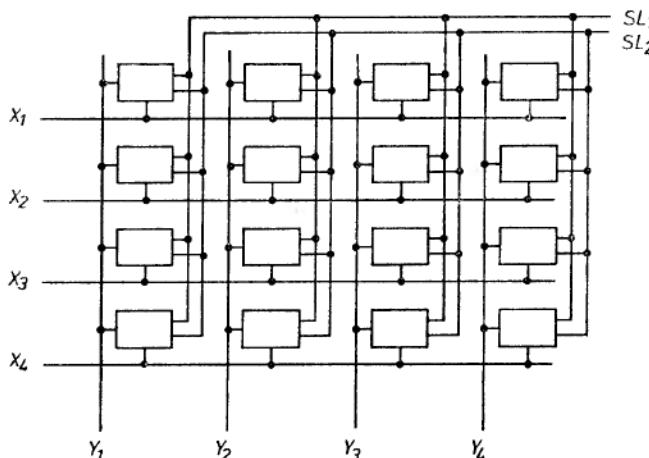
A statikus és dinamikus RAM-ok ideiglenes (átmeneti) tárolók. A tápfeszültség kimeradása esetén a tároló tartalma elvész.

A tárolótartalom elvesztése elleni biztosítékul telepeket ajánlatos beépíteni. A nem elvesző tartalmú statikus RAM-ok kifejlesztése már megtörtént, de jelenleg még ezek a RAM-ok nagyon drágák. Ennek ellenére a jövőben várhatóan nagyobb jelentőségük lesz. A statikus RAM-okat különféle technológiákkal állítják elő. Az áramkörök különböző áramkörcsaládokhoz tartoznak: vannak TTL, ECL, NMOS és CMOS technológiával gyártott RAM-ok. A dinamikus RAM-okat különféle MOS technológiákkal készítik.

35.3.1. A tárolók felépítése

A statikus és dinamikus RAM-okat különböző tárolókapacitással, különféle szervezési formákban kínálják. A cím segítségével kiválasztható tárolórekesz egy vagy több tárolóelemből állhat. Ha csupán egy tárolóelemből áll, akkor bitszervezésű tárolóról beszélünk. minden egyes tárolóelemnek, azaz minden egyes bitnek saját címe van és ezzel választható ki. Az ilyen tárolók felépítésének vázlatát mutatja a 35.13. ábra. A 16×1 jelölés jelentése: teljes kapacitás 16 bit, egy tárolórekesz kapacitása 1 bit. Ha a tárolórekesz több tárolóelemből áll, akkor a tároló szószervezésű.

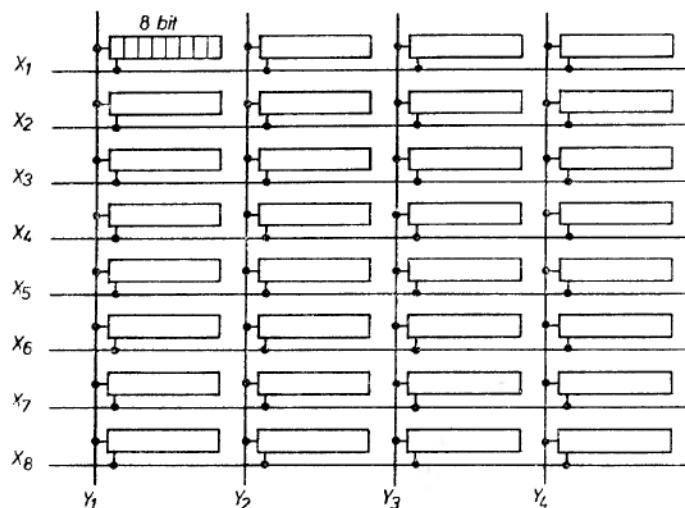
A 35.14. ábrán egy 32×8 bites tároló felépítésének vázlata látható. A tároló 32 tárolórekeszt tartalmaz, amelynek mindegyike 8 bites. minden egyes 8 bites egység egyetlen



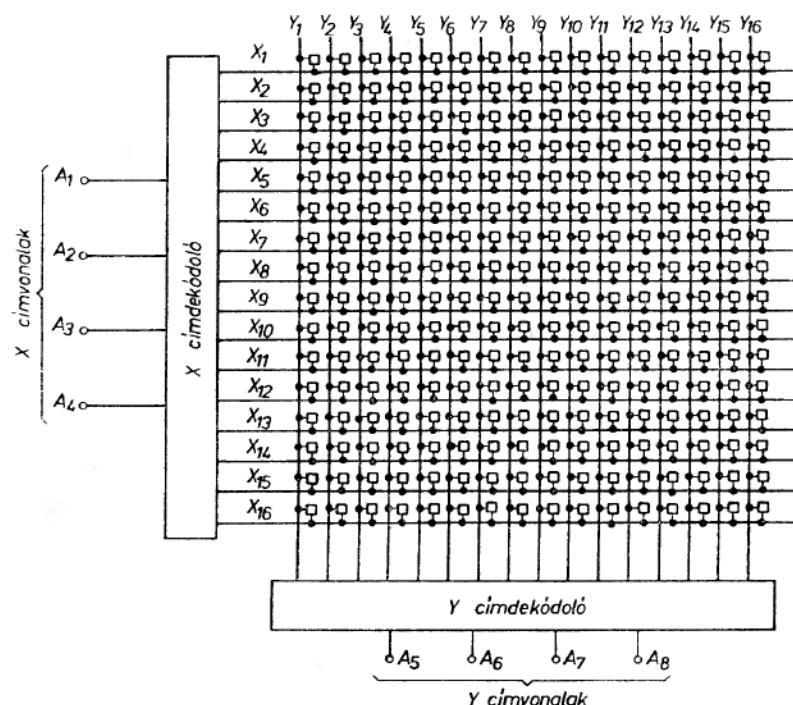
35.13. ábra. 16×1 bites tároló szerkezete

címmel választható ki. A tárolórekesz 8 bitjét minden egyszerre írjuk és olvassuk. Egy 256×1 -es tárolóhoz 16X koordinátavonal és 16 Y koordinátavonal szükséges (35.15. ábra).

Kedvezőtlen lenne, ha ezeket a koordinátavonalakat kívülre, tehát az integrált áramkör csatlakozókapcsaira vezetnénk ki. Az ilyen áramköröknek nagyon sok csatlakozópontja lenne, ezért címdekódolót kellene alkalmazni (l. a 34. fejezetet).

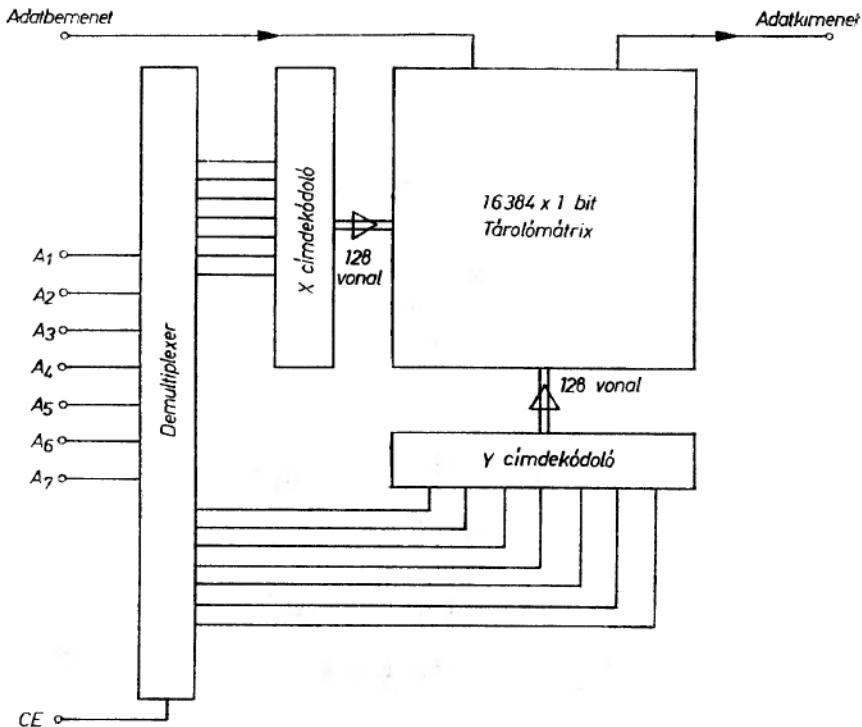


35.14. ábra. 32×8 bites tároló szerkezete
(SL_1 és SL_2 nyolc vezetékét az áttekinthetőség kedvéért elhagytuk)



35.15. ábra. 256×1 bites, címekóderes tároló szerkezete
(SL_1 és SL_2 nyolc vezetékét az áttekinthetőség kedvéért elhagytuk)

A 16 koordinátavonalal kiválasztásához 4 címvonal kell. A címvonalakat az integrált áramkör csatlakozókimeneteire vezetik ki. Hogy néz ez most ki egy koordinátavonalakkal és címvonallakkal rendelkező $16 \text{ Kbit} \times 1$ bites tároló esetén? 16 384 bitnek (16 Kbitnek) kell kiválaszthatónak lennie. Ehhez 128 X koordinátavonalra és 128 Y koordinátavonalra van szükség. A 128 koordinátavonalal kiválasztásához 7 vezérlővonal szükséges (35.16. ábra). Összesen 14 címvonalat kellene az integrált áramkör kivezetéseihöz vezetni. Mivel azonban az adatbevitel és -kivitel, valamint a vezérlőjellek (mint pl. az író- és olvasójelek) számára további kivezetésekre van szükség, igen sok csatlakozókimenet adódna. Ennek elkerülésére demultiplexert (34. fejezet) építenek az áramkörbe. Az $A_1 \dots A_7$ bemenetekre először az X címeket adjuk, majd ezután ugyanazokra a bemenetekre az Y címeket. Az átkapcsolás egy S vezérlőjellel történik. A címjelek multiplexálása kisebb IC-tokok alkalmazását teszi lehetővé.



35.16. ábra. Címdekódolóval és multiplexerrel ellátott $16 \text{ K} \times 1$ bites tároló szerkezete

35.3.2. A tárolók jellemző adatai

A tárolók megválasztása szempontjából a tároló jellemző adatainak nagy jelentősége van. Itt elsősorban tárkapacitásról és társzervezésről van szó, de lényeges a működési sebesség és a fogyasztás is. A következőkben a tárolók legfontosabb jellemzőit vizsgáljuk meg.

Tárolókapacitás

A tárolókapacitás a tároló áramkörben levő tárolóelemek számát, pontosabban a tárolható bitek számát adja meg.

Társzervezés

A társzervezés egy tárolórekesz kapacitásáról és a hozzáférés módjáról ad felvilágosítást.

Elérési idő

Az elérési idő az az idő, amely adott tárolóelem címzésének időpillanatától kezdve az információ adatkimeneten való megjelenésig eltelik.

Ciklusidő

Ciklusidőn két egymásra következő írási és olvasási művelet közötti legrövidebb időt értjük.

Fogyasztás

Az integrált áramkör összes teljesítményszükségletét adja meg. Üzemi és nyugalmi állapotban eltérő lehet.

Elektromos üzemi feltételek

Itt adjuk meg a szükséges tápfeszültsége(ke)t és a szükséges jelszinteket, valamint ezek tűrését (l. a 32. fejezetet), ill. az elektromos határértékeket.

Üzemihőmérséklet-tartomány

Az üzemihőmérséklet-tartomány az, amelyben a tároló az előírt üzemi feltételek mellett biztonságosan működik.

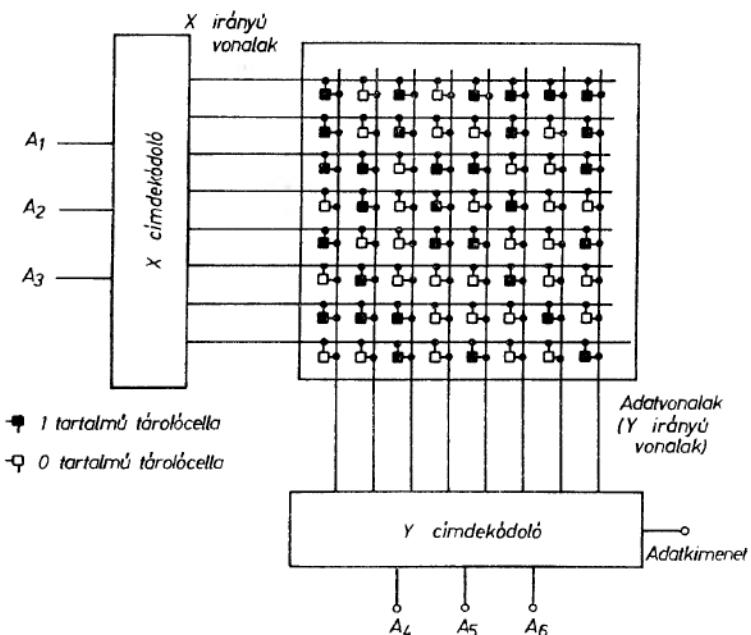
35.4. Csak olvasható (ROM) tárolók

A csak olvasható tárolók nem törölhető és nem változtatható információt tartalmaznak. A ROM megnevezés az angol *Read Only Memory* (csak olvasó memória) rövidítése. Az információt a gyártó vagy a felhasználó írja be a ROM-ba. A ROM egy könyvhöz hasonlítható: a benne levő információ bármikor kiolvasható. Nem lehet viszont ezt az információt másikkal kicserélni. ROM-ban tároljuk a gyakran használt információkat, pl. vezérlőprogramokat, ill. táblázatokat. Lehetséges lenne pl. hogy a bértáblázatot ROM-ban tároljuk. Szükség estén az egyes táblázati értékek kiolvashatók.

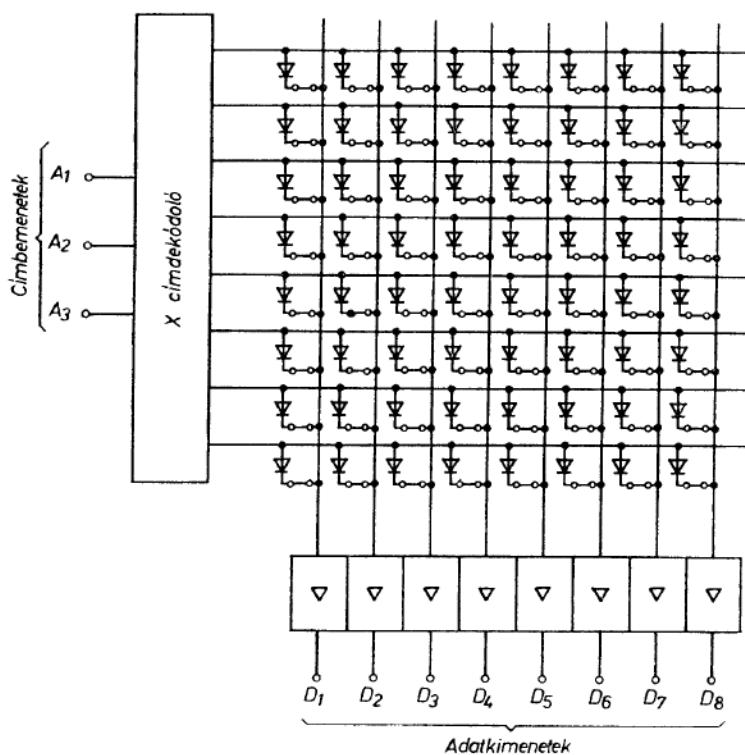
A ROM felépítéséhez kétfajta tárolóelemre van szükség. Az egyik fajta tárolóelemnek minden 1, a másiknak minden 0 értéket kell tartalmaznia.

A ROM felépítése és szervezése a RAM-éhoz hasonló. A tárolómátrix sorokból és oszlopokból áll. Az egyes tárolórekeszeket a cím segítségével választjuk ki (35.17. ábra).

A ROM-okat többnyire NMOS technológiával állítják elő, amelynek lényege, hogy az integráció sűrűsége nagy, a fogyasztás kicsi. Milyen annak a tárolóelemnek a szerkezete, amelynek tartalma minden 1? Az 1-et egy hiányzó NMOS tranzisztor állítja elő. Azt a tárolóelemet, amelynek az értéke minden 0, NMOS tranzisztorral hozzuk létre.



35.17. ábra. 64×1 bites ROM szerkezete



35.18. ábra. 8×8 bites diódás PROM felépítése

35.5. Programozható, csak olvasható (PROM) tároló

A PROM az angol *Programmable Read Only Memory* (programozható, csak olvasható tároló) rövidítése. A PROM kifejlesztését az az igény váltotta ki, hogy a felhasználó maga szerette volna a saját adatait a tárolóba beírni. Nem akarta azt sem, hogy nagy darabszámokhoz legyen kötve. A kis sorozatnagyságú, sőt egyedi ROM-ok gazdaságos előállítása volt a cél.

Képzeljünk el egy olyan ROM-ot, amely csupa 0 típusú tárolóelemből épül fel (35.17. ábra)! A vonalak kereszteződési pontjaiban tehát térvézérlésű tranzisztorok vannak. Ha valamelyik tranzisztor átégné, azon a helyen 1 adat tárolódna. Miért ne égetténk át a tranzisztorokat szándékosan azokon a helyeken, ahol az 1 adatokat szeretnénk? A PROM-ot így programozzák, azaz látták el információval. Különféle PROM-típusok vannak. A bipoláris PROM-oknak, amelyeknek a csomópontjaiban diódák és tranzisztorok vannak, jelenleg igen nagy a jelentőségük. A 35.18. ábrán egy 8×8 bites diódás PROM-szerkezet látható. A diódának igen vékony (20...30 nm széles, 100 nm vastag) krómnikkel ötvözettel készült hozzávezetései vannak. Ha az áram egy adott értéket tűllép, ezek a vezetékek leélegnek. A PROM programozásához külön programozóberendezés szükséges. Magától értetődő, hogy az információtárolás folyamata nem fordítható vissza. Ha az ember eltéveszti, akkor a PROM többnyire selejt, és el lehet dobni. A legritkább esetben lehet csak kijavítani a PROM-okat.

35.6. Törölhető, újraprogramozható, csak olvasható (EPROM) tároló

A törölhető és újraprogramozható, csak olvasható tároló lehetővé teszi a bevitt információ törlését és az azt követő újraprogramozást.

A törlést és az újraprogramozást elvileg tetszőleges gyakorisággal megismételhetjük anélkül, hogy a tárolóelem károsodna.

Az EPROM-oknak két csoportját különböztetjük meg. Az egyik csoportban az információt ultraibolya (UV) fénnel törlik. Ezt a fajta állandó tárolót EPROM-nak (*Erasable Programmable Read Only Memory* – törölhető, programozható állandó tároló) vagy REPROM-nak (*Reprogrammable Read Only Memory* – újraprogramozható állandó tároló) nevezik.

A másik csoportba tartozó törölhető, programozható állandó tárolók tartalmát elektromos feszültséggel törlik. Ezt a típust EEPROM-nak (*Electrically Erasable Read Only Memory* – elektromos úton törölhető állandó tároló) vagy EAROM-nak (*Electrically Alterable Read Only Memory* – elektromos úton átprogramozható állandó tároló) nevezik.

Az EPROM-ok tárolócelláit az X és az Y koordináta vonalakon keresztül kiválasztva, egymás után programozzák. A programozás folyamata a biztonság kedvéért többször megismételhető. A gyártó adatai alapján a MOSFET kapu töltése több évig megmarad. Az adatok megőrzési ideje 1 és 100 év között változik.

A programozott EPROM a bevitt információt állandóan megőrzi.

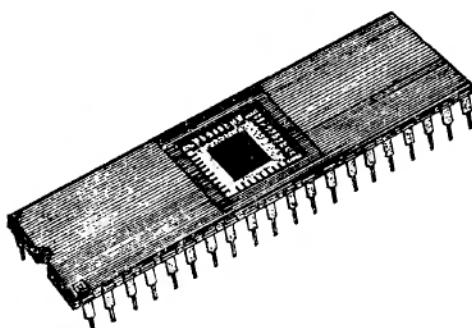
A nevesebb gyártók az adatok megőrzésére 10 év garanciát vállalnak.

Az EPROM tetején levő ablakon keresztül ultraibolya fénnyel töröljük a tárban levő információt.

A törlés azon alapszik, hogy a jó szigetelő tulajdonságú anyagot a besugárzás ionizálja és gyengén vezetővé teszi. A kapuk töltése lassan megszűnik, kb. 10 Ws/m^2 sugárzási teljesítmény hatására a kapu 20...30 perc alatt kisül.

Az EPROM tokján a kristálylapka teljes felületére kiterjedő ablak van (35.19. ábra), amit kvarcüveg fed. Az UV-fény tehát az összes tárolóelemet eléri és egyszerre törli.

Az EPROM törlésekor minden az összes információ törlődik.



35.19. ábra. EPROM tokozása

Törlés után az alkatrésznek hűlni kell, mert észrevehető mértékben felmelegszik. mindenekelőtt a szigetelőanyag ionizációjának kell megszűnnie, és az anyagnak újból teljesen szigetelővé kell válnia. Az újraprogramozás csak ezt követően kezdődhet meg. A korai EPROM-oknál a hűtési időnek legalább fél óráig, de inkább egy teljes óráig kellett tartania, az új változatok külön hűtést nem igényelnek.

A fénysugárzásnak kitett EPROM akaratlanul is törlődhet.

Napsugárzás hatására kb. 3 nap alatt bekövetkezik a törlés. Egy fénycső fénye kb. 3 héttel törli az információt. A nem szándékos törlés elkerülése céljából célszerű az ablakot sötét ragasztószalaggal letakarni.

A törlési folyamat hatására az alkatrész anyaga nem számottevően változik, úgyhogy nagyon sok törlésre és újraprogramozásra van lehetőség.

Az EPROM típusú, törölhető, újraprogramozható, csak olvasható tárrak kapacitása néhány 100 bit-től 64 Kbyte-ig ($8 \times 65\,536 = 524\,288$ bitig) terjed.

EEPROM tárolók

Az EEPROM típusú tárrak az előzőekben leírt, csak programozható tárrakhoz hasonló szerkezetűek, és szintén törölhetők és újraprogramozhatók. A törlés és programozás tetszőleges számban megismételhető. Mégis van azonban egy lényeges különbség:

Az EEPROM típusú tárolók törlése elektromos úton történik.

A tárolócellák általában két önzáró, n csatornás MOSFET-ből épülnek fel. A programozás ugyanúgy történik, mint az EPROM esetében.

Az elektromosan tölthető állandó tárakat ma úgy alakítják ki, hogy a bennük levő összes információt egyszerre lehessen törlni. Indokolt lenne, ha az EEPROM megnevezést ezekre az egyszerre törlhető elemekre alkalmaznánk.

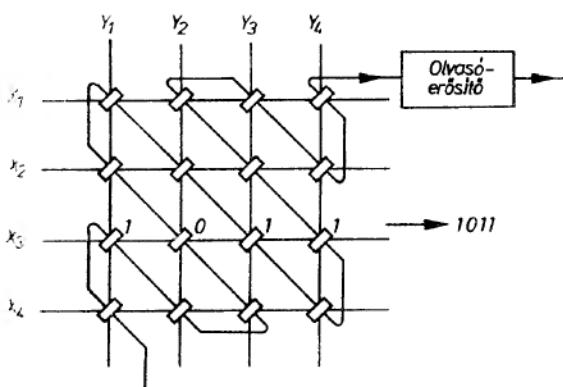
Lehetőség van azonban arra is, hogy az EEPROM tárolót úgy alakítsuk ki, hogy minden egyes tárolóelemet egyenként lehessen törlni. Az ilyen tároló bitről bitre átprogramozható. Az effajta tárolót EAROM-nak (*Electrically Alterable ROM* – elektromos úton átprogramozható ROM) is nevezik.

35.7. Ferritmagos tároló

A ferritmagos tároló ferritgyűrűkből épül fel. minden egyes ferritgyűrű 1 bitet tárol. A gyűrű mágnesezettségi állapota a tápfeszültség kiesése esetén is megmarad, a ferritmagos tároló tehát nem elvesző tartalmú tároló. A tárban levő adatok elérési ideje csekély (kb. 0,5 μ s). Előállítása igen költséges. Napjainkban már nem nagyon alkalmazzák az ilyen tárolókat a számítástechnikában, mert a félvezetős tárolók egyre inkább kiszorítják őket. Az információ bevitelre többséleképpen történhet. Az ún. féláramú eljárást könnyű megérteni. Tekintsük a 35.20. ábrát! A tárolómátrix valamennyi gyűrűjének 0 állapotban kell lennie, tehát nem tartalmazhat információt.

Az információ tárolásához az szükséges, hogy bizonyos gyűrűket 1 állapotba állítsunk, ugyanakkor másokat meghagyunk 0 állapotban.

A tárolást beírásnak is nevezik. Egy ferritgyűrű átbillentéséhez kb. 300 mA áram szükséges, azaz a gyűrűkön át megfelelő irányban 300 mA-es áramnak kell folynia ahhoz, hogy 1 állapotba billjen. Ha egy adott ferritgyűrűt 1 állapotba kell kapcsolni, akkor a két koordinátavezetéken egyenként kb. 150 mA-es áramot kell átvezetni. Ha pl. a 2. sor 3. elemét kell 1 állapotba hozni, akkor az X_2 koordinátavezetéken 150 mA



35.20. ábra. Ferritmagos tárolómátrix felépítése

áramnak, az Y_3 vezetéken keresztül szintén 150 mA áramnak kell folynia. A kiválasztott gyűrűn keresztül most összesen 300 mA áram fog átfolyni, aminek hatására a ferritgyűrű átbillen. A biztonság kedvéért a féláramnál valamivel nagyobb áramerősséget (pl. 160 mA) szoktak átengedni a koordinátavezetéken. Az X_2 koordinátavonal többi ferritgyűrűjén szintén 160 mA fog átfolyni. Ugyancsak 160 mA folyik át az Y_3 koordinátavonal többi ferritgyűrűjén is. Ez az áram az átbillenéshez nem elegendő, így a koordinátavonalakhoz tartozó többi ferritgyűrű nem fog átbillenni.

A féláramok segítségével az 1 állapotba való átkapcsolás csak sorban egymás után lehetséges. Az áramok vezérléséhez külön áramkör szükséges. Az információ kivitelét olvasásnak nevezzük. Olvasáskor meg kell állapítani, hogy melyik gyűrű van az 1 és melyik a 0 állapotban. Az egyes ferritgyűrűket egymás után kérdezik le úgy, hogy koordinátavonalaikon az íráskor alkalmazottal ellentétes irányú féláramokat küldenek át. Ha a gyűrűt 0 állapotban találjuk, akkor ezzel az árammal a negatív telítettség irányában tovább mágnesződik, tehát mágneses tere alig változik. Ha a ferritgyűrű 1 állapotban volt, akkor átbillen 0 állásba, mágneses tere megfordul, és az olvasónalon feszültségimpulzus indukálódik. A feszültségimpulzust erősítővel felerősítik és feldolgozzák.

Olvasáskor minden gyűrű, amelyik 1 állapotban volt, 0 állapotba billen át, ezáltal az információ törlődik.

Az információ olvasáskor bekövetkező törlődése hátrány. Ha továbbra is szükség van az adatra, átmenetileg tárolni kell, és az olvasás után újból be kell írni.

36. Számláló áramkörök

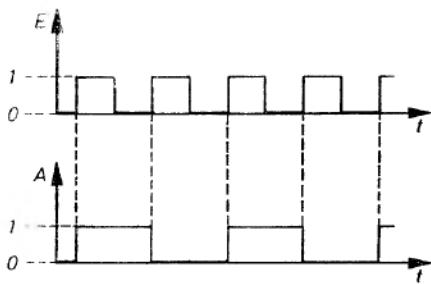
36.1. Frekvenciaosztó

A 36.1. ábra szerinti U_E négyszögrezgés frekvenciáját 2:1 arányban kell leosztani, vagyis az alapfrekvenciáját felezni kell. Ez flip-floppal valósítható meg. Egy flip-flopra van szükség, amely minden egyes bemeneti impulzus hatására átbillen. Itt elvben teljesen mindegy, hogy ez a flip-flop a bemenőjel 0-ról 1-re, vagy 1-ről 0-ra átmeneténél billen. A feladathoz T flip-flopot választunk.

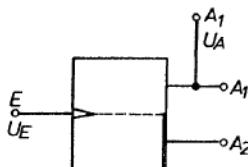
A flip-flop (36.2. ábra) nyugalmi állapotban van ($U_A = 0$ V).

Az első impulzus selfutó élénél a flip-flop működési állapotba kapcsol át ($U_A = +5$ V). Újabb átbillenés a második impulzus selfutó élénél következik be. Ekkor a flip-flop újból nyugalmi állapotba vált át ($U_A = 0$ V).

A harmadik impulzus selfutó érére a flip-flopot ismét működési állapotba billentik. A flip-flopot kimenetén megjelenő U_A feszültség frekvenciája fele az alapfrekvenciának.

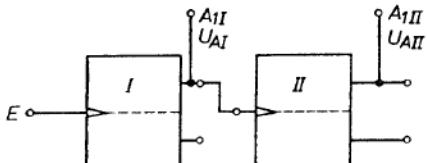


36.1. ábra. 2:1 arányú frekvenciaosztó



36.2. ábra. Számláló flip-flop

Tekintsük most a 36.3. ábrán bemutatott két flip-flopos kapcsolást és a 36.4. ábra szerinti diagramot! Az 1. impulzus pozitív élénél az I. flip-flop működési állapotba, az I. flip-flop A_1 kimenete pedig 0-ból 1-be megy át. Ezáltal azonban a II. flip-flop működési helyzetbe billen, mert e bemenete az A_1 -vel van összekötve.



36.3. ábra. Két flip-flop összekapcsolásával létrehozott frekvenciaosztó

A 2. impulzus felfutó élével az I. flip-flop ismét nyugalmi helyzetbe kapcsol ($A_1 = 0$). A II. flip-flop erre nem reagál, mivel az A_1 1 állapotból 0 állapotba megy át. A 3. impulzus felfutó éle az I. flip-flopot türa működési állapotba billenti, s mivel A_1 most 0-ról 1-re vált, a II. flip-flop is átbillen. Az U_{AI} négyszögfeszültség alapfrekvenciája a II. flip-flop kimenetén az U_E alapfrekvenciájának csupán egynegyede.

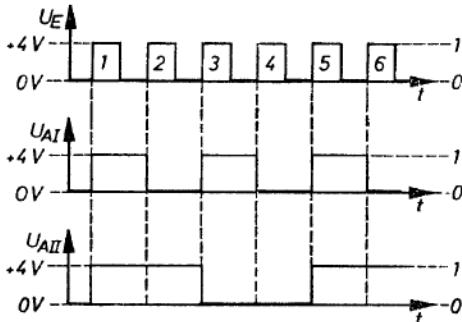
Minden egyes flip-flop felére csökkenti a bemenőjel frekvenciáját.

Feladat

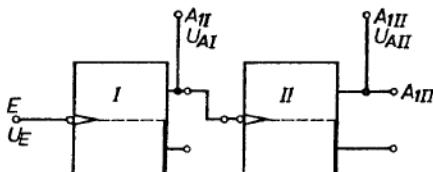
A 36.5. ábra szerinti frekvenciaosztó áramkört a 36.4. ábra szerinti U_E négyszögfeszültséggel vezéreljük. Hogy néz ki az U_{AI} és U_{AI} kimeneti feszültségek idődiagramja?

Megoldás

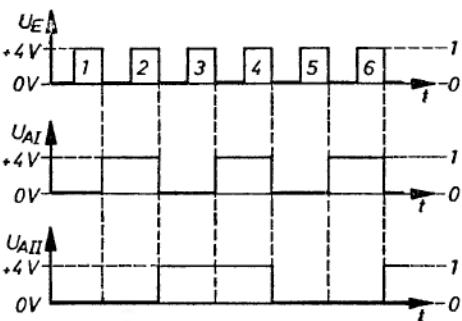
Olyan flip-flopokat alkalmazunk, amelyek a bemenőjel 1 állapotából 0 állapotba való átmeneténél kapcsolnak. Csak a 36.6. ábra U_{AI} diagramján látható lefutó élek váltanak ki átkapcsolást. Az U_{AI} feszültségnek az U_E bemeneti feszültséghoz képest fél-



36.4. ábra. A 36.3. ábra szerinti kapcsolás idődiagramja



36.5. ábra. Frekvenciaosztó kapcsolás



36.6. ábra. A 36.5. ábra szerinti kapcsolás idődiagramja

akkora alapfrekvenciája van. Az U_{AII} feszültség alapfrekvenciája az U_E alapfrekvenciájának csupán egynegyede.

A frekvenciaosztást gyakran alkalmazzák elektronikus óráknál. Egy kvarcgenerátor egészen nagy frekvenciáját az osztóval 1 Hz-re, tehát másodperces ütemig le lehet osztani.

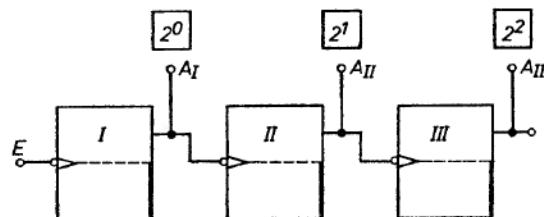
36.2. Előreszámítás

A 36.7. ábrán látható kapcsolás nagyon hasonlít a 36.5. ábrán bemutatott frekvenciaosztó áramkörhöz, csak a III. flip-flop révén további osztófokozat van hozzákapcsolva.

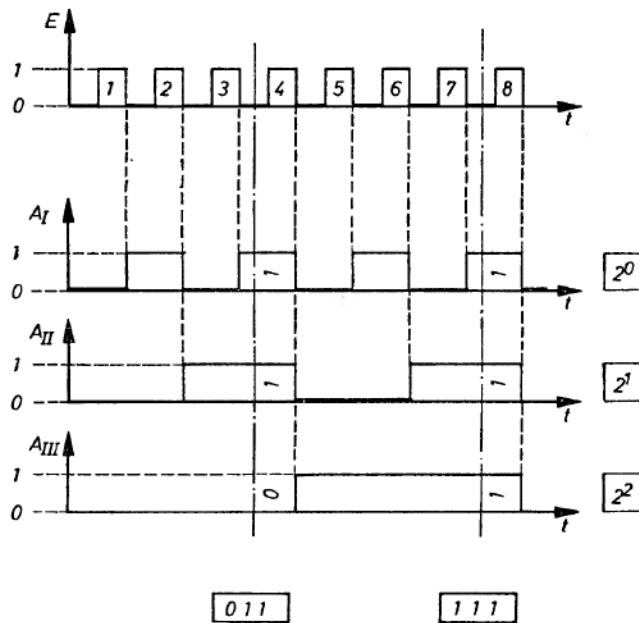
Ha az A_I , A_{II} és A_{III} kimenetekhez 2^0 , 2^1 és 2^2 értékeket rendeljük, akkor megállapíthatjuk, hogy ez a frekvenciaosztó áramkör mint bináris számláló működik, azaz a bemeneti impulzusokat számlálja.

A 3. impulzus után a kimeneteken a $011 = 3_{10}$ bináris szám jelenik meg, a 7. impulzus után pedig a bináris $111 = 7_{10}$ (36.8. ábra) érték.

A 8. impulzus után valamennyi kimenet 0 állapotba kerül. A három flip-flopból álló számláló csak 7-ig tud számolni. A három flip-flopból álló számlálókapcsolást kissé átrajzolhatjuk úgy, hogy a 36.9. ábra szerinti kapcsolás adódjék, ahol a kimenetek



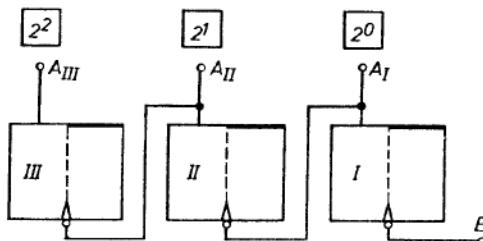
36.7. ábra. Bináris előreszámítás



36.8. ábra. A 36.7. ábra szerinti kapcsolás idődiagramja

helyzete a bináris számok helyi értékének felel meg. Ez az ábrázolás azonban szám-lálónál nem meggyőző.

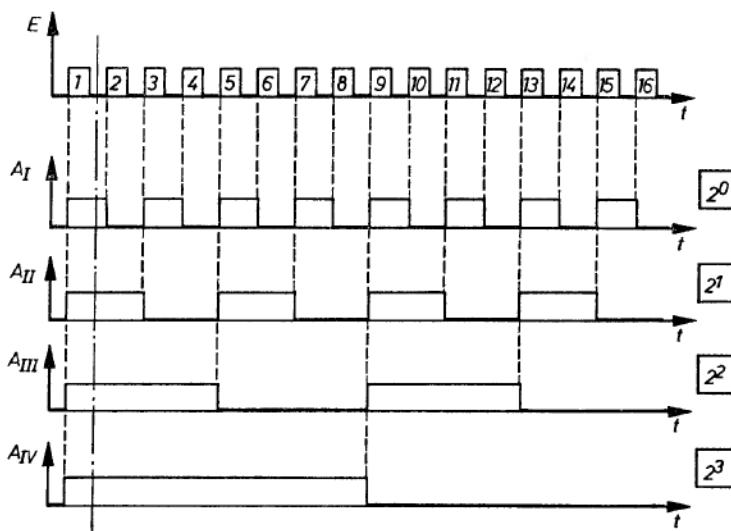
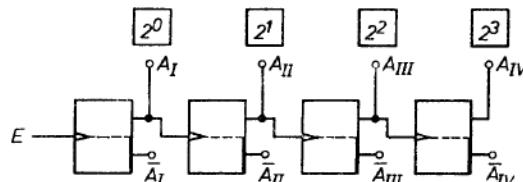
Azt a számlálót, amely nullától a legnagyobb számig számlál, előszám-lálónak nevezük.



36.9. ábra. Átrendezett bináris számláló

36.3. Visszaszámító

Flip-flopokat úgy is összekapcsolhatunk, hogy olyan számláló keletkezzen, amely az első impulzus megjelenésekor a legnagyobbszámot jelzi ki. minden



36.10. ábra. Bináris visszaszámító és impulzusdiagramja

további impulzus eggyel csökkenti a kijelzett számot. Az ilyen számláló visszafelé számlál, és ezért visszaszámlálónak nevezzük.

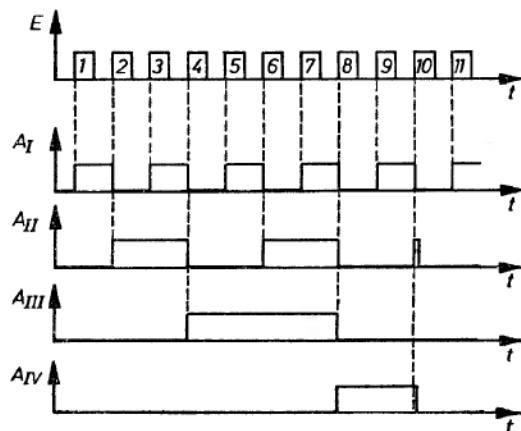
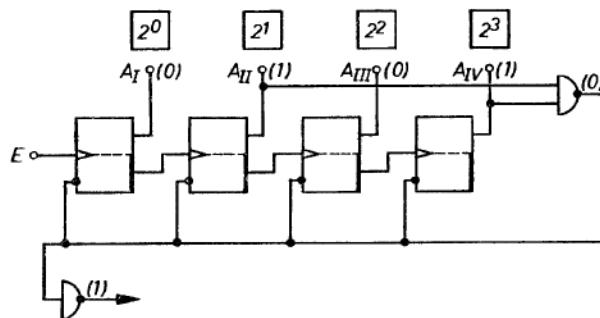
A 36.10. ábra ilyen számlálót ábrázol. Az első impulzus felfutó éle után a bináris $1111 = 15_{10}$ szám jelenik meg. A második impulzus után a kimeneten az $1110 = 14_{10}$ szám van. Ha az A_I , A_{II} és A_{III} kimenetek helyett az \bar{A}_I , \bar{A}_{II} és \bar{A}_{III} kimeneteket használnánk a következő flip-flop vezérlésére, akkor előreszámlálót kapnánk.

Azt a számlálót, amely az 1. impulzus hatására a legnagyobb számot jelzi ki, majd minden további impulzus hatására a számot eggyel csökkenti, visszaszámlálónak nevezzük.

36.4. Dekadikus számlálók

Gyakori igény, hogy a számlálás eredményét decimális számként jelezzük ki. Természetesen bármely bináris szám átkódolható tízes számrendszerbeli számmá. Gyakran választjuk azonban azt az utat, hogy decimális számok körében végezzük a számlálást. Ez az ún. dekadikus számlálókkal valósítható meg.

A dekadikus számláló 4 bites bináris előreszámláló, amely csak bináris $1001 = 9_{10}$ -ig számlál, majd a 10. impulzus megérkezésekor nullára áll.



36.11. ábra. Egy dekadikus számláló és impulzus-diagramja

A dekadikus számlálók felépítéséhez visszaállító bemenettel rendelkező flip-flopra van szükség. A 36.11. ábrán egyszerű dekadikus számláló kapcsolását láthatjuk. A 10. impulzus megérkezésekor az $1010 = 10_{10}$ kimeneti állapotot adódik, amely a NAND kapun keresztül 0 állapotot hoz létre, és ezzel visszaállítja a flip-flopokat. Ugyanakkor a következő dekád bemenetére 1-est küldünk.

A 36.11. ábrán látható áramkör kis szépséghibája, hogy az $1010 = 10_{10}$ szám közvetlenül a visszaállítás előtt rövid ideig a kimeneten van. A kimeneti állapotok decimális számmá való átalakításakor külön kell gondoskodni arról, hogy az $1010 = 10_{10}$ rövid idejű megjelenése hatástalan maradjon.

37. D/A és A/D átalakítók

37.1. Digitális–analóg (D/A) átalakító

A digitális–analóg átalakító, más néven D/A átalakító feladata, hogy a digitális jeleket analóg jelekké alakítsa át.

37.1.1. A digitális–analóg átalakítás elve

Tekintsünk egy szinusztáblázatot! Ez a szinuszértékeket digitális formában tartalmazza. A szinusztáblázat alapján a szinuszgörbe felrajzolható. A táblázat görbüvé való átalakítása digitális–analóg átalakítás. A digitális vezérléstechnikában az információk többsnyire bináris alakban állnak rendelkezésre, valamelyen meghatározott kódban kifejezve. Ezt a kódot a digitális–analóg átalakítónak ismernie kell.

A digitális–analóg átalakító csak meghatározott bináris kód szerinti jeleket képes analóg jelé átalakítani.

Bizonyos bináris kódok nem alkalmasak digitális–analóg átalakításra. Ezek az érték nélküli kódok. Érték nélkülinek nevezik azt a kódot, amelynek elemeihez nincsenek adott számértékek hozzárendelve. A bináris kód pl. értékkel rendelkező kód. minden egyes elemhez, tehát minden helyi értékhez a kettőnek valamelyik hatványa van hozzárendelve. A BCD kód is értékkel rendelkező kód. A Gray-kód ezzel szemben érték nélküli kód, elemeihez konkrét számérték hozzárendelve (l. a 28. fejezetet).

Az érték nélküli kódokat a digitális–analóg átalakítás előtt értékkel rendelkező kódokat kell átalakítani.

Az átalakítás megfelelő kódátalakítókkal nehézség nélkül megoldható. A digitális–analóg átalakítás elvét a 37.1. ábra mutatja. 4 bites egységekkel 16 számérték képezhető. Analóg jelként lépcsős feszültség adódik. Négy bittel tehát 16 különböző amplitúdóérték, 5 bittel értélemszerűen 32, 6 bittel pedig 64 amplitúdófokozat állítható elő. (37.1. táblázat).

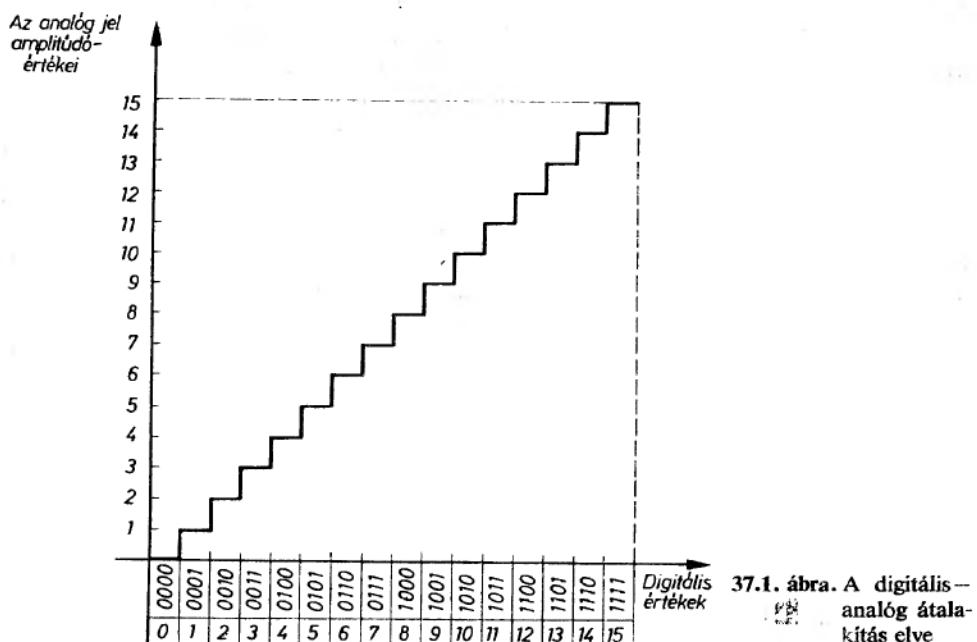
A digitális–analóg átalakításkor adódó analóg jel meghatározott számú amplitúdóértékből álló lépcsős jel.

A lépcsőzés tetszőleges finomságú lehet. Annál finomabb, minél nagyobb a digitális jel bitjeinek a száma.

A lépcsők szűrőtag segítségével elsimíthatók, úgyhogy folyamatos analóg jel keletkezik.

37.1. táblázat. A bitszám és a felbontás közti kapcsolat

A bitek száma	Az amplitúdó-lépcsők száma
4	16
5	32
6	64
7	128
8	256
9	512
10	1 024
11	2 048
12	4 096
13	8 192
14	16 384
15	32 768
16	65 536

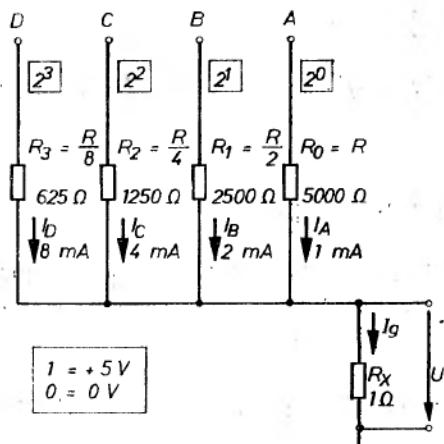


37.1. ábra. A digitális–analóg átalakítás elve

37.1.2. Ellenállás-létrahálózatos D/A átalakító

Az ellenállás-létrahálózatos D/A átalakítók elvi kapcsolását a 37.2. ábra mutatja. Az A, B, C és D bemenetre 4 bites digitális jelet kapcsolunk. Az $R_0 \dots R_3$ ellenállások a kettes számrendszer helyi értékeinek megfelelően méretezendők. Érvényes a következő egyenlet:

$$R_n = \frac{R}{2^n}$$



37.2. ábra. A bináris kódokat ellenállás-létrahálózattal előállító D/A átalakító elvi kapcsolása

R értéke bonyos határok között szabadon megválasztható. Itt 5000Ω -ot választottunk. Az R_1 -re 2500Ω , az R_2 -re 1250Ω adódik. minden további ellenállás éppen félakkora, mint az előző.

A 37.2. táblázat a logikai $1 = +5 \text{ V}$ esetén adódó áramokat tartalmazza. A kimeneti feszültség 1 mV -os lépésekben 0 és 15 mV között változik. A feszültség mindenig annyi mV , amennyi a 4 bites bináris jel értéke.

37.2. táblázat. A 37.2. ábra szerinti D/A átalakító áramai és kimeneti feszültsége különböző bemeneti jelkombinációk esetén

Decimális érték	D 2^3	C 2^2	B 2^1	A 2^0	I_D, mA	I_C, mA	I_B, mA	I_A, mA	I_g, mA	Q_A, mV
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1	1
2	0	0	1	0	0	0	2	0	2	2
3	0	0	1	1	0	0	2	1	3	3
4	0	1	0	0	0	4	0	0	4	4
5	0	1	0	1	0	4	0	1	5	5
6	0	1	1	0	0	4	2	0	6	6
7	0	1	1	1	0	4	2	1	7	7
8	1	0	0	0	8	0	0	0	8	8
9	1	0	0	1	8	0	0	1	9	9
10	1	0	1	0	8	0	2	0	10	10
11	1	0	1	1	8	0	2	1	11	11
12	1	1	0	0	8	4	0	0	12	12
13	1	1	0	1	8	4	0	1	13	13
14	1	1	1	0	8	4	2	0	14	14
15	1	1	1	1	8	4	2	1	15	15

37.2. Analóg—digitális (A/D) átalakító

Az analóg—digitális átalakító, más néven A/D átalakító az analóg jeleket megfelelő digitális jelekké alakítja át.

37.2.1. Az analóg—digitális átalakítás elve

Az analóg jel bizonyos számú amplitúdóértékkel közelíthető. A jel amplitúdóját pl. minden $10\ \mu s$ -ban megmérjük. A mért számértékeket helyes sorrendben egymás után tároljuk. Az összes számérték együtt képezi a digitális jelet.

Valamely időben változó mennyiséget digitális formája számértékek sorozatából áll.

A számértékek tetszőleges szám- vagy kódrendszerben adhatók meg. A 37.1. ábrán ezek decimális és bináris számrendszerben vannak.

Az analóg—digitális átalakítók a számértékeket többnyire kettes számrendszerben vagy BCD kódban adják ki.

37.2.2. Az A/D átalakítók tulajdonságai

Az amplitúdóértékeket meghatározott léptékben adják meg, pl. mV-ban.

Ha pl. feszültségértékeket kell 4 V-ig 1 mV pontossággal átalakítani, akkor 4000 amplitúdófokozat szükséges, amelynek ábrázolásához 12 jegyű bináris számra van szükség. Mindegyik amplitúdóértéket 12 biten ábrázoljuk. A számábrázolás finomsága a bites számától függ. Ezt a tulajdonságot felbontóképességnak nevezzük.

Az A/D átalakítónak annál nagyobb a felbontóképessége, minél több bit áll rendelkezésre a számábrázoláshoz.

A felbontóképességet nem szabad az A/D átalakító pontosságával összetéveszteni. A pontosság a kiadott számértékek helyességétől függ.

A nagy felbontású számértékek, vagyis a sok biten ábrázolt számok is lehetnek pontatlannak.

Minden analóg—digitális átalakító meghatározott pontossággal működik.

A pontosság azt adja meg, hogy az átalakítás eredménye a helyes értéktől felfelé és lefelé mennyivel térhet el. Pl. 10^{-3} pontosság esetén az eredmények 1/1000-del lehetnek nagyobbak vagy kisebbek, mint a helyes érték, azaz $\pm 0,1\%$ eltérés lehet. Nagyobb költségek árán 10^{-5} pontosság is elérhető. Az időben változó analóg jeleket meghatározott időközönként kell mintavételezni. Ez azt jelenti, hogy az amplitúdó-

értékeket minden μ s-ban, 10 μ s-ban vagy 1 ms-ban meg kell mérni és tárolni. Az amplitúdóértékek megállapításának gyakoriságát annál nagyobbra kell venni, minél gyorsabban változik az analóg jel. Általánosságban érvényes:

Az analóg jel mintavételezési gyakoriságának legalább kétszer akkorának kell lennie, mint az analóg jel változásának legnagyobb frekvenciája.

Ha egy 50 Hz...20 kHz tartományba eső analóg hangfrekvenciás jelet kell megfelelő digitális jellé átalakítani, akkor másodpercenként legalább 40 000 mintavétel szükséges.

Az ún. mintavételi frekvencia tehát 40 kHz. Nagyobb lehet, de kisebb nem. Ha mégis kisebb, a frekvenciasáv szűkül. Az analóg–digitális átalakítókat főként integrált áramkörök által állítják elő. A diszkrét alkatrészekből való felépítés igen költséges lenne. Az integrált áramkörös A/D átalakítók között jelenleg a CMOS áramkörök vannak többségben. A TTL és az ECL technológiával készült áramköröket – viszonylag nagy teljesítményigényük miatt – csak ott alkalmazzák, ahol nagy sebességre van szükség. Az A/D átalakítókat lényegében a következő tulajdonságok különöztetik meg egymástól:

felbontóképesség	(bitek száma);
pontosság	(az eredmény %-os hibája vagy a legnagyobb érték %-ában megadott hiba);
sebesség	(az átalakítási folyamat időtartama vagy az időegységenkénti átalakítások száma);
feszültségtartomány	(a legkisebb és a legnagyobb feszültségérték, amelyek között a jelváltozás bekövetkezik.)

Igen sokféle átalakítási eljárást és áramkört alkalmaznak, de ezekkel hely hiányában nem foglalkozunk.

38. Aritmetikai áramkörök

Digitális áramkörökkel számtoni műveletek is végezhetők, pl. összeadás és kivonás. Az ilyenfajta áramköröket aritmetikai áramköröknek nevezzük.

Az aritmetikai áramkörök számítási műveleteknek megfelelő logikai kapcsolatot valósítanak meg a bemeneti változók között.

A bemenetet képező számokat megfelelő bináris kódban kifejezve kell megadni. Az eredményként kapott számok ugyanolyan kódban adódnak.

Minden aritmetikai áramkör csak egyfélé kód, ill. annak megfelelő számrendszer fogadására alkalmas.

Gyakran alkalmazzák a bináris kódot, vagyis a kettes számrendszt és a BCD kódot (l. a 28. fejezetet).

38.1. Félösszeadó

A legegyszerűbb aritmetikai áramkör a félösszeadó.

A félösszeadó két bináris számot tud összeadni.

A következő szabályok érvényesek:

$$0+0 = 0$$

$$0+1 = 1$$

$$1+0 = 1$$

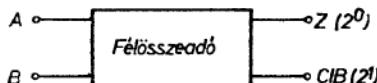
$$1+1 = 10$$

Az egyik összeadandó változó az A, a másik pedig a B. Az áramkörnek kétkimenetűnek kell lennie. Nevezzük a 2^0 értékű kimenetet Z-nek, a 2^1 értékű kimenetet C/B-nek (átvitel). Ha a 0 számjegyet a bináris 0, az 1 számjegyet a bináris 1 állapothoz rendeljük, akkor a 38.1. ábra szerinti igazságtablázat adódik.

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

$$C/B = A \cdot B$$

Az egyenletekből adódó kapcsolást a 38.2. ábra mutatja.



Eset	B	A	C/B	Z
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

38.1. ábra. Félösszeadó és igazságítáblázata

38.2. Teljes összeadó

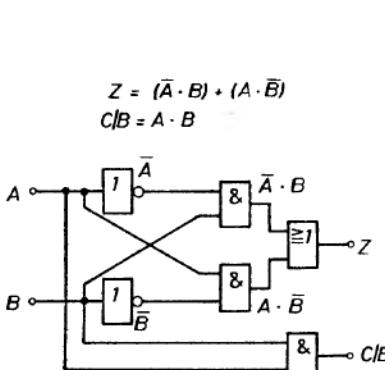
Az összeadó felépítéséhez olyan áramkörökre van szükség, amelyek három bináris számot tudnak összeadni, mivel két bináris szám összeadásakor az előző fokozatban kapott átvitelleket is összegezni kell.

Példa

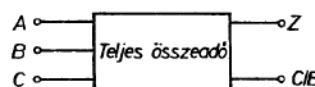
$$\begin{array}{r}
 1 \ 1 \ 1 \ 1 \\
 1 \ 0 \ 1 \ 1 \\
 0 \ 1 \ 1 \ 1 \\
 \hline
 + 1 \ 0 \ 0 \ 1 \ 0
 \end{array} \quad \text{átvitel az alacsonyabb helyi értékről}$$

A teljes összeadó olyan áramkör, amely három bináris számot tud összeadni.

A teljes összeadó áramkör a hálózatsintézis szabályai szerint tervezhető meg. A teljes összeadóhoz három bemenetre van szükség — minden összeadandó bináris számhoz egyre. Ezeket A, B és C jelöli. A kimeneteket a félösszeadóhoz hasonlóan Z-vel és C/B-vel jelöljük (38.3. ábra).



38.2. ábra. Alapkapukból létrehozott félösszeadó



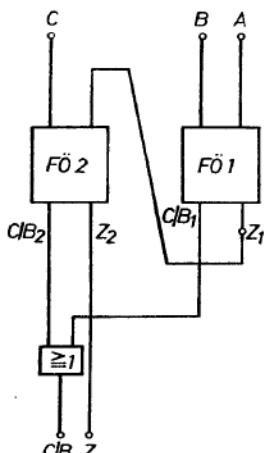
Eset	C	B	A	C/B	Z
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	1	1

38.3. ábra. Teljes összeadó és igazságítáblázata

A teljes összeadó igazságítáblázata az összeadás szabályaiból adódik, és a 38.3. ábrán tüntettük fel.

Az 1. esetben C/B és Z nulla, mivel minden bemenőszám nulla. A 2. esetben a 0+0+1 összeadásból Z = 1 és C/B = 0 adódik. (Ehhez hasonló a helyzet a 3. és 5. esetben is.) A 4. (és 6., ill. 7.) esetben 0+1+1 összeget kell kiszámítani, és ez Z = 0, C/B = 1 eredményre vezet. A 8. eset, az 1+1+1 összegzése, Z = 1 és C/B = 1 értéket eredményez.

A teljes összeadó két félösszeadóból és egy VAGY kapuból is felépíthető (38.4. ábra).



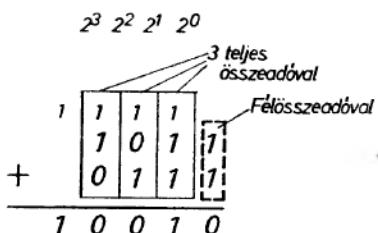
38.4. ábra. Két félösszeadóból felépített teljes összeadó

38.3. Párhuzamos összeadó

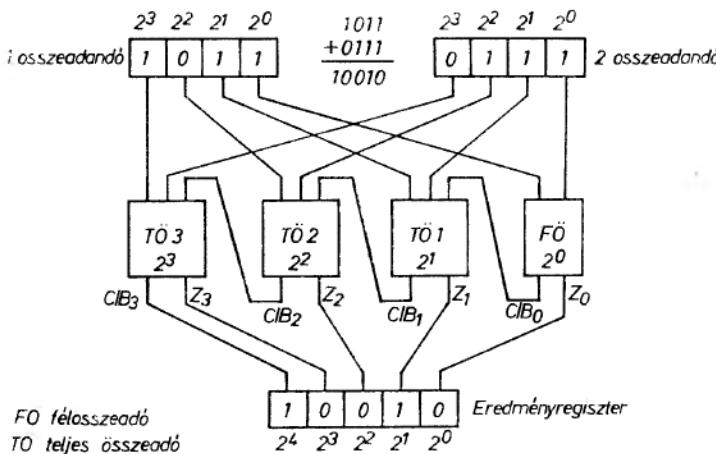
Ha két négyjegyű, kettes számrendszerbeli számot egy lépésekben akarunk összeadni, akkor egy félösszeadóra és három teljes összeadóra van szükség. Jobbról az első, legkisebb helyi értékű oszlop félösszeadóval összegezhető, mivel ide az előző fokozat hiányában nem jöhét átvitel. A többi három oszlophoz viszont jöhét, így ezeknek az oszlopoknak az összeadásához teljes összeadóra van szükség (38.5. ábra).

Az egyetlen lépésekben végzett összeadást párhuzamos összeadásnak nevezzük.

A 38.6. ábra egy 4 bites párhuzamos összeadó kapcsolását mutatja. A FÖ félösszeadó bemeneteire a két összeadandó szám első jobb oldali (2^0 helyi értékű) számjegyét kapcsoljuk. A Z_0 kimenet az eredményregiszterhez kapcsolódik. Az átvitel kimenete pe-



38.5. ábra. Két négyjegyű bináris szám összeadása



38.6. ábra. 4 bites párhuzamos összeadó

dig a TÖ1 teljes összeadó C bemenetével van összekötve, mert ebben az oszlopból az esetleg keletkező átvitel is összegezni kell. A második oszlopból a TÖ1 teljes összeadó a félösszeadóból származó átvitelen kívül megkapja a két összeadandó szám második (a 2^1 helyi értékű) számjegyeit is.

A teljes összeadó Z₁ kimenete az eredmény következő számjegyét szolgáltatja. A C/B₁ átvitel a TÖ2 teljes összeadó bemenetére kerül, ahol a harmadik oszlop (2^2 helyi értékű) jegyeivel adódik össze. A negyedik (2^3 helyi értékű) oszlop összeadására szolgáló TÖ3 teljes összeadó az előzőekhez hasonlóan működik. Ennek a teljes összeadónak az átviteli jele az eredményregiszter 5. pozíciójába kerül.

Két nyolcjegyű bináris szám párhuzamos összeadója egy félösszeadóból és hét teljes összeadóból áll.

38.4. Összeadó-kivonó áramkör

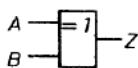
Az előző pontban vizsgált 4 bites párhuzamos összeadó könnyűszerrel átalakítható úgy, hogy összeadásra és kivonásra egyaránt alkalmas legyen.

Összeadásra való használat esetén a következő két intézkedést kell megtenni:

1. A kivonandó regiszter tartalmának negálását el kell hagyni.
2. Az 1-nek a TÖ1 C bemenetén keresztüli hozzáadását nem szabad elvégezni.

Az invertereket kizáró VAGY kapukkal helyettesítjük (38.7. ábra), és a B bemenetüket vezérlésre használjuk. Ha B = 0, nincs negálás, ha B = 1, akkor van. Az így keletkező összeadó-kivonó kapcsolást a 38.8. ábra szemlélteti.

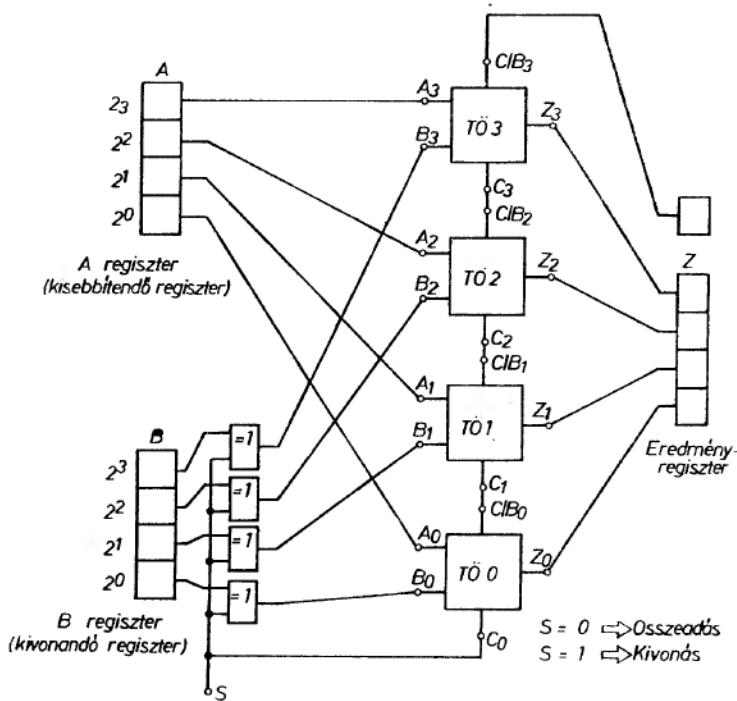
Ha az S vezérlőbemenetre 0 jelet kapcsolunk, akkor a Z = A + B összeadás hajtódiik végre. Ha az S bemenetre 1 jelet kapcsolunk, az áramkör különbségképző áramkörként működve a Z = A - B különbséget fogja képezni. A 4 bites összeadó-kivonó áramkör még univerzálisabbá tehető. Ha az A regiszter kimenete után szintén kizáró VAGY kaput kapcsolunk, akkor megfelelő vezérléssel a B - A különbség is számítható. Ha ezenkívül az A regiszter kimeneteit és a B regiszter kimeneteit ÉS kapukkal kombináljuk, még további lehetőségek is adódnak.



Eset	B	A	Z
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

$B=1$ -nél az
A-t negáljuk

38.7. ábra. Kizáró VAGY kapu rajzjele és igazságáblázata



38.8. ábra. 4 bites összeadó-kivonó áramkör

39. Mikroprocesszorok és mikroszámítógépek

39.1. A mikroprocesszor mint univerzális áramkör

Lehet-e olyan áramkört építeni, amely összeadni, kivonni és szorozni is tud, és ezenkívül a bináris adatok között minden lehetséges logikai kapcsolatot el tud végezni? A bevitt jeleket – adatoknak is nevezik – időben egymás után, meghatározott igény szerint kell feldolgozni. A feldolgozások időbeli sorrendjét, azaz a végrehajtandó logikai műveletek egymásutánját a program rögzíti.

Az ilyen „mindentudó” áramkör univerzálisan alkalmazható lenne és bármiféle logikai áramkört helyettesíthetne. Az elképzelt kapcsolatot megvalósító áramkört nem kellene többé különböző elemekből összeállítani, hanem vennénk az univerzális áramkört, és úgy programozhatnánk, hogy a kívánt kapcsolatot állítsa elő. Az ilyen egyetemes kapcsolás bizonyára elégé bonyolult lenne, az áramkör előállítási költsége pedig valószínűleg magas. A korszerű integrált áramköri technika azonban megadja azt a lehetőséget, hogy bonyolult áramkörököt is olcsón állítsunk elő.

Az ilyenfajta meggondolások indították el azoknak az univerzális áramköröknek a fejlesztését, amelyeket mikroszámítógépeknek nevezünk. A mikroszámítógépek fő része a mikroprocesszor. Napjainkban a különféle típusú mikroprocesszorokat integrált áramkörökkel viszonylag kedvező áron kínálják.

A bonyolult vezérlőkapcsolások, amelyeknek alapkapukból és flip-flopokból való felépítése drága lenne, mikroszámítógépek segítségével igen olcsón létrehozhatók.

39.2. Aritmetikai-logikai egység (ALU)

Az univerzális áramkörök kifejlesztésekor célszerű a 38.8. ábra szerinti 4 bites összeadó-kivonó áramkörből kiindulni, amelyet az előző fejezetben közelebbről is megismertünk.

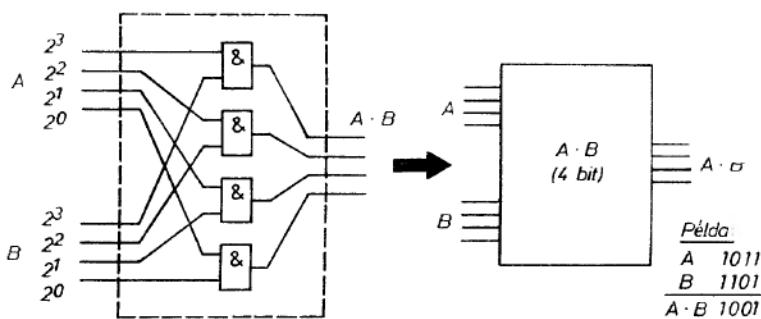
Ezzel az áramkörrel az A és a B bemenőadatok – a vezérléstől függően – összeadhatók és kivonhatók. Ezenkívül szükség van arra, hogy az A és a B jelek között ÉS kapcsolatot,

VAGY kapcsolatot,

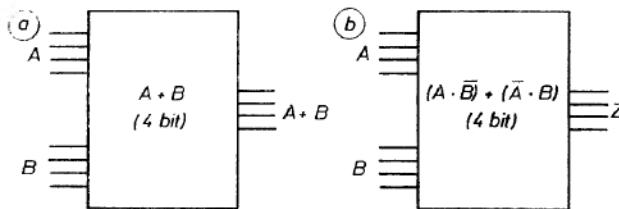
kizáró VAGY kapcsolatot

tudjunk megvalósítani. A 4 bites ÉS kapcsolat előállítására alkalmas áramkört a 39.1. ábrán lájtuk. Ennek megfelelően épül fel a 4 bites, VAGY kapcsolat-előállító áramkör is (39.2. ábra).

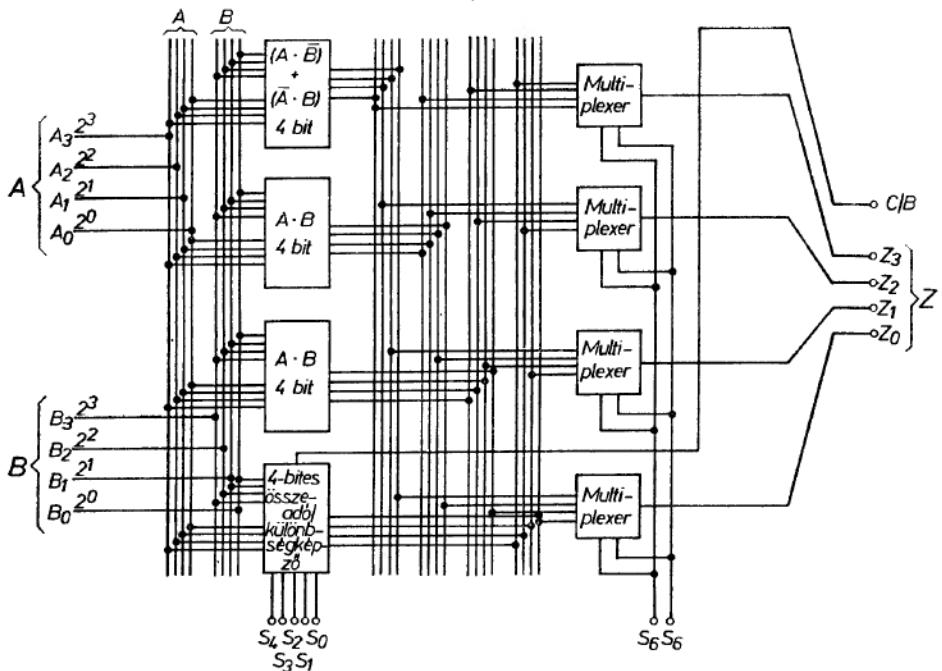
Azt az áramkört, amely két n bites szó között tetszés szerint összeadást, kivonást, ÉS, VAGY, ill. kizáró VAGY műveletet képes elvégezni, aritmetikai-logikai egységnek (ALU) nevezzük.



39.1. ábra. Két 4 bites szó között ÉS kapcsolatot létrehozó kapcsolás



39.2. ábra. a) 4 bites VAGY kapcsolatot előállító áramkör; b) 4 bites kizáró VAGY kapcsolatot előállító áramkör



39.3. ábra. 4 bites aritmetikai-logikai egység telepítése

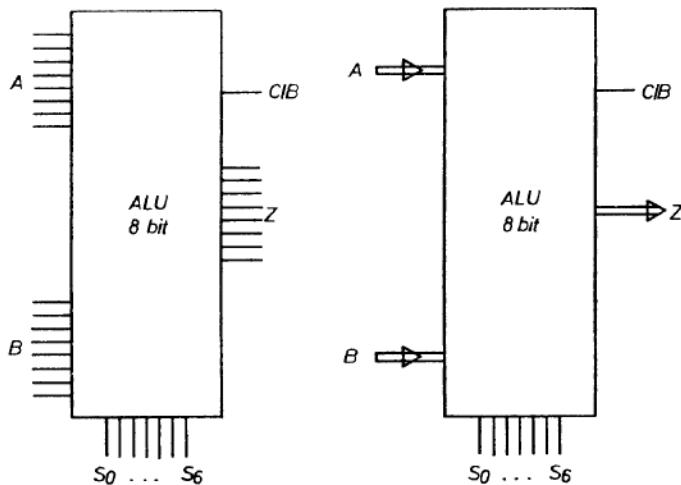
A 4 bit szóhosszúságú ALU tehát a 38.8. ábra szerinti összeadó-kivonó áramkörből, valamint 4 bites ÉS kapcsolatot, 4 bites VAGY kapcsolatot és 4 bites kizáró VAGY kapcsolatot létrehozó áramkörökből áll. A négy 4 bites kimenet bármelyikét négy multiplexerrel keresztül (l. a 34. fejezetet) a 4 bites Z kimenetre kapcsolhatjuk. Az összeadó-kivonó áramkörönek ezenkívül van egy C/B átvitelkimenete is, amelyet kivezetnek (38.3. ábra).

Az aritmetikai-logikai egységeket 4 bites, 8 bites és 16 bites szóhosszúsággal, integrált áramköröként állítják elő.

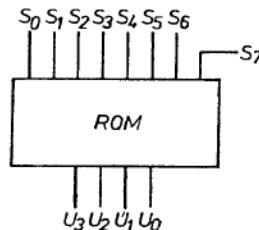
Leggyakrabban a 8 bites ALU-t alkalmazzák, amelynek egyszerűsített ábrázolása a 39.4. ábrán látható.

Mivel a 8 bites ALU alapvetően 8 A bemenettel, 8 B bemenettel és 8 Z kimenettel rendelkezik, a 8 vezetéket egyetlen jelképes vezetékkal ábrázoltuk. A kapcsolási rajzok ezáltal áttekinthetőbbé válnak.

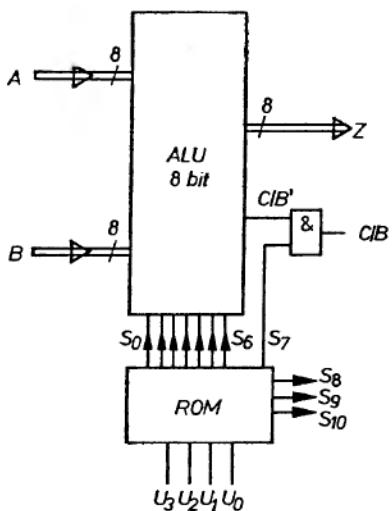
A hét vezérlővonalon ($S_0 \dots S_6$) keresztül összesen $2^7 = 128$ különböző vezérlőutasítás adható be. E vezérlőutasítások közül csak 13-ra van szükség. Célszerű tehát átkódolást alkalmazni. Ezt egy ROM (l. a 35.4. szakaszt) végzi. Négy vezérlőbemenetet alkalmazunk (39.5. ábra), ezzel 16 különböző utasítást tudunk megadni. 3 lehetséges utasítás kihasználatlan marad. Az utasításokat a 39.1. táblázatban foglaltuk össze. Néhány utasítás esetében el kell nyomni az átvitelt. Ezt végzi a ROM S_7 kimenete, amely akkor 0, ha a C/B kimeneten nem jelenik meg vitel. Az S_8 , S_9 és S_{10} kimeneteket további vezérlési funkciók ellátására használják. A 39.6. ábra az átkódoló ROM-mal és átvitelelnyomással rendelkező 8 bites ALU blokksémáját mutatja be.



39.4. ábra. 8 bites ALU sematikus ábrázolása



39.5. ábra. 7 vezérlőbemenetről 4-re átkódoló ROM



39.6. ábra. Átkódoló ROM-mal, ill. ÉS kapus átvitelelnyomással felépített ALU

39.1. táblázat. Az ALU utasításkészlete

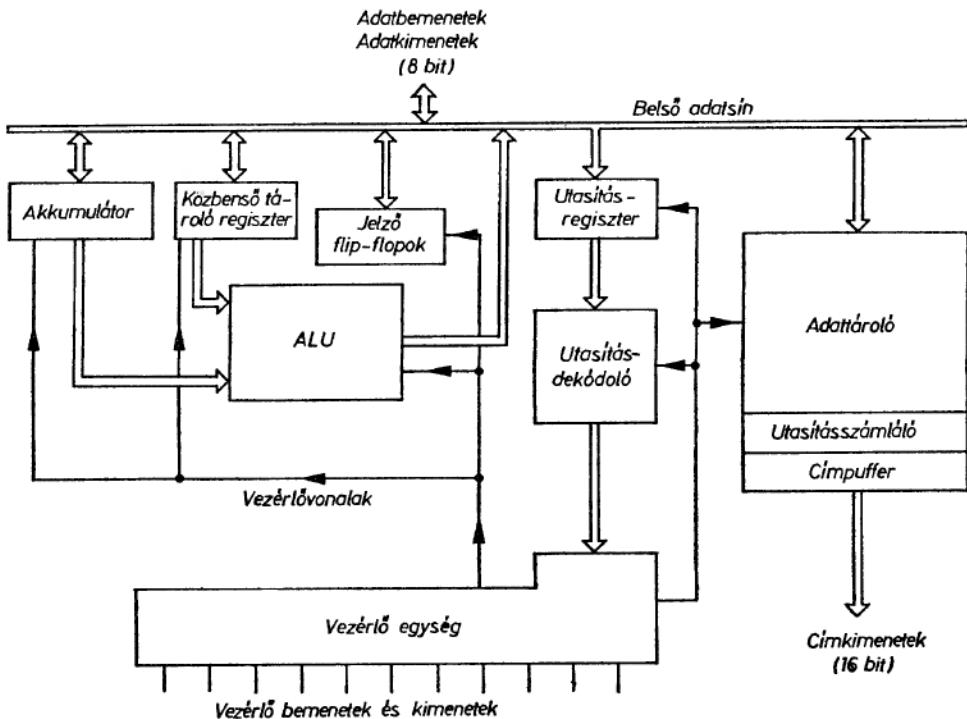
Eset	Utasításkód				Funkció
	U ₃	U ₂	U ₁	U ₀	
1	0	0	0	0	A
2	0	0	0	1	1
3	0	0	1	0	\bar{A}
4	0	0	1	1	B
5	0	1	0	0	0
6	0	1	0	1	A+1
7	0	1	1	0	A-1
8	0	1	1	1	A+B
9	1	0	0	0	A-B
10	1	0	0	1	A \wedge B
11	1	0	1	0	A \vee B
12	1	0	1	1	(A \wedge \bar{B}) \vee (\bar{A} \wedge B)
13	1	1	0	0	-1
14	1	1	0	1	üres kódok
15	1	1	1	0	
16	1	1	1	1	

39.3. Az ALU-tól a mikroprocesszorig

Ahhoz, hogy az aritmetikai-logikai egységekből (ALU-ból) eljussunk a mikroprocesszorig, tovább kell folytatni a kiépítést: adattároló RAM-mal és egy utasítástároló RAM-mal kell ellátni, valamint egy utasításdekódolóval és egy vezérlő áramkörrel. Az utasításdekódoló feladata, hogy a beadott utasításokat vezérlőjelekkel alakitsa át. A vezérlőáramkör kiértékeli ezeket a vezérlőjeleket, és irányítja a feldolgozás menetét. Ezenkívül még egy közbenső tárolóregiszterre is szükség van, amely az adatok és uta-

sítások átmeneti őrzését biztosítja. Mindezeket összeépítve készen is van a mikroprocesszor. minden egyes mikroprocesszornak más és más a belső felépítése, az ún. architektúrája. A SAB 8080A mikroprocesszor egyszerűsített belső felépítését a 39.7. ábra mutatja.

A teljes felépítés a 39.8. ábrán látható. A mikroprocesszor igen bonyolult felépítésű. Az egész áramkör egy kb. 5 mm × 5 mm felületű félvezető lapkán helyezkedik el.



39.7. ábra. A SAB 8080A mikroprocesszor egyszerűsített belső felépítése

39.4. A mikroprocesszor építőelemei

39.4.1. Mikroprocesszortípusok

Mivel a mikroprocesszorfejlesztés irányára többséle, az eredmények is különbözök. Igen sokféle mikroprocesszortípus létezik tehát, amelyek nagyon eltérő tulajdonságúak. Jelenleg kb. 80-féle mikroprocesszortípus (μ P) kapható. Mindegyiket egytokos mikroprocesszorként gyártják, azaz egyetlen integrált áramkörből állnak.

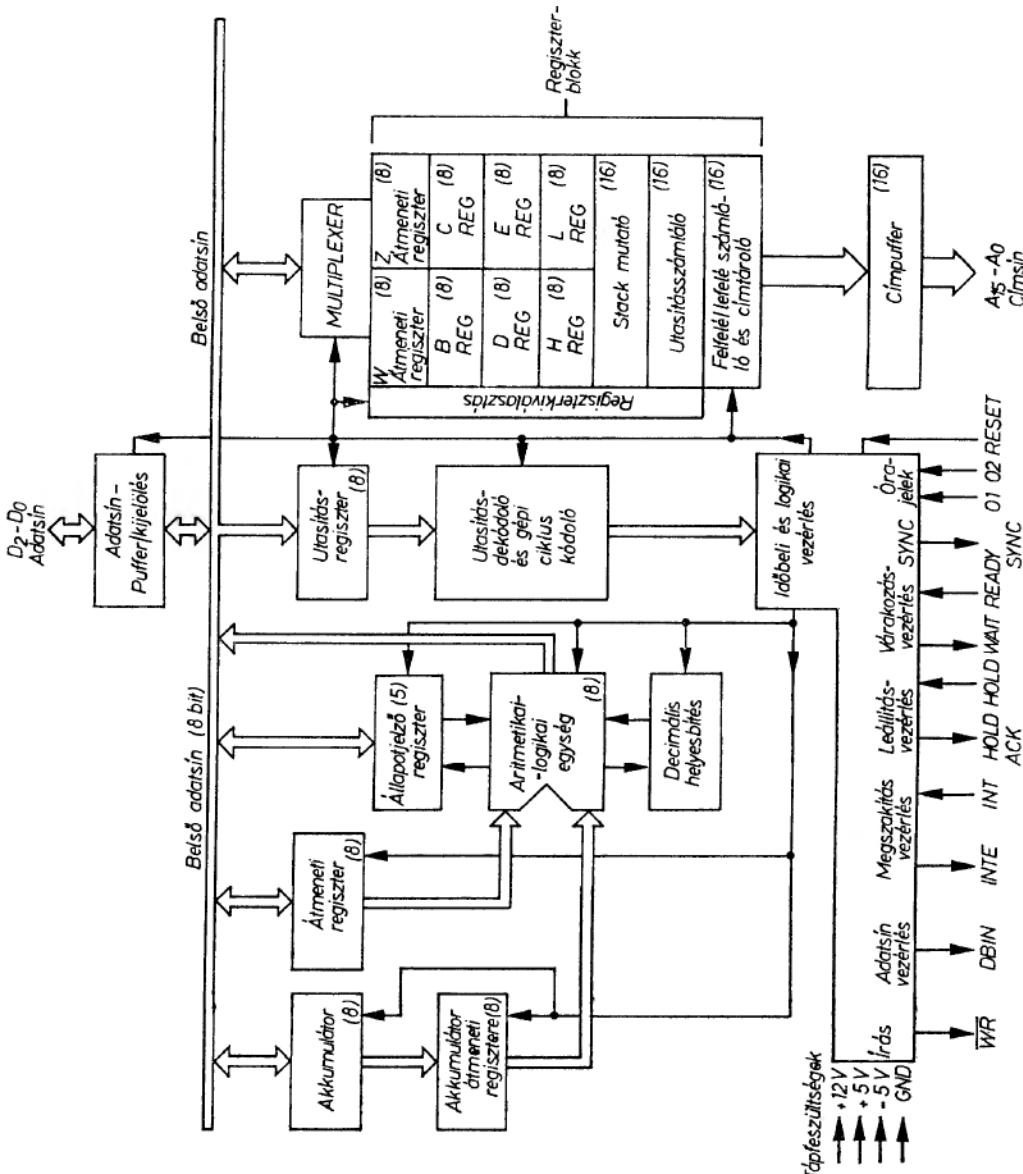
A mikroprocesszorokat kizárolag integrált áramkörként állítják elő.

Rendszerint 40 kivezetéses dual-in-line tokozást alkalmaznak. A mikroprocesszorok elsősorban az alábbi paraméterekben különböznek egymástól:

1. Széhosszúság

A széhosszúság azt adja meg, hogy hány bitet képes a mikroprocesszor párhuzamo-

39.8. ábra. A SAB 8080A mikroprocesszor belső felépítése



san feldolgozni, azaz hogy a bemeneti és a kimeneti mennyiségek hány bitből állnak.

Vannak 4, 8, 16 és 32 bites mikroprocesszorok.

2. Számítási sebesség

Számítási sebesség vizsgálatakor az ún. ciklusidőket hasonlítjuk össze. Ciklusidőn többnyire azt az időt értjük, amely két bináris szám párhuzamos összeadásakor a számok bevitelétől az eredmény kiadásáig eltelik. A ciklusidő szokásos értékei 0,1 μ s-tól 10 μ s-ig terjednek.

3. Technológia (áramkörcsalád)

A mikroprocesszorokat túlnyomó többségben MOS technológiával állítják elő. Ennél a technológiánál érhető el a legnagyobb integrálási sűrűség. Viszonylagボnyolult kapcsolások építhetők fel. Van azonban néhány bipoláris mikroprocesszoris, amelyek a Schottky-TTL áramkörcsaládhoz tartoznak. Működésük gyors, szerkezetük azonban viszonylag egyszerű.

A MOS áramkörcsaládon belül az MOS típusok vannak többségben. Teljesítmény-felvételük kb. 0,5...1,5 W. CMOS típus kevesebb van. Különleges előnyük az igen kis (1...5 mW) teljesítményigény.

Megkülönböztetünk továbbá statikus és dinamikus mikroprocesszorokat. A statikus mikroprocesszoroknak statikus belső RAM-juk van, ami nem igényel frissítést. A dinamikus mikroprocesszorok dinamikus belső RAM-mal rendelkeznek. Ezek frissítőjel hatására őrzik meg tartalmukat.

4. Utasításkészlet

Az utasításkészlet nagysága a mikroprocesszor teljesítőképességének fokmérője (de nem az egyetlen). A teljesítőképesség az utasítás típusától is függ. A sok alkalmasan megválasztott utasítás nagy teljesítőképességet eredményez. A gyártók adatai alapján a lehetséges utasítások száma 48-tól 158-ig terjed.

A 39.2. táblázatban néhány gyakran alkalmazott mikroprocesszor fontosabb tulajdonságait foglaltuk össze.

39.2. táblázat. Néhány fontosabb mikroprocesszor jellemzői

Típus	A gyártó cégek	Szóhossz, bit	Az utasítások száma	Ciklusidő (kb.), μ s	Technológia
4040	Intel	4	60	10,0	NMOS
3850 (F8)	Fairchild	8	72	2,0	NMOS
8080A	Intel/Siemens	8	78	2,0	NMOS
8085	Intel/Siemens	8	80	1,3	NMOS
IM 6100	Intersil	8	87	2,5	CMOS
M 6800	Motorola	8	72	2,0	NMOS
SCMP	National Semic.	8	46	2,0	PMOS/NMOS
Z80	Zilog	8	158	1,0	NMOS
8086	Intel	16	135	0,5	NMOS
68000	Motorola	16	56	0,6	CMOS/NMOS
TMS 9900	Texas Instr.	16	69	7,5	NMOS
80386	Intel	32	150	0,3	CMOS
68020	Motorola	32	100	0,4	CMOS/NMOS
32332	National Semic.	32	130	0,3	NMOS

39.4.2. SAB 8080A típusú mikroprocesszor

A 8080A típusjelű mikroprocesszort már eddig is igen nagy számban alkalmazták, és mára egyfajta „szabvány” mikroprocesszorrá lépett elő. Többek között az Intel és a Siemens gyártja. A Siemens által használt típusjel a SAB 8080A.

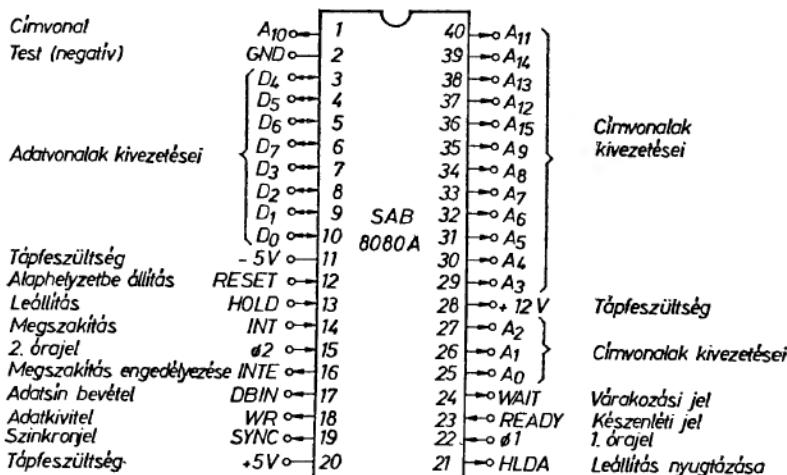
A SAB 8080A NMOS technikával készült 8 bites mikroprocesszor, amelynek ciklusideje 2 µs. Utasításkészlete 78 utasítást foglal magában. A mikroprocesszor TTL-kompatibilis, tri-state sínrendszerű (azaz az adatbemenetek és -kimenetek, valamint a címkimenetek az L és H szinteken kívül egy nagy ellenállású – lekapcsolt – állapotot is fel tudnak venni).

A SAB 8080A mikroprocesszorban dinamikus RAM van, amelyet frissíteni kell. A frissítés órajelre következik be. A SAB 8080A mikroprocesszort 40 kivezetéssel dual-in-line tokban szállítják, és lábkiosztását a 39.9. ábra mutatja. A 3–10. kivezetés a 8 bites adatsín. Ez az adatsín kétféle irányban használhatnak.

A mikroprocesszornak 16 címvonalra van, ezzel $2^{16} = 65\,536$ különböző cím választható ki. A mikroprocesszor tehát 64 Kbyte-os külső tárral képes együttműködni. minden egyes címmel 1 byte = 8 bit címzhető. A bemeneti és kimeneti egységek vezérléséhez további címek állnak rendelkezésre.

Egy 64 Kbyte-os tároló helyett több kisebb tároló is csatlakoztatható, és ezek minden egyike RAM, ROM vagy PROM lehet.

Címkimenet az 1., 25., 26., 27. és a 29–40. láb (39.9. ábra). Ezeket a lábakat címsínnel kell összekötni.



39.9. ábra. A SAB 8080A mikroprocesszor (Siemens) lábkiosztása

A további kivezetések a tápfeszültség csatlakoztatására és a mikroprocesszor vezérlésére szolgálnak. A bekötendő tápfeszültségek a +12 V (28. láb), +5 V (20. láb), -5 V (11. láb).

A földet a 2. lábra kell bekötni. A mikroprocesszorhoz külső óragenerátorra van szükség, amely két különböző órajelet állít elő (az 1. órajelet a 22., a 2. órajelet a 15. lábra kell kötni).

A RESET bemenet (12. lab) az utasításszámlálót nullára állítja. A HOLD (13. lab) csatlakozásra adott L szintű jellet a mikroprocesszor működése felfügeszthető. Ezen idő alatt adatbevitel és -kivitel hajtható végre

Az INT (14. lab) és az INTE (16. lab) csatlakozások programmegszakításra és a magszakítás engedélyezésére szolgálnak.

A DBIN kivezetésen levő jel azt közli, hogy az adatsínen adatbevitel folyik, azaz a mikroprocesszor adatok átvitelére képes. A WR kivezetésen addig van L szint, amíg külső tároló felé irányuló adatkivitel van folyamatban. A SYNC kivezetés szinkronjelet szolgáltat, ha új műveleti ciklus kezdődik.

Különösen fontosak a WAIT (várakozás, 24. lab) és a READY (kész, 23. lab) csatlakozások. A WAIT csatlakozáson H szint van, ha a mikroprocesszor várakozik. A READY csatlakozáson levő H szint pedig azt jelzi, hogy az adatsínen átadásra kész adatok vannak. Ilyenkor a mikroprocesszor rövid időre megállítható, és az adatok átvihetők.

A HLDA (21. lab) csatlakozáson keresztül a mikroprocesszor a HOLD jelre válaszként nyugtájójelet ad ki. Azt jelzi, hogy az adat- és címsín nagy ellenállású állapotba kapcsol.

A SAB 8080A belső felépítése kissé szövevényes, ezért két lépésben tárgyaljuk. A belső felépítés egyszerűsített vázlatán (39.7. ábra) az ALU könnyen felismerhető. Az A bemenetek visszavezetése a belső adatsínen keresztül történik. A belső adatsínen azok az adatok is áthaladnak, amelyeket az adattárolókba írunk és amelyeket onnan kiolvasunk.

A utasításokat az adatbemeneteken keresztül visszük be; a belső adattárákban átmenetileg tárolhatók az adatok. Külön belső programtároló nincs, de külső programtároló csatlakoztattható. Az utasítások az utasításregiszteren keresztül az utasítás-dekódolóba jutnak, és onnan mint vezérlőjelek adódnak át a vezérlő áramkörnek. A vezérlő áramkör néhány vezérlőbemeneten és -kimeneten keresztül látja el feladatait. A külső egységek (pl. RAM és ROM) vezérléséhez szükséges címeket az utasítászámláló és a címpufferek állítják elő. A címek 16 bites szavakból állnak, és ezeket a címsíne vezetik.

A gyártó a SAB 8080A mikroprocesszor teljes belső felépítését megadja a gyártmányismertetőin (39.8. ábra). A mikroprocesszor ezek szerint további egységekkel egészül ki, mint pl. az adatok és címek pufferregiszterei. (A puffer közbenső tároló.) Az adattár sok regiszterből áll, amelyek multiplexeren keresztül kiválaszthatók.

Szinte hihetetlen, hogy ez az igen bonyolult áramkör egyetlen, $5 \text{ mm} \times 5 \text{ mm}$ méretű szilíciumlapkán helyezkedik el. És bámulatos, hogy ma már milyen olcsón megvásárolható. Egyedül a mikroprocesszorral nem építhető fel vezérlő áramkör. További külső egységekre van szükség, pl. óragenerátorra, bemeneti-kimeneti egységre és egy vagy több program- és adattárolóra, amely lehet ROM, PROM, EPROM vagy RAM. Természetesen tápegységre is szükség van.

39.5. A mikroprocesszorok kiegészítő egységei

A mikroprocesszorok működését órajel vezérli. Legalább egy órajelre szükség van, gyakran azonban két különbözőre.

Kevés mikroprocesszornak van saját belső óragenerátora. Ezekben a mikroprocesszorokban az óragenerátort a mikroprocesszor IC-je tartalmazza. Más típusok esetén kiegészítő egység kell, amely előállítja az igényelt órajelet.

A SAB 8080A mikroprocesszorhoz a SAB 8224 óraáramkört ajánlják kiegészítő egysékként. Ezt az egységet egy 16 kivezetéses dual-in-line tokban szállítják. Az egység egy kvarcoszcillátort foglal magában, amelynek frekvenciáját kívülről csatlakoztatott kvarckristály határozza meg. A kvarc sajátfrekvenciája tág határok között megválasztható, de ez befolyásolja a mikroprocesszor műveleti sebességét. A kvarckristály szokásos frekvenciája kb. 18 MHz. A kvarcoszcillátor rezgését négy szögrezgessé alakítják át, amit azután 9-cel osztanak. Így adódik a 2 MHz-es működési frekvencia. Az adatbevitel és -kivitel számára ugyancsak kiegészítő egységek szükségesek. Ezeket I/O egységeknek (*input/output* egységeknek) nevezünk. Ezek az adatokat közbenső tárolókon (puffer) keresztül veszik át. Az I/O egység adatbevitel esetén vezérlőjelet állít elő, amelynek hatására a mikroprocesszor felfüggesszi a működését mindenkor, amíg az adatbevitel le nem zajlik. Az I/O egységek gyakran tartalmaznak egységki-választó áramkört. Ha a mikroprocesszorból kell adatokat kivinni, akkor az I/O egység kezdeményezi a kívánt adatkivitelt. A SAB 8080A mikroprocesszorhoz különféle I/O egységek kapcsolhatók. Gyakran alkalmazzák a SAB 8212 típusjelű egységet. A tárolóegységek igen fontos kiegészítő elemek. Alkalmazhatók RAM, ROM és PROM egységek, valamint EPROM egységek is, és a megadott maximális tárolókapacitáson belül tetszőlegesen kombinálhatók. A SAB 8080A mikroprocesszor megengedett összes tárolókapacitása 64 Kbyte, mivel a mikroprocesszor összesen 65 636 különböző címet tud kezelni.

Azért olyan fontosak a tárolóegységek, mert a mikroprocesszorok belső tárolókapacitása viszonylag kicsi, ezért a programokat és az adatokat külső tárolóban kell elhelyezni.

Külső tárolóként mágnesszalagos egységeket (pl. kazettás magnetofon) és mágneslemezes egységeket is használhatunk. Az ilyen eszközök csatlakoztatásához illesztőegységekre van szükség, amelyek az adatokat meghatározott módon átalakítják, valamint vezérlőjeleket képesek értelmezni és előállítani. Pl. a mikroprocesszor 8 bites párhuzamos kimenetén kiadott adatokat a mágneslemezes tároló számára alkalmas adatformátumúvá alakítják át.

Az I/O egységek, a tároló- és illesztőegységek számára vezérlőjelek kellenek, amelyeket a mikroprocesszor egyáltalán nem, vagy csak tökéletlenül tud előállítani. Emiatt sok esetben további kiegészítő áramkörök rendszervezérő egység is szükséges. Ez az egység gondoskodik minden olyan jelről, ami a kiegészítő egységek közvetlen összeköttetéséhez szükséges. A SAB 8228 rendszervezérő egységet a SAB 8080A mikroprocesszorhoz fejlesztették ki. A rendszervezérő az adatsín 8 bites, kétirányú sínmeghajtóját is magában foglalja.

A vezérlő rendszer szolgáltatja az összes szükséges vezérlőjelet, azonkívül a program-megszakítások egyszerű lekezeléséhez és a több-byte-os utasítások alkalmazásához szükséges egyéb vezérlőjeleket is. A több-byte-os utasítások két vagy több byte-ból állnak.

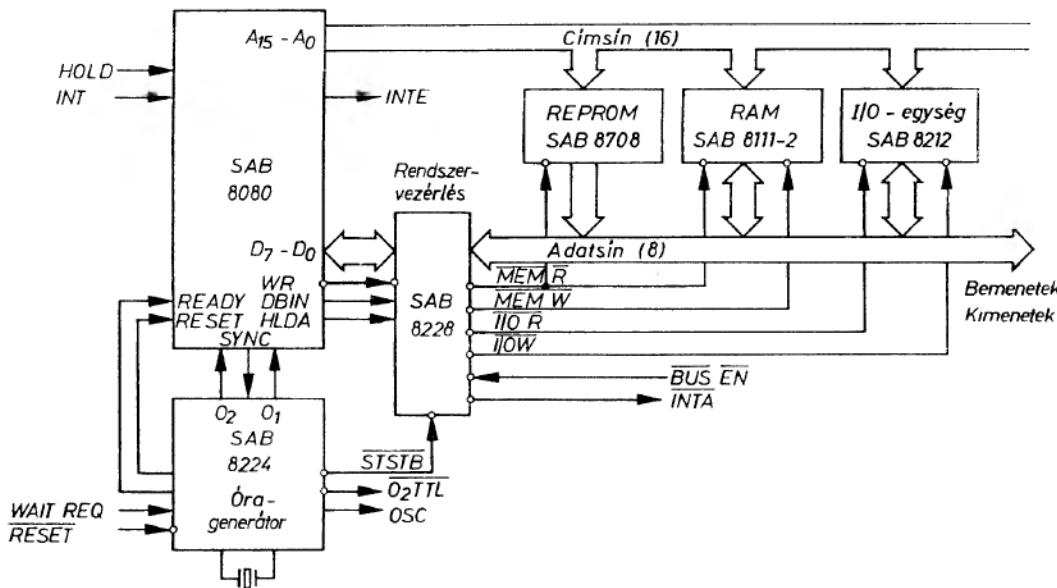
39.6. Mikroszámitógép

Ha a mikroprocesszort a szükséges kiegészítő egységekkel összekapcsoljuk, mikroszámitógépet kapunk. Néhány kiegészítő egység – mint pl. az óragenerátor és a tároló – feltétlen szükséges. A további egységeket a teljesíteni kívánt feladat szerint választják meg.

A mikroszámítógép mikroprocesszorból és kiegészítő egységekből álló, működőképes vezérlőegység.

A mikroszámítógépeket többnyire egyetlen áramköri lapon építik fel. Egy ilyen áramköri lap pl. a következő egységeket tartalmazhatja:

mikroprocesszor	(SAB 8080A)
óragenerátor	(SAB 8224)
EPROM	(SAB 8708)
RAM	(SAB 8111-2)
I/O egység	(SAB 8212)
rendszervezérlő egység	(SAB 8228)
az óragenerátor kvarckristálya	



39.10. ábra. Mikroszámítógép felépítése

A 39.10. ábra egy ilyen mikroszámítógép vázlatát adja meg. A SAB 8080A-ból kiinduló adatsín áthalad a SAB 8228 rendszervezérő egységen. Itt erősödnek fel az érkező és távozó adatok. Az adatsín 8 bites, kétirányú sín.

A címsín 16 bites, egyirányú sín. A címek mindenkor a mikroprocesszorból érkeznek. Az adatsínhez és a címsínhez csatlakoznak az I/O egységek és a tárolóegységek. A vezérlést a SAB 8228 egység látja el. Az egymás után végrehajtandó utasításoknak – azaz a programnak – az EPROM-ban kell lenniük. A szükséges adatokat kívülről olvassuk be a RAM-ba. Most következhetnek a vezérlési és számítási műveletek. Az eredményeket ismét a RAM-ban tároljuk, de kívánság szerint külső tárra is kiírhatók. A mikroszámítógép programozását meglehetősen nehéz megtanulni. Sajnos minden mikroprocesszortípusnak kicsit más utasításai vannak.

Kezdetben ezért célszerű egy konkrét mikroprocesszortípus programozásába belemályogni, és csak ezzel az egy mikroprocesszorral foglalkozni. Ha egy mikroprocesszor-

típus utasításkészletét már alaposan elsajátítottuk, viszonylag könnyű egy másikra átállni. A gyártók a programozás megtanulásához segítséget nyújtanak. Vannak olyan gyakorló-mikroszámítógépek is, amelyeken a programozást lépésről lépésre, magunktól megtanulhatjuk. Tanfolyamon való részvétel ajánlatos.

Mikroprocesszorokból és kiegészítő elemekből sokféle mikroszámítógép állítható össze. Különféle kiegészítő egységeket választhatunk ki és kombinálhatjuk őket különböző típusú és kapacitású tárolókkal azért, hogy az adott feladatot optimálisan oldjuk meg.

A gyártó által kínált fejlesztőszerek lényegesen leegyszerűsítik a fejlesztési munkát. A fejlesztési feladat egyszerű megoldását adják az egyetemes mikroszámítógépek, amelyek egyetlen félvezető lapkára vannak integrálva. Mivel az egyetemes mikroszámítógép minden szükségeset tartalmaz, a rendszerfelépítéssel és a kiegészítő egységekkel nem kell foglalkoznunk. Mégis mit tartalmaz egy ilyen mikroszámítógép valójában? Mindenekelőtt milyen a tárolója és annak mekkora a tárolókapacitása?

Ha megvizsgálunk egy piacon kapható egyetemes mikroszámítógépet – pl. a Texas Instruments TMS 1000 gépet –, megállapíthatjuk, hogy a tárolókapacitás elég kicsi. A ma elérhető maximális elemsűrűség nem tesz lehetővé nagyobb tárkapacitást. A mikroprocesszor is sokszor csupán 4 bites kivitelben készül.

Az egyetemes mikroszámítógép jelenleg csak egyszerűbb vezérlési feladatokra alkalmas.

Az egyetemes mikroszámítógép speciális feladatokhoz nehezen illeszthető.

Az elérhető integráltsági fok azonban várhatóan növekedni fog. Ez azt jelenti, hogy az egyetemes mikroszámítógépek a jövőben bonyolultabbak lehetnek és nagyobb lehet a tárkapacitásuk is. Ezzel együtt a jelentőségük is nőni fog.

148 Ft