

f o k	b i q	Kritikus csillap.		Bessel		Butterworth	
		a	b	a	b	a	b
1	1	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
2	1	1.2872	0.4142	1.3617	0.6180	1.4142	1.0000
3	1	0.5098	0.0000	0.7560	0.0000	1.0000	0.0000
3	2	1.0197	0.2599	0.9996	0.4772	1.0000	1.0000
4	1	0.8700	0.1892	1.3397	0.4889	1.8478	1.0000
4	2	0.8700	0.1892	0.7743	0.3890	0.7654	1.0000
5	1	0.3856	0.0000	0.6656	0.0000	1.0000	0.0000
5	2	0.7712	0.1487	1.1402	0.4128	1.6180	1.0000
5	3	0.7712	0.1487	0.6216	0.3245	0.6180	1.0000
6	1	0.6999	0.1225	1.2217	0.3887	1.9319	1.0000
6	2	0.6999	0.1225	0.9686	0.3505	1.4142	1.0000
6	3	0.6999	0.1225	0.5131	0.2756	0.5176	1.0000
8	1	0.6017	0.0905	1.1112	0.3162	1.9616	1.0000
8	2	0.6017	0.0905	0.9754	0.2979	1.6629	1.0000
8	3	0.6017	0.0905	0.7202	0.2621	1.1111	1.0000
8	4	0.6017	0.0905	0.3728	0.2087	0.3902	1.0000

Analóg szűrők

Irodalom: Tietze-Schenk: Analóg és digitális áramkörök.
Műszaki Könyvkiadó

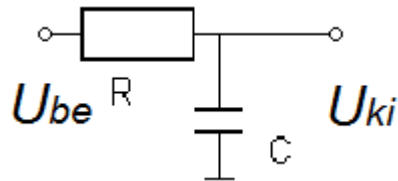
Fontosabb szűrőtípusok:

- aluláteresztő (LP: low pass)
- feluláteresztő (HP: high pass)
- sávszűrő (BP: band pass)
- sávzáró szűrő (BS: band stop)
- mindent áteresztő (AP: all pass)

Aluláteresztő szűrők:

Elméleti alapok:

A leggyakoribb aluláteresztő szűrő:



$$A = \frac{U_{ki}}{U_{be}} = \frac{1}{1 + j\omega RC} = \frac{1}{1 + P}$$

-20 db/dekád, ha meredekebb kell, többet sorba kapcsolunk.
Ilyenkor:

$$A = \frac{1}{f(P)}, \text{ ha } P \gg 1, \text{ akkor } A \approx \frac{1}{P^n}$$

Azaz $n \times -20\text{db/dekád}$ a kapott meredekség

Az n . fokú szűrő általános alakja:

$$A(P) = \frac{A_0}{1 + c_1 P + c_2 P^2 + \dots + c_n P^n}$$

c_1 - c_n pozitív, valós együtthatók. A szűrő fokszáma megegyezik P legnagyobb hatványkitevőjével, és c értékétől függ a szűrő típusa.

A nevező polinomját tényezőkre bontjuk. Ha komplex pólusok is vannak, akkor nem lehet elsőfokú tényezőkre bontani, másodfokú tagok is szerepelnek a tényezők között:

$$A(P) = \frac{A_0}{(1 + a_1 P + b_1 P^2)(1 + a_2 P + b_2 P^2) \dots}$$

Minden szűrő felépíthető másodfokú szűrők sorba kapcsolásával.

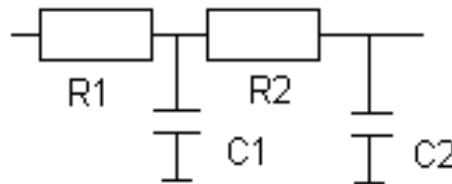
Aluláteresztő szűrő típusok:

Az $a_1..a_n$ és a $b_1..b_n$ együtthatók megválasztásával különböző tulajdonságú szűrők hozhatók létre

1. Kritikus csillapítású szűrő:

- a nevező nem tartalmaz komplex gyököket
- RC hálózatokkal realizálható
- ugrásfüggvény bemenet esetén túllövés nem keletkezik

Másodrendű RC szűrő:

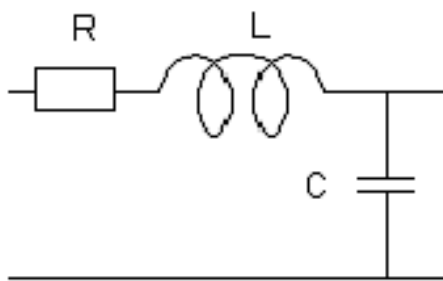


$$A(p) = \frac{1}{1 + (R_1 C_1 + R_2 C_2 + R_1 C_2)p + R_1 C_1 R_2 C_2 p^2}$$

2. Bessel típusú szűrő:

Cél a lehető legjobb impulzusátvitel

- frekvenciától független futási idő
- frekvenciával arányos fázistolás (az áteresztő tartományban)
- megvalósítása RLC áramkört igényel



$$A(p) = \frac{1}{1 + RCp + LCp^2}$$

3. Butterworth típusú szűrő

Cél a maximális laposságú átvitel az áteresztő tartományban

Hogyan érhető ez el?

Az N-ed fokú aluláteresztő szűrő frekvenciamenetének abszolút érték négyzete:

$$|A|^2 = \frac{A_0^2}{1 + A_2\Omega^2 + A_4\Omega^4 + \dots + A_{2n}\Omega^{2n}}$$

Áteresztő tartományban $\Omega < 1$, tehát akkor teljesül a max. lapos menet, ha Ω -nak csak legmagasabb hatványa szerepel. E szerint:

$$|A|^2 = \frac{A_0^2}{1 + A_{2n}\Omega^{2n}}$$

$\Omega=1$ -nél az átvitel 3 dB-el csökken. Ebből A_{2n} :

$$|A|^2 = \frac{A_0^2}{2} = \frac{A_0^2}{1 + A_{2n}}, \quad \text{tehát} \quad A_{2n} = 1$$

Ez alapján a frekvenciamenet abszolút értékének négyzete:

$$|A(\Omega)|^2 = \frac{A_0^2}{1 + \Omega^{2n}}$$

Butterworth szűrő tulajdonságai:

- maximális laposságú átvitel az áteresztő tartományban
- záró tartományban jó csillapítás a max. laposság miatt
- egységugrásra jelentős a túllövés

Például az erősítés négyzete másodfokú szűrő esetében:

$$A(\Omega) = \frac{1}{1 + aP + bP^2} \quad (P = j\Omega), \quad A(\Omega) = \frac{1}{1 + aj\Omega - b\Omega^2}$$

$$|A(\Omega)|^2 = \frac{1}{1 + (a^2 - 2b)\Omega^2 + b^2\Omega^4}$$

Max. lapos akkor lesz, ha $a = \sqrt{2}$ és $b = 1$ ($\Omega < 1$)

4. Csebisev típusú szűrő:

Az áteresztő tartományban megengedünk adott mértékű, szokásosan 0.5, 1 vagy 3 dB értékű ingadozást

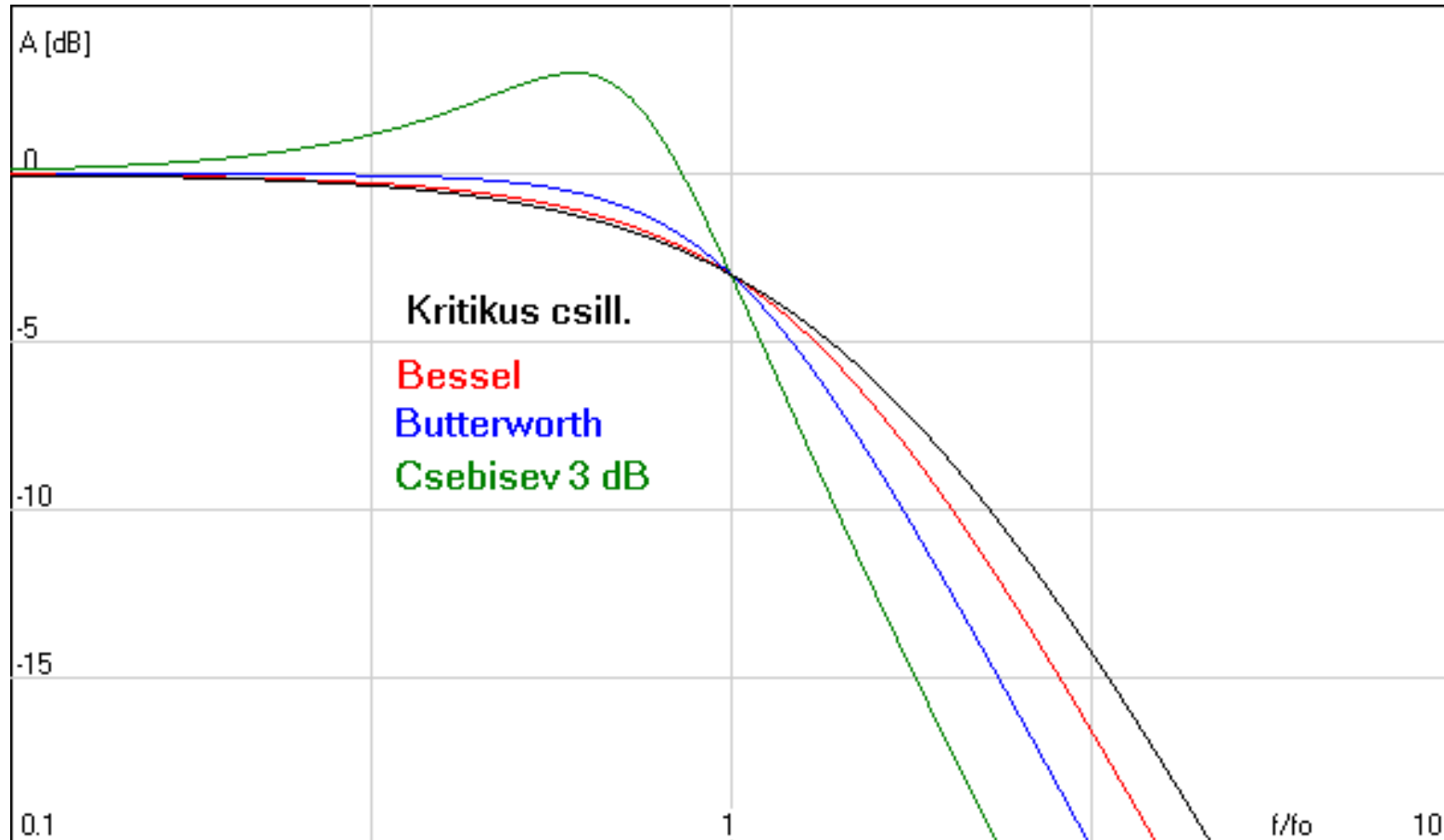
- záró tartományban a legnagyobb csillapítást adja
- egységugrásra igen jelentős a túllövés
- pontos mérőáramkörökben nem használható az átvitel ingadozás miatt

Az ingadozás miatt fellépő hiba mértéke:

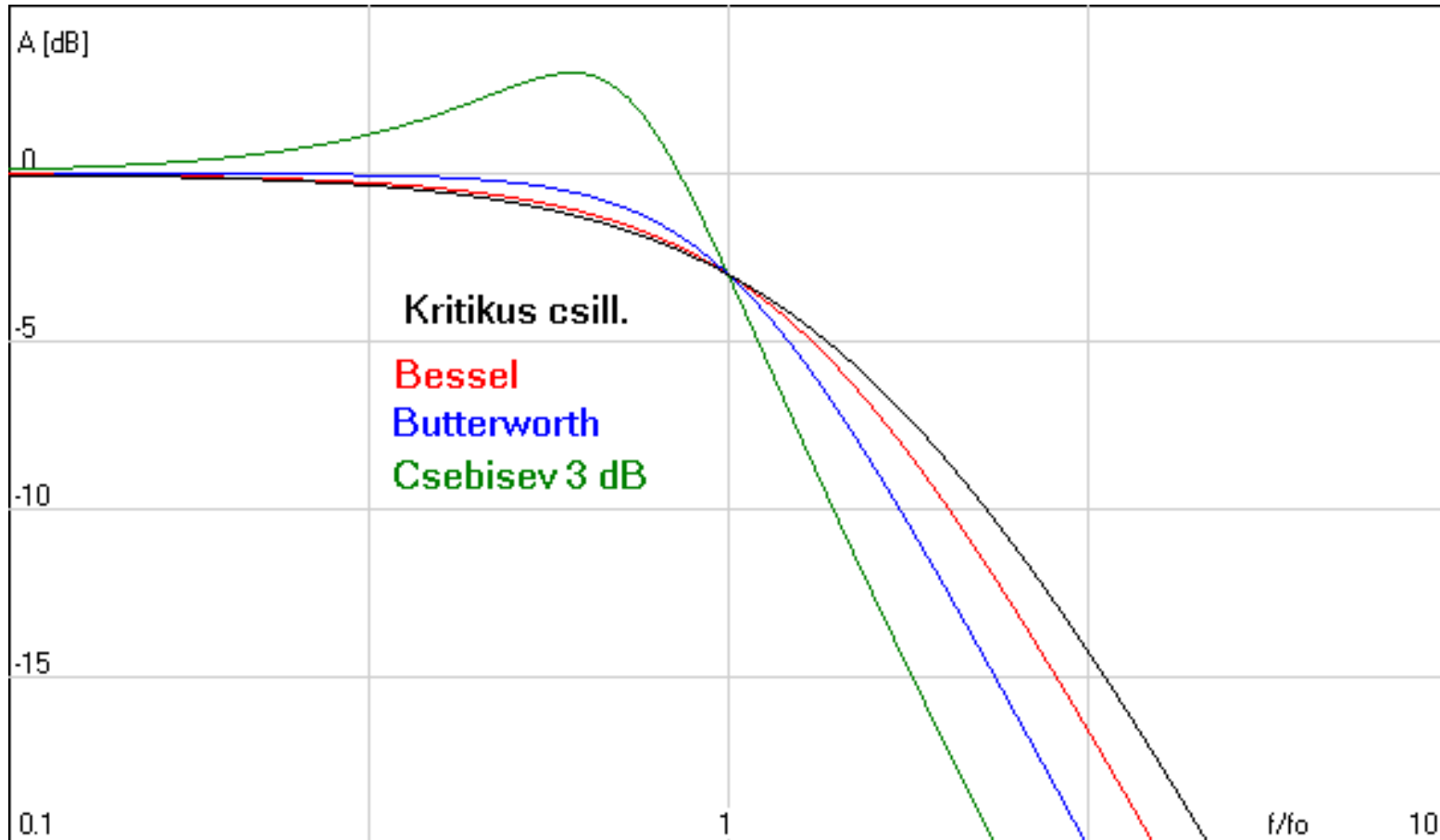
dB	hiba %
3	41.4
1	12.2
0.5	5.9

Hangtechnikai áramkörökben kedvelt, szűrő ott ez nem probléma.
Mi nem fogjuk használni.

Különböző típusú 2. rendű szűrők átviteli görbéje

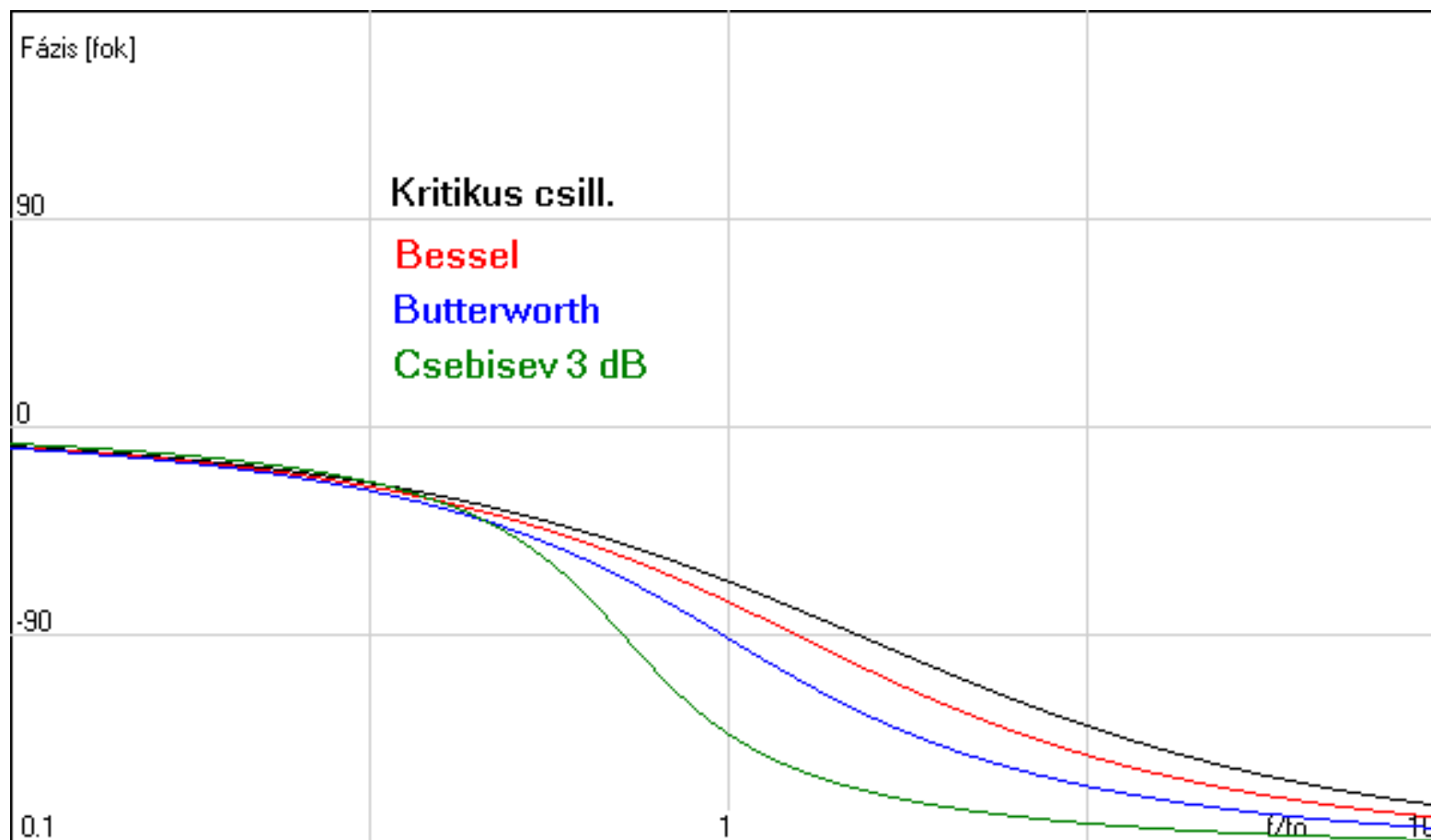


Különböző típusú 2. rendű szűrők átviteli görbéje



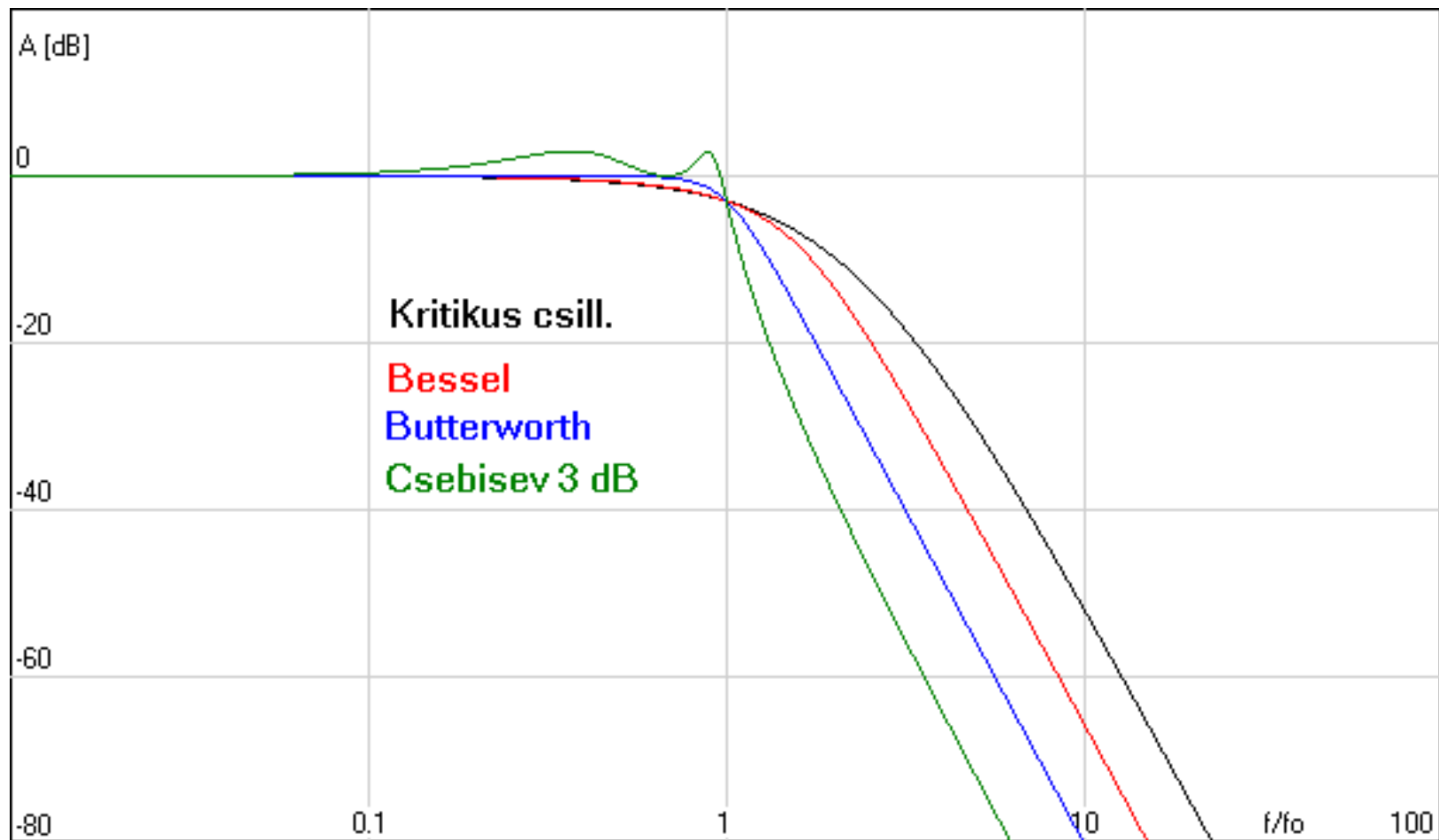
Magasabb frekvencián mindhárom görbe párhuzamos, mert természetesen azonos a meredekségük -40 dB/dek.

Különböző típusú 2. rendű szűrők fázis görbéje

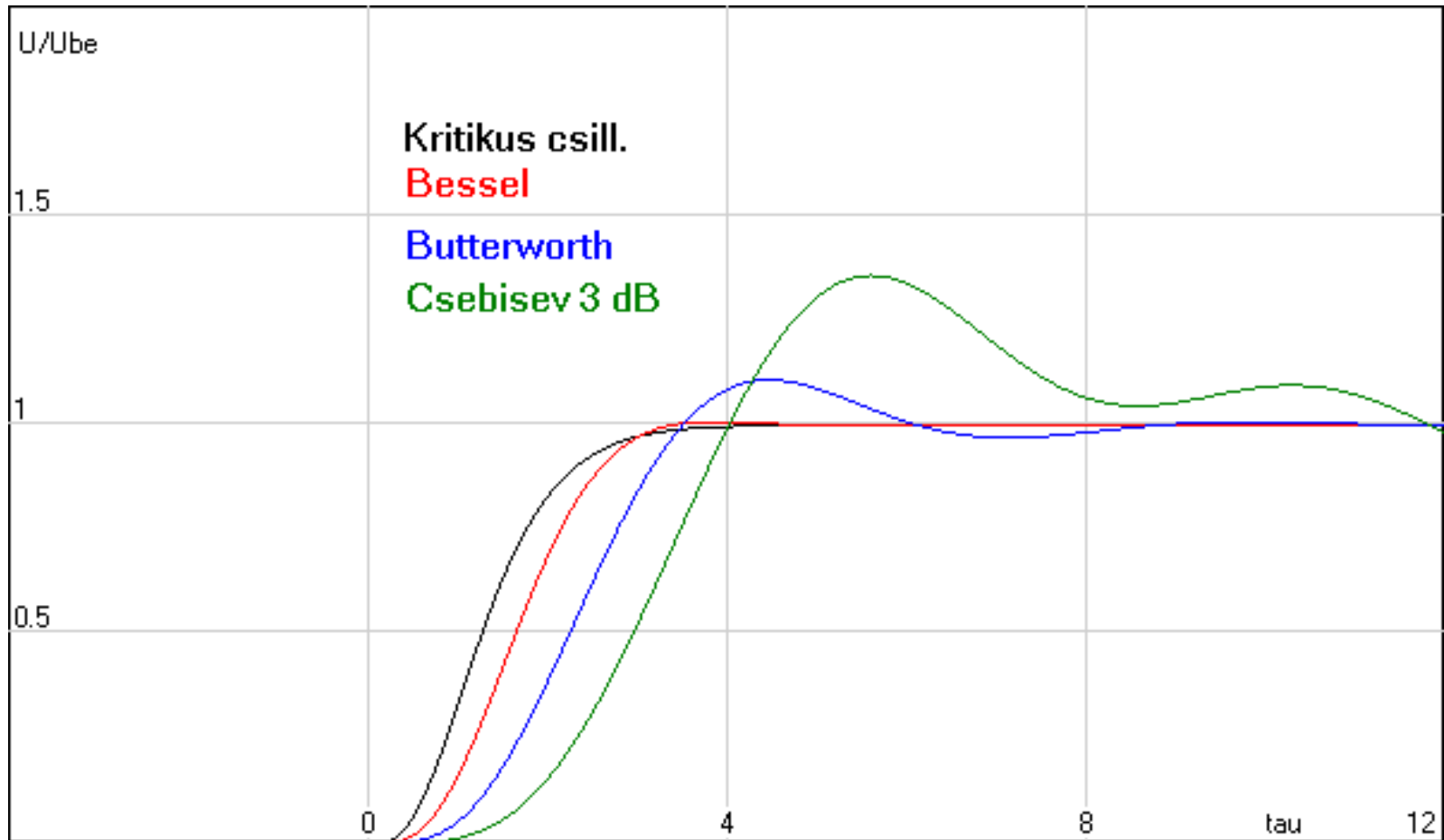


Természetesen mindegyik a -180° -hoz tart.

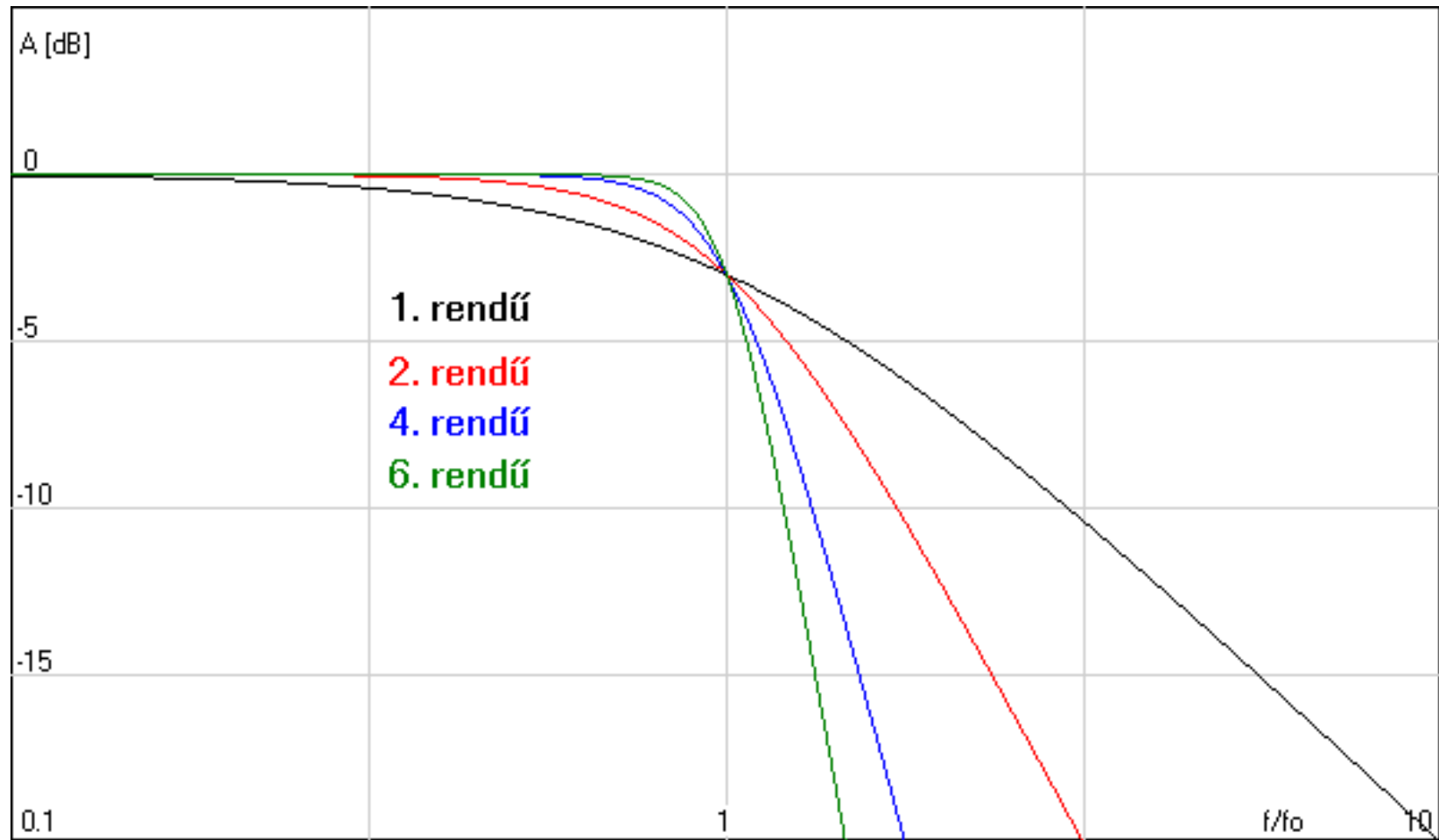
Különböző típusú 4. rendű szűrők átviteli görbéje



Különböző típusú 4. rendű szűrők tranziens válasza

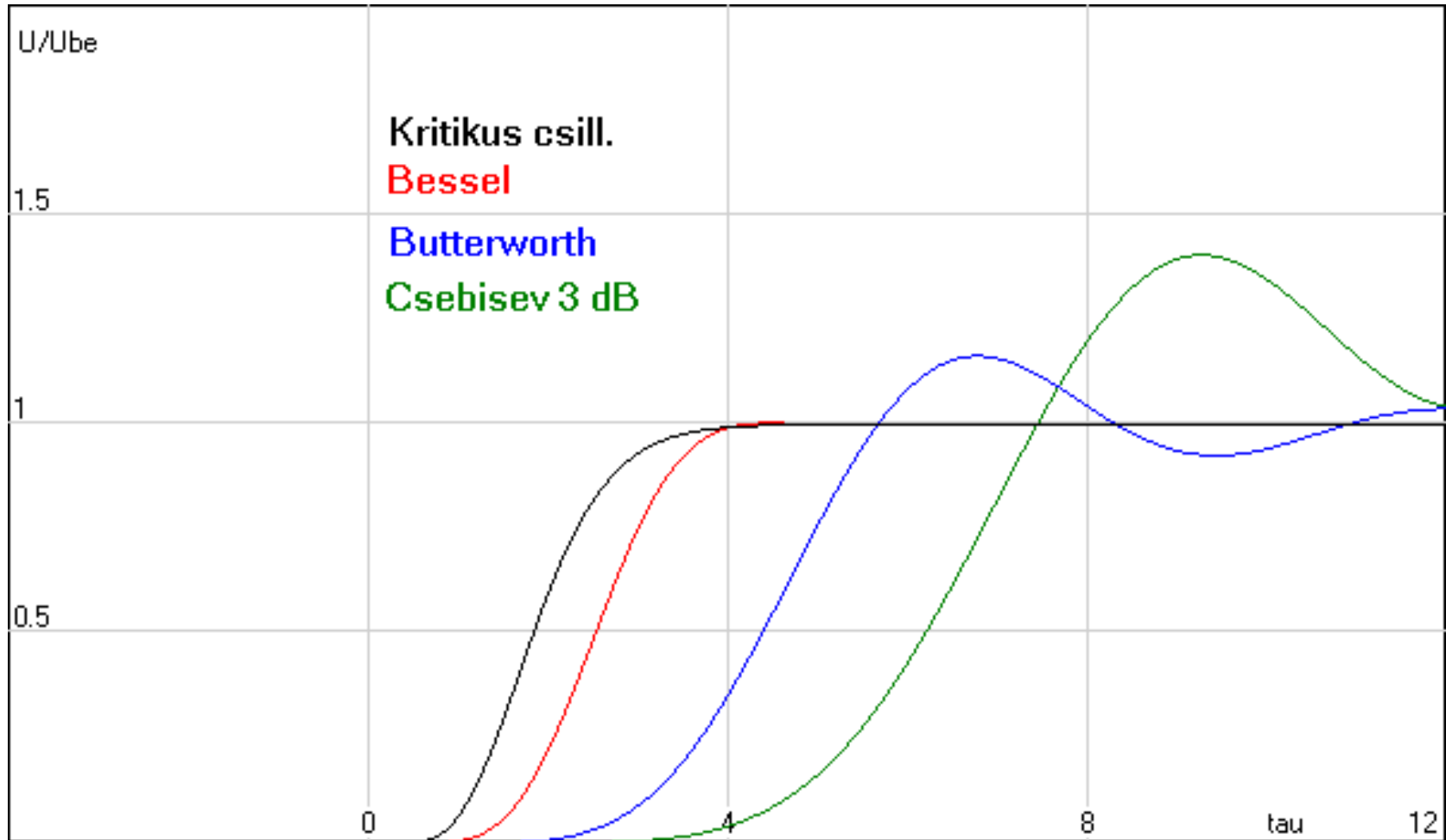


Különböző fokszámú Butterworth szűrők átviteli görbéje



$$|A|^2 = \frac{1}{1 + \Omega^{2n}}$$

Különböző típusú 8. rendű szűrők tranziens válaszai



Mindezek alapján hogyan válasszunk szűrőket?

Aluláteresztő szűrők együtthatói

fokszám	biquad	Kritikus csill.		Bessel		Butterworth	
		a	b	a	b	a	b
1	1	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
2	1	1.2872	0.4142	1.3617	0.6180	1.4142	1.0000
3	1	0.5098	0.0000	0.7560	0.0000	1.0000	0.0000
3	2	1.0197	0.2599	0.9996	0.4772	1.0000	1.0000
4	1	0.8700	0.1892	1.3397	0.4889	1.8478	1.0000
4	2	0.8700	0.1892	0.7743	0.3890	0.7654	1.0000
5	1	0.3856	0.0000	0.6656	0.0000	1.0000	0.0000
5	2	0.7712	0.1487	1.1402	0.4128	1.6180	1.0000
5	3	0.7712	0.1487	0.6216	0.3245	0.6180	1.0000
6	1	0.6999	0.1225	1.2217	0.3887	1.9319	1.0000
6	2	0.6999	0.1225	0.9686	0.3505	1.4142	1.0000
6	3	0.6999	0.1225	0.5131	0.2756	0.5176	1.0000
8	1	0.6017	0.0905	1.1112	0.3162	1.9616	1.0000
8	2	0.6017	0.0905	0.9754	0.2979	1.6629	1.0000
8	3	0.6017	0.0905	0.7202	0.2621	1.1111	1.0000
8	4	0.6017	0.0905	0.3728	0.2087	0.3902	1.0000

Példák

Minden szűrő felépíthető másodfokú szűrők sorba kapcsolásával.

$$A(P) = \frac{A_0}{(1 + a_1P + b_1P^2)(1 + a_2P + b_2P^2)\dots}$$

Harmadrendű Bessel típusú aluláteresztő szűrő:

$$A(P) = \frac{1}{(1 + 0.756P)(1 + 0.9996P + 0.4772P^2)}$$

Negyedrendű Butterworth típusú aluláteresztő szűrő:

$$A(P) = \frac{1}{(1 + 1.8478P + P^2)(1 + 0.7654P + P^2)}$$

Felüláteresztő szűrők

Előállítható aluláteresztő szűrőből $P=1/P$ helyettesítéssel.
Másodrendű felüláteresztő szűrő átviteli függvénye:

$$A(P) = \frac{1}{1 + \frac{a_1}{P} + \frac{b_1}{P^2}} = \frac{P^2}{b_1 + a_1 P + P^2}$$

Mindent áteresztő szűrő

- frekvenciától független állandó erősítés
- a számláló a nevező komplex konjugáltja
- fázisforgatás frekvenciafüggő
- alkalmazása: fáziskorrekció
- elsőrendű szűrő karakterisztika:

$$A(P) = \frac{1 - P}{1 + P}$$

- másodrendű szűrő karakterisztika:
- az együtthatókat célszerű a Bessel típusból választani

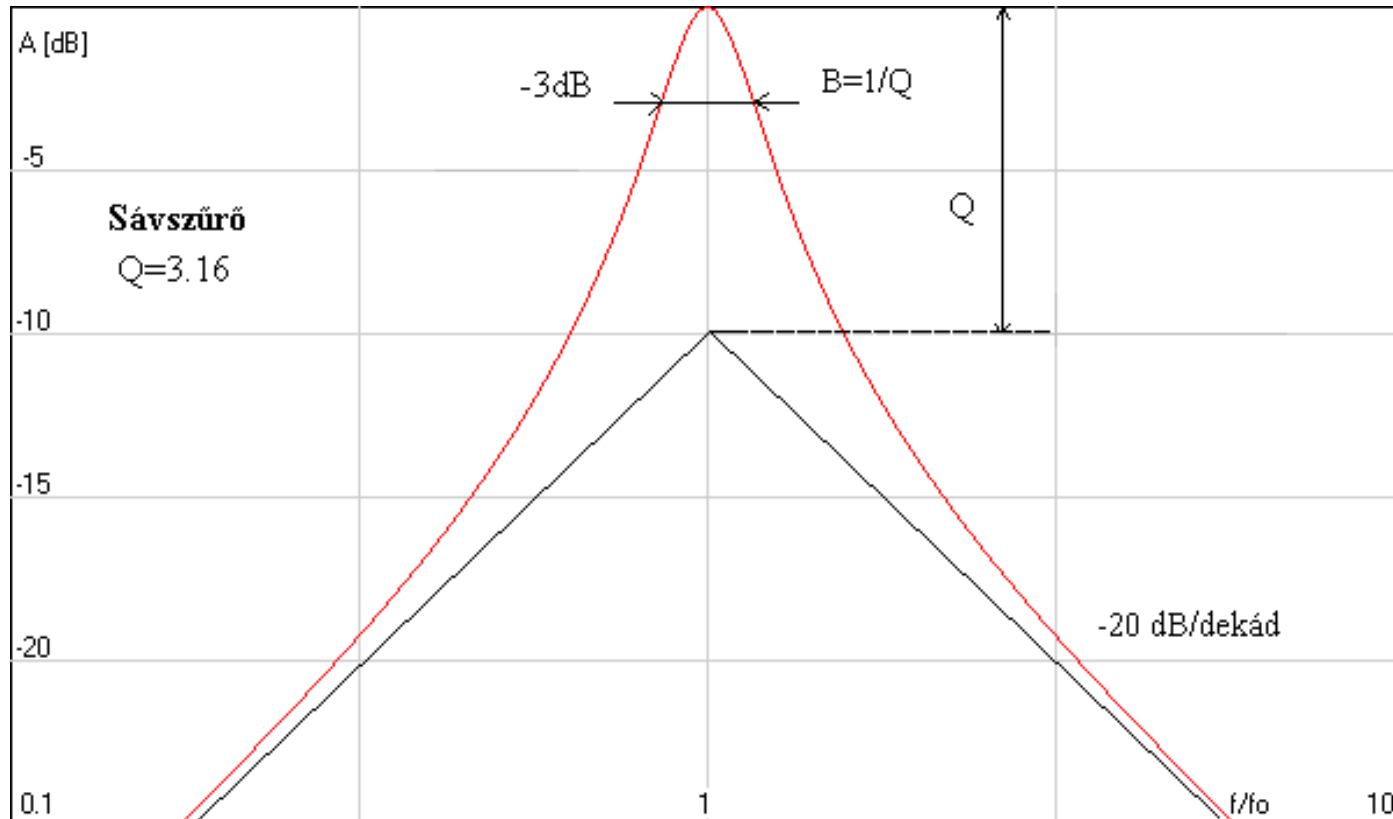
$$A(P) = \frac{1 - a_1P + b_1P^2}{1 + a_1P + b_1P^2}$$

Sávszűrők

- egy adott frekvenciasávot enged át
- a sávszélesség a jósági tényezőtől függ
- átviteli karakterisztika (Q =jósági tényező):

$$A(P) = \frac{\frac{1}{Q} P}{1 + \frac{1}{Q} P + P^2}$$

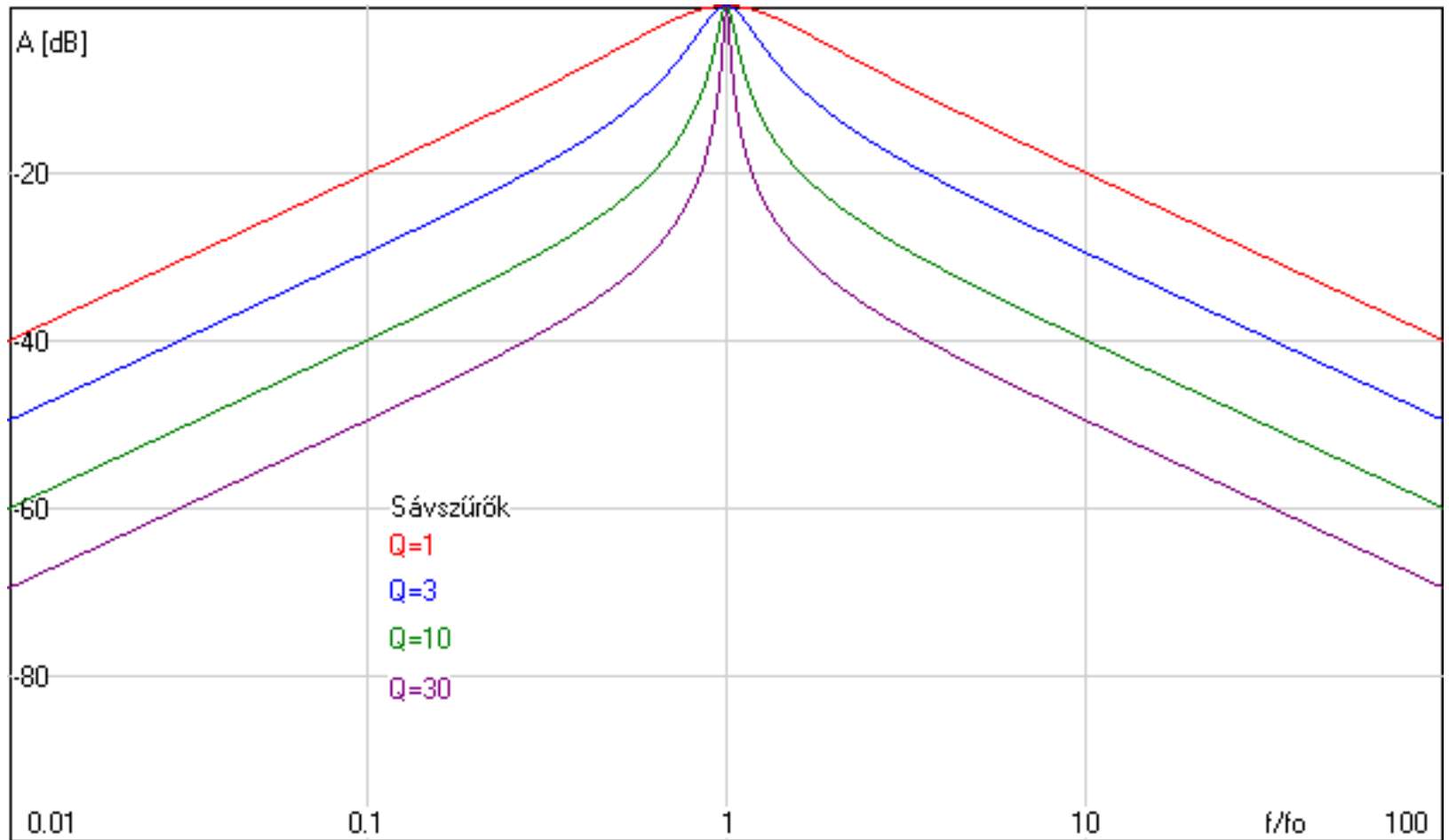
$Q=3.16$ jósági tényezőjű sávszűrő karakterisztika



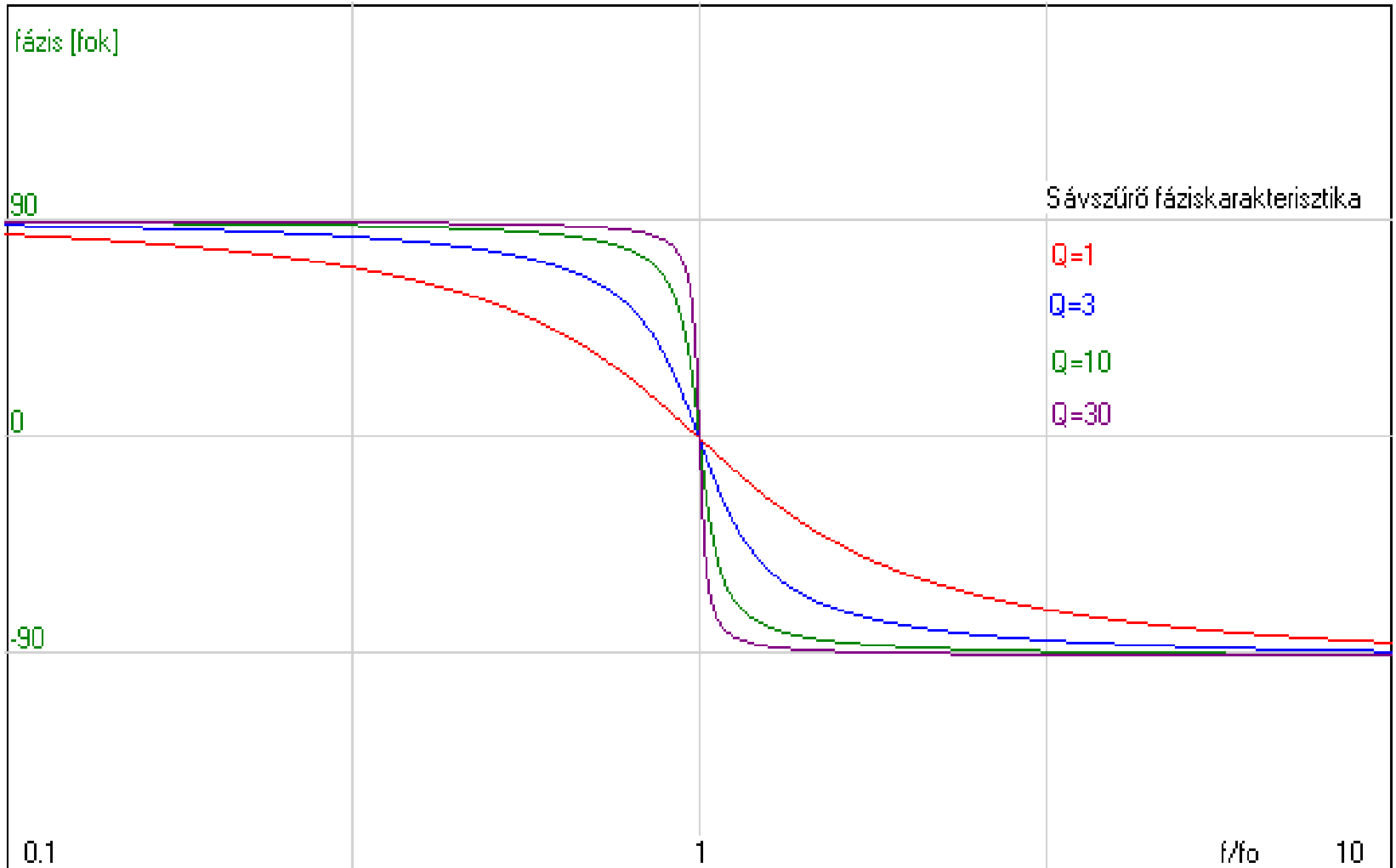
- sáv szélesség $B=1/Q$
- oldalsáv meredekség 20 dB/dekád
- csillapítás a záró tartományban $1/|A| = Q*\Omega$ (ha $\Omega \gg 1$)
 $1/|A| = Q*1/\Omega$ (ha $\Omega \ll 1$)
- csillapítás az áteresztő tartományban ($\Delta\Omega < B/2$) :

$$A(\Omega) = \frac{1}{1 + j \frac{\Delta\Omega}{B/2}}$$

Sávszűrők amplitúdó görbéi



Sávszűrők fázis görbéi



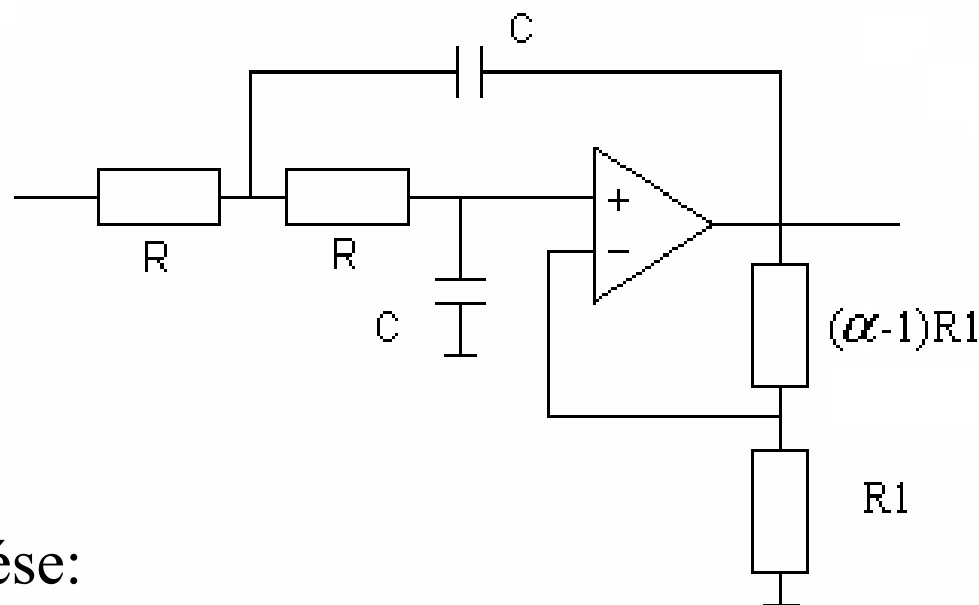
Sávszáró szűrők

- egy adott frekvenciasávot nem enged át
- a sávszélesség a jósági tényezőtől függ
- átviteli karakterisztika (Q=jósági tényező):

$$A(P) = \frac{1 + P^2}{1 + \frac{1}{Q} P + P^2}$$

Analóg szűrők megvalósítása

2. rendű aluláteresztő szűrő:



A kapcsolat erősítése:

$$A(P) = \frac{\alpha}{1 + P\omega_h RC(3 - \alpha) + P^2\omega_h^2 R^2 C^2}$$

A szűrőparaméterek felhasználásával adódik:

$$f_h = \frac{\sqrt{b_1}}{2\pi RC} \quad \alpha = A_0 = 3 - \frac{a_1}{\sqrt{b_1}}$$

Az α erősítés értéke csak a_1 és b_1 -től függ, és f_h nem befolyásolja.

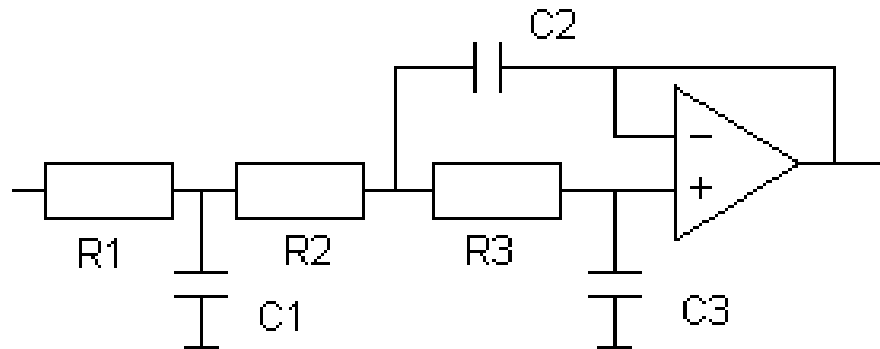
α Értéke meghatározza a szűrő karakterisztika jellegét:

α értéke	típus
1.000	Kritikus csillapítás
1.268	Bessel
1.586	Butterworth
2.234	Csebisev 3 dB
3.000	Oszcillátor

3. rendű aluláteresztő szűrő:

Egy erősítő fokozattal megvalósítható.

A méretezési eljárás bonyolult.



Alkatrész értékek 100 Hz-es határfrekvenciájú Bessel típusú aluláteresztő szűrő esetén:

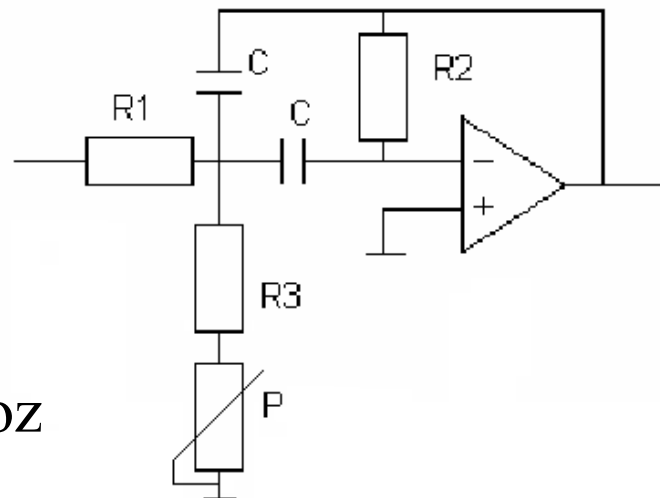
$$R1=16 \text{ k}\Omega \quad C1=100 \text{ nF}$$

$$R2=25 \text{ k}\Omega \quad C2=47 \text{ nF}$$

$$R3=76 \text{ k}\Omega \quad C2=10 \text{ nF}$$

Sávszűrő:

P potenciométer hangoláshoz



A méretezéshez szükséges összefüggések:

$$f_0 = \frac{1}{2\pi C} \sqrt{\frac{R_1 + (R_3 + P)}{R_1 R_2 (R_3 + P)}}, \quad A = -\frac{R_2}{2R_1}, \quad Q = \pi R_2 C f_0$$

A gyakorlatban a következő feltételek állnak fenn:

- $A=1$ ekkor $R_2 = 2R_1$
- általában igaz, hogy $R_1 \gg R_3$
- a C értékét megválaszthatjuk

így a következő összefüggéseket használhatjuk:

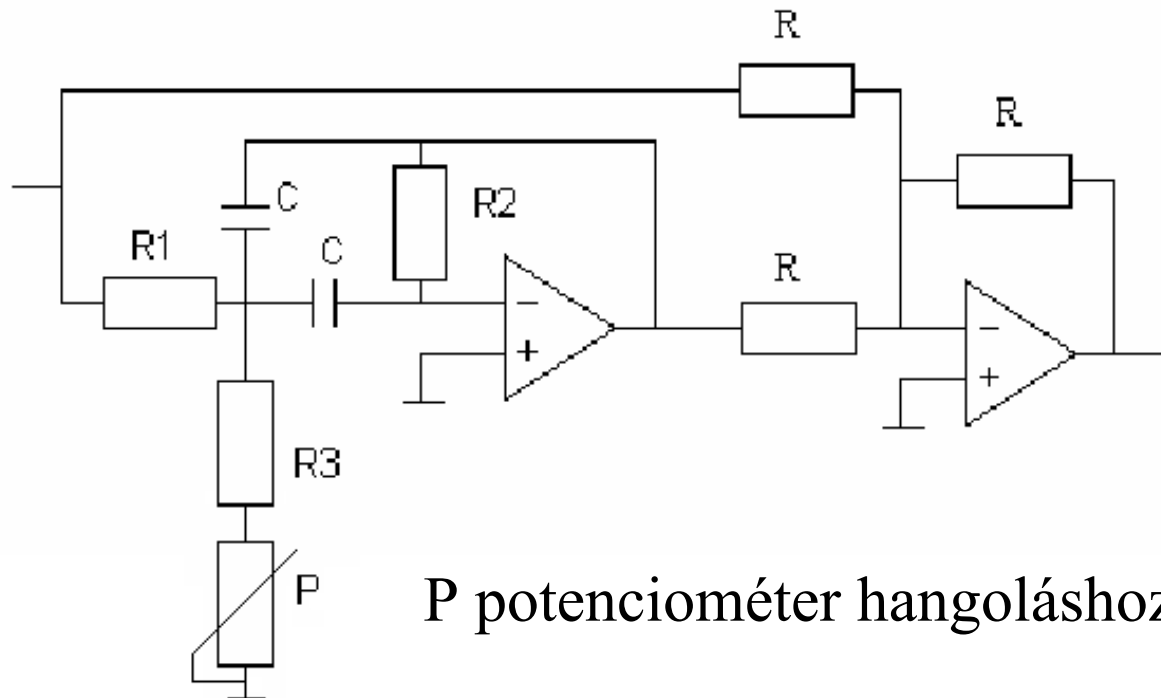
$$R_2 = \frac{Q}{\pi C f_0}$$

$$R_1 = R_2 / 2$$

$$R_3 = \frac{1}{R_2 (2\pi C f_0)^2} \quad (R_1 \gg R_3)$$

A sávszűrő hangolható ha R_3 változtatható ellenállás

Sávzáró szűrő:



P potenciométer hangoláshoz

Méretezés a sávszűrőhöz hasonló

Mérőváltók

Primer feszültség és áramváltók

- hálózatba beépített nagyméretű eszközök
- általában transzformátorok
- Lehet Kapacitív osztó

Feladatuk:

- szintillesztés
- galvanikus leválasztás
- védelmek, és mérőrendszerek táplálása

Áramváltó 400 kV



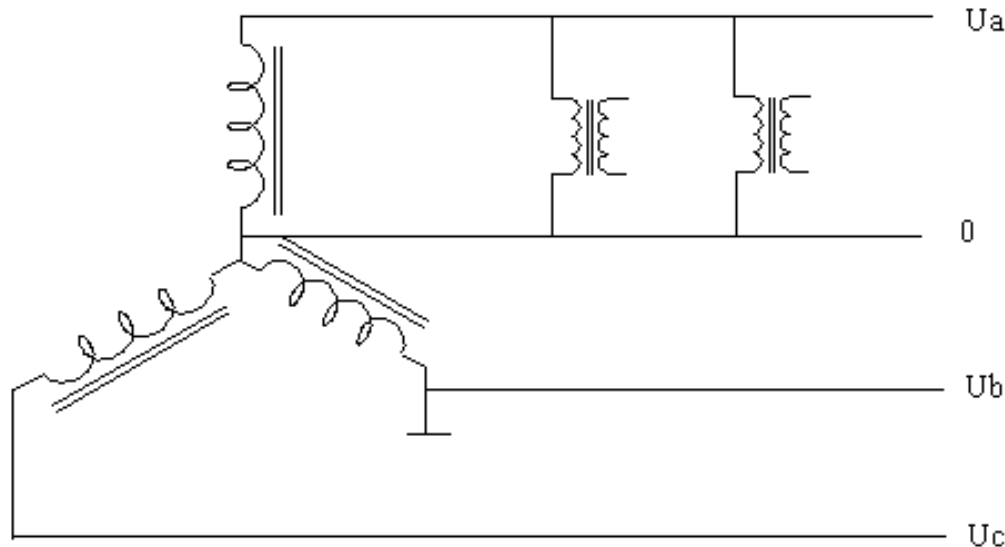
Feszültségváltó 120 kV



Kombinált mérőváltó 220 kV



Primer feszültségváltók kapcsolása:



A feszültségváltók csillagban vannak kötve.

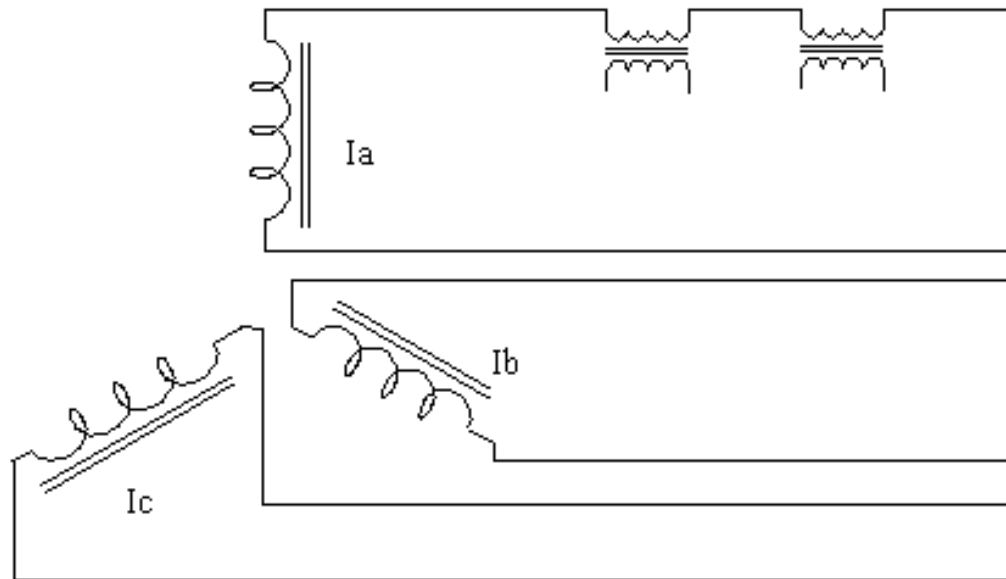
Névleges szekunder feszültségek 100V és 200V

Általában a B fázis földelt!

A 0 vezeték nem földelhető! (Védelem kismegszakítóval)

A fogyasztók párhuzamosan kapcsolódnak

Primer áramváltók kapcsolása:

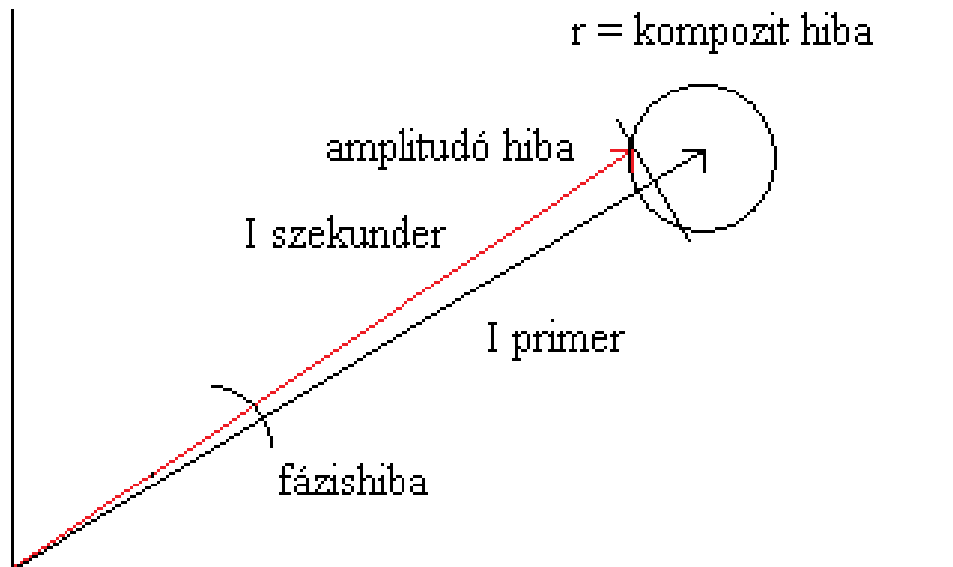


Névleges szekunder áramok: 1A és 5A
A fogyasztók sorosan kapcsolódnak
Az áramkör megszakítása tilos!

Áramváltók pontossága

Az áramváltók esetén háromféle hibát határozhatunk meg:

- amplitúdó hiba
- fázishiba
- kompozit hiba (vektoriális különbség)



- az amplitúdó hibát az effektív érték mérésénél kell figyelembe venni,
- a fázis hiba a teljesítmény mérések esetén lehet fontos,
- a kompozit hiba egy közös mennyiséggel jellemzi a pontosságot.

Az egyes hibák számítása (százalékban):

$$\varepsilon_A = 100 \frac{|K_n I_s| - |I_p|}{|I_p|} \quad \text{Kn névleges áttétel}$$

$$\varepsilon_{f_i} = \text{arc}(I_p) - \text{arc}(I_s)$$

$$\varepsilon_k = 100 \frac{|(K_n I_s - I_p)|}{|I_p|}$$

Mérő áramváltók

- alapvetően elszámolási mérési feladatokra szolgálnak.
- mérési tartomány max $1,2 I_n$

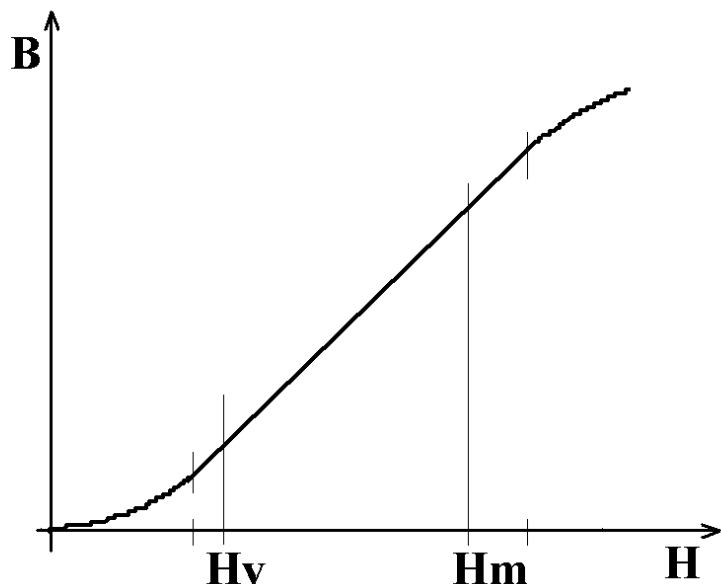
Mérő áramváltók megengedett hibái a kivezérlés függvényében:

Pontossági osztály	Amplitudó hiba [%]				Fázis hiba [perc]			
	5	20	100	120	5	20	100	120
kivezérlés %	5	20	100	120	5	20	100	120
0,1	0,4	0,2	0,1	0,1	15	8	5	5
0,2	0,75	0,35	0,2	0,2	30	15	10	10
0,5	1,5	1,5	0,5	0,5	90	45	30	30
1	3,0	3,0	1,0	1,0	180	90	60	60

Védelmi célú áramváltók

- alapvetően vezérlő és védelmi berendezések ellátására szolgál
- kivezérlési tartománya $10-20 I_n$
- pontosság 2.5, 5, 10 %

Miért alkalmazunk kétféle mérőváltót, mire jó ez?



A mérő áramváltók pontossági igényük miatt a gerjesztési görbe lineáris szakaszán dolgoznak, névleges áramhoz tartozó munkapont a felső könyökponthoz közelében van. Túlvezérlés 1.2-szeres.

A védelmi célú áramváltókkal szembeni kivezérlési tartománya miatt ez lehetetlen, tehát esetükben a névleges áramhoz tartozó munkapont az alsó könyökponthoz közelében van. Csak így biztosítható az akár húszszoros névleges áramátvitel. A két igény egyszerre nem teljesíthető!

Kétfázisú áramváltók.

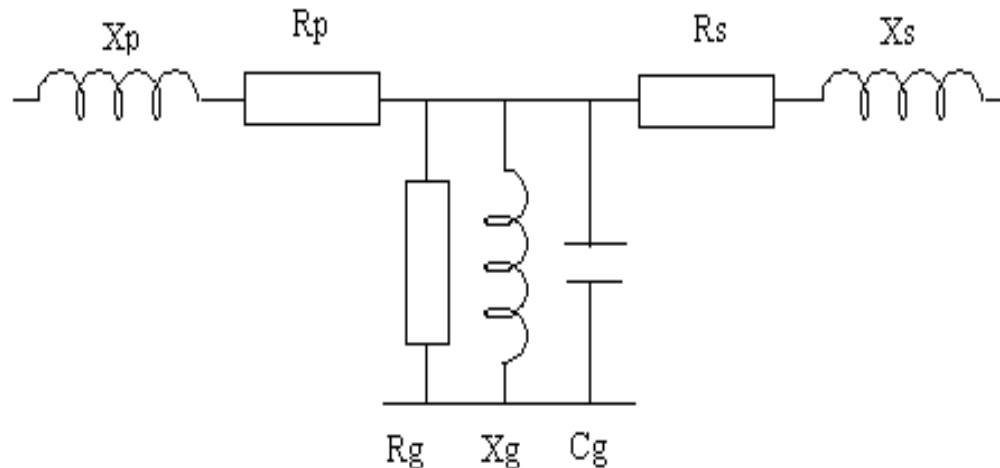
Szekunder váltók

A primer váltók által szolgáltatott feszültséget vagy áramot illesztik a mérőberendezés elektronikus részéhez.

Feladatok:

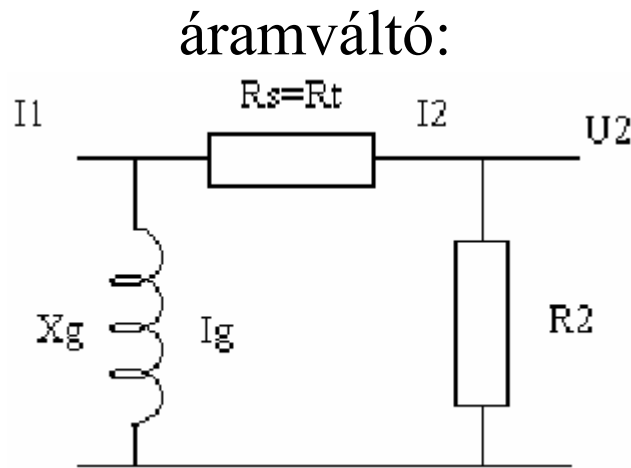
- szintillesztés (100, 200V ill. 1, 5A-ről 5-10V-os szintre)
- galvanikus leválasztás
- zavarvédelem

A transzformátor helyettesítő képe:



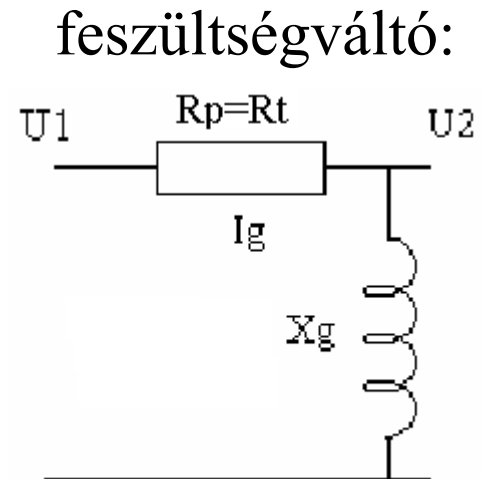
A helyettesítő kép egyszerűsíthető szekunder váltók esetére

- a gerjesztési ág ellenállása elhanyagolható
- a gerjesztési ág kapacitása elhanyagolható $C_g=0$ (50 Hz-en)
- a transzformátor kis méretű, ezért 50 Hz-en az X_s és X_p értéke elhanyagolható az R_s és R_p mellett (lásd később)



-a meghajtás áramgenerátoros jellegű, ezért a primer elemek (X_p , R_p) értéke elhanyagolható, tehát:

$$I_2 = I_1 - I_g$$



-a lezárás nagy impedanciás, ezért a szekunder elemek (X_s , R_s) értéke elhanyagolható, tehát:

$$U_2 = U_1 - R_t * I_g$$

Az áramváltó helyettesítő áramkör eredő impedanciája:

$$Z = (R_t + R_2) \times jX_g$$

Az áramváltó fázishibája ennek megfelelően:

$$\varphi = \arctan\left(\frac{R_t + R_2}{X_g}\right)$$

Az áramváltó amplitúdó hibája:

$$h = 1 - \frac{U_{2val}}{U_{2id}} = 1 - \frac{I_1 |Z| \frac{R_2}{R_t + R_2}}{I_1 R_2} = 1 - \frac{|Z|}{R_t + R_2} = 1 - \frac{1}{\sqrt{1 + \left(\frac{R_t + R_2}{X_g}\right)^2}} \cong \frac{1}{2} \left(\frac{R_t + R_2}{X_g}\right)^2$$

A szekunder áramváltók névleges szekunder árama általában 10 mA

A kimenő feszültség ekkor

R₂=50 Ohm esetén 0.5V,

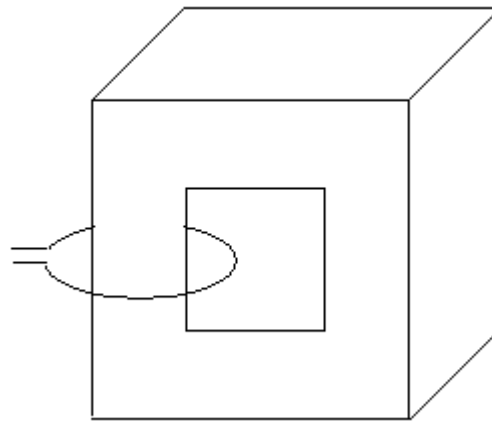
R₂=200 Ohm esetén 2.0V

R₂=500 Ohm esetén 5.0 V

Jósági tényező változása a méret függvényében

(X_s és $X_p \ll R_s$ és R_p bizonyítása)

Határozzuk meg az alábbi fojtótekercs jósági tényezőjét:



Legyenek a méretek az alábbiak:

tekercs keresztmetszet A_t , menethossz L_t , menetszám n ,

vasmag: keresztmetszet A_v , hossz L_v

A tekercs ellenállása:

$$R = \frac{\rho L}{A} = \frac{\rho n L_t}{A_t/n} = \frac{\rho n^2 L_t}{A_t}$$

A tekercset $i=i_0 \sin(\omega t)$ árammal gerjesztve adódik

$$H = \frac{n i_0 \sin(\omega t)}{L_v} \quad \Theta = \mu_0 \mu_r A_v H$$

$$U = n \frac{d\Theta}{dt} = \frac{n^2 \mu_0 \mu_r A_v}{L_v} i_0 \omega \cos(\omega t)$$

A jósági tényező ezekből:

$$Q = \frac{X}{R} = \frac{U}{iR} = \frac{\mu_0 \mu_r}{\rho} \omega \frac{A_v A_t}{L_v L_t}$$

Változtassuk meg a geometriai méreteket k -szorosra, ekkor

$$A_t' = k^2 A_t \quad \text{és} \quad A_v' = k^2 A_v \quad \text{és} \quad L_t' = k L_t \quad \text{és} \quad L_v' = k L_v$$

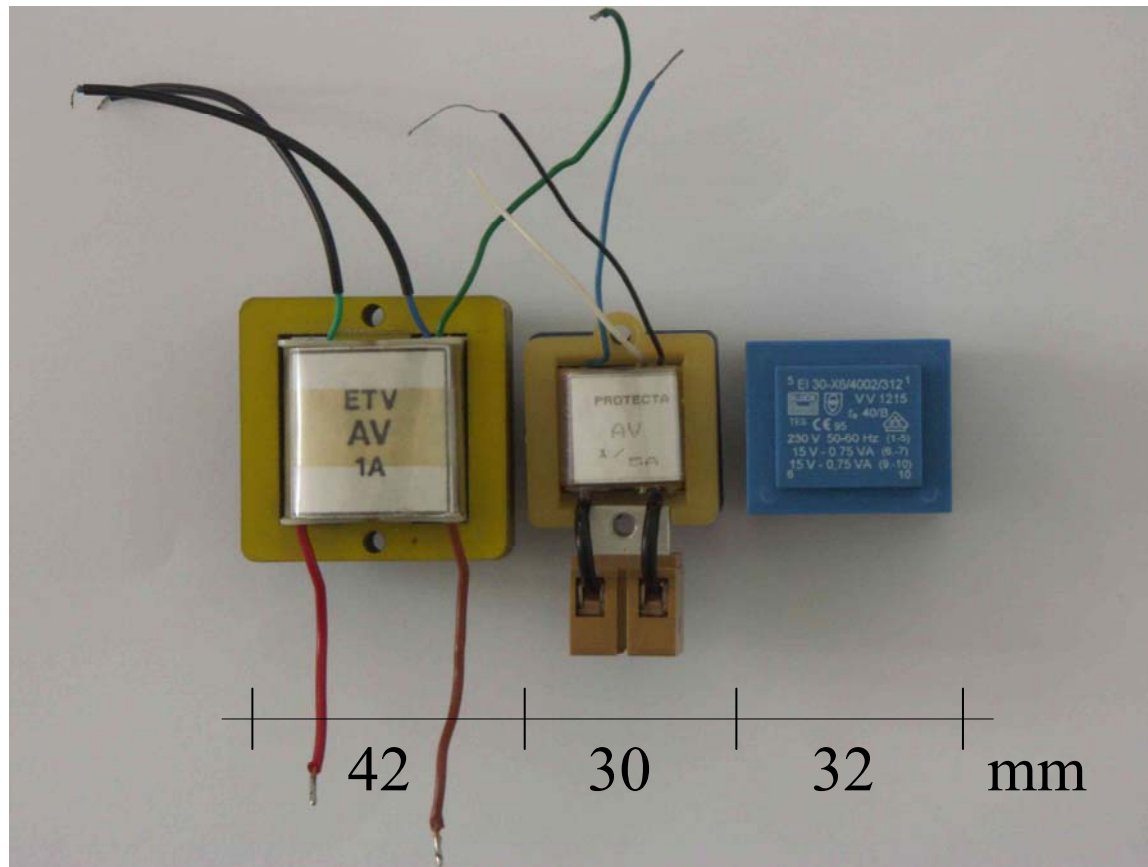
behelyettesítve:

$$Q' = k^2 Q$$

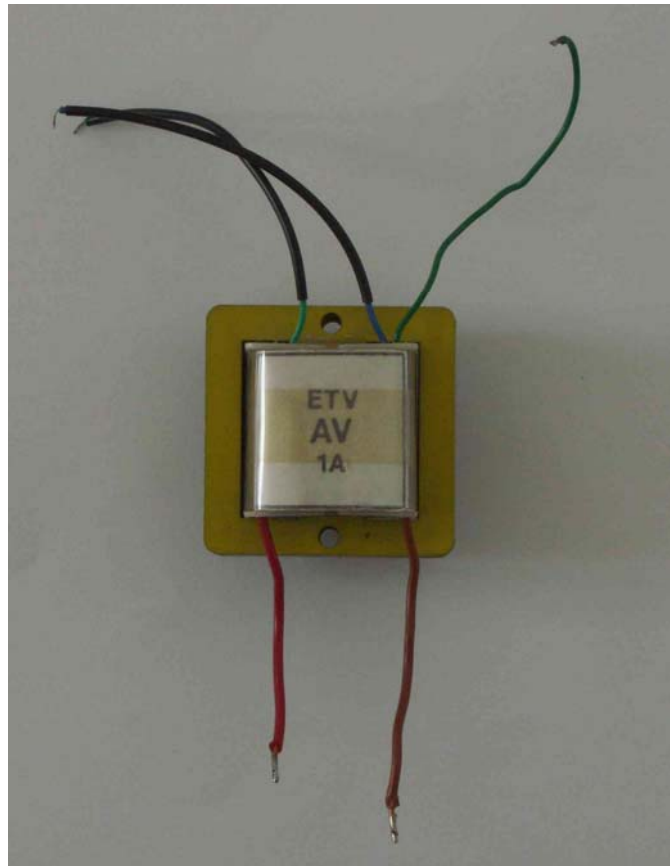
Látható, hogy a jósági tényező a geometriai méret függvénye, kis méretű induktivitásokkal nem lehet nagy jósági tényezőt létrehozni. A jósági tényező növelésének egyedüli lehetősége a μ_r növelése, ezért szekunder áramváltókat nagy permeabilitású anyaggal célszerű készíteni, ezek azonban igen drágák.

Szekunder váltók esetében $X_g > R_t$, de nem **sokkal** nagyobb (kis méretük miatt), viszont X_t (szórási reaktancia) $\ll X_g$ így ezekből következik, hogy $X_t \ll R_t$, amiből indultunk.

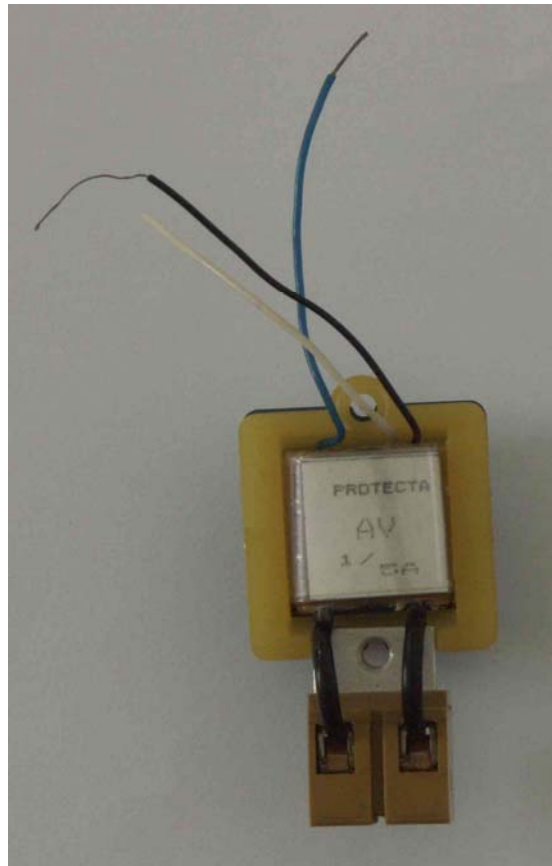
Szekunder mérőváltók



Áramváltó M42-es magon



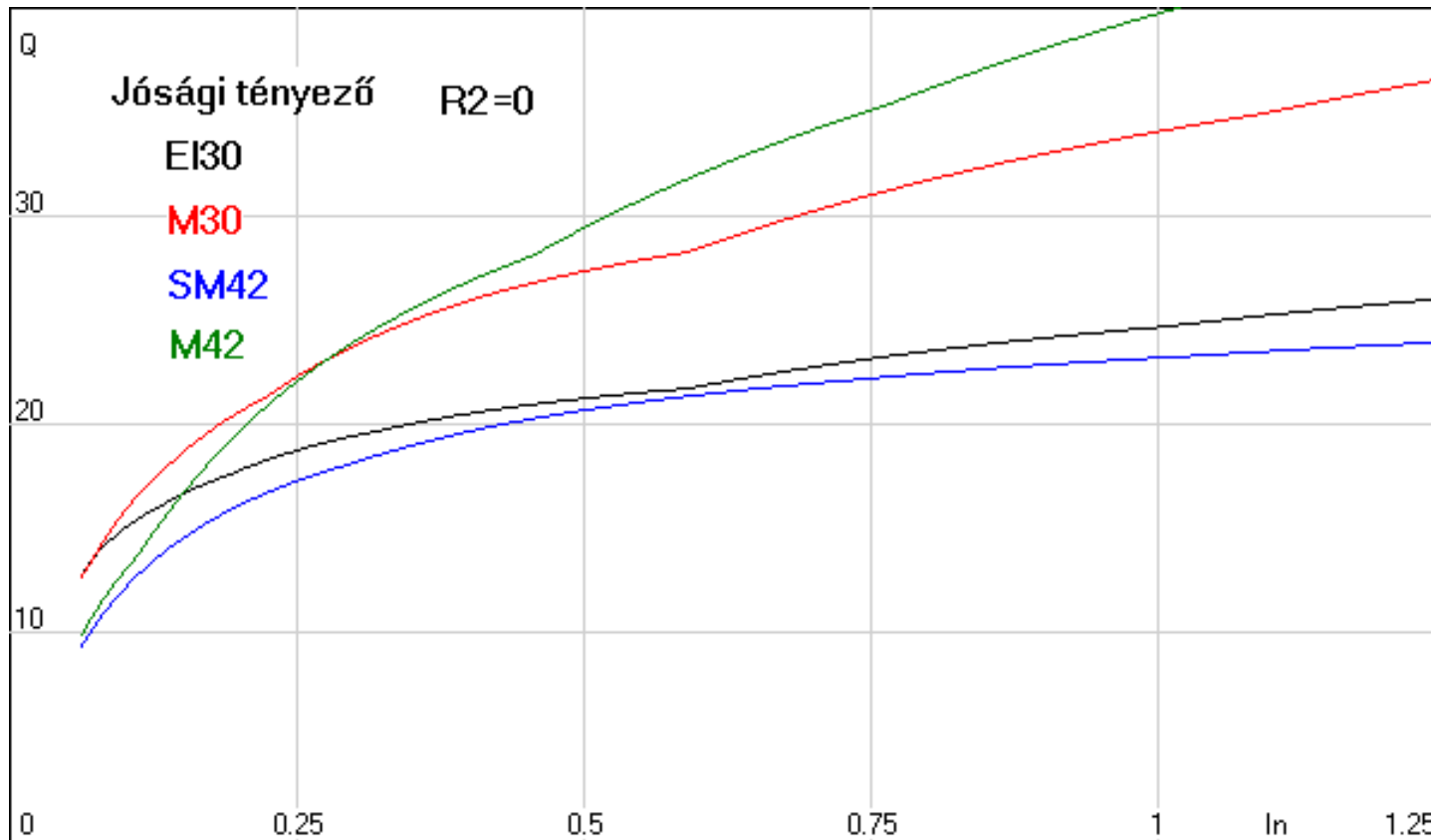
Áramváltó M30-as magon



Különböző típusú szekunder áramváltók jósági tényezője az áram függvényében.

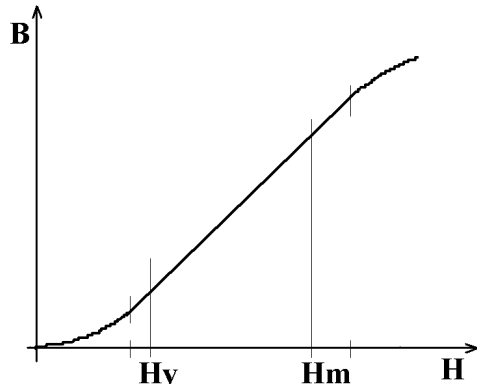
A vizsgált típusok: EI30, M30, SM42, M42.

Az egyes transzformátorok jósági tényezője (X_g/R_s):



Jól látható, hogy az M42-es permalloy B, mag a legjobb.

Miért változik a jósági tényező a kivezérlés függvényében?



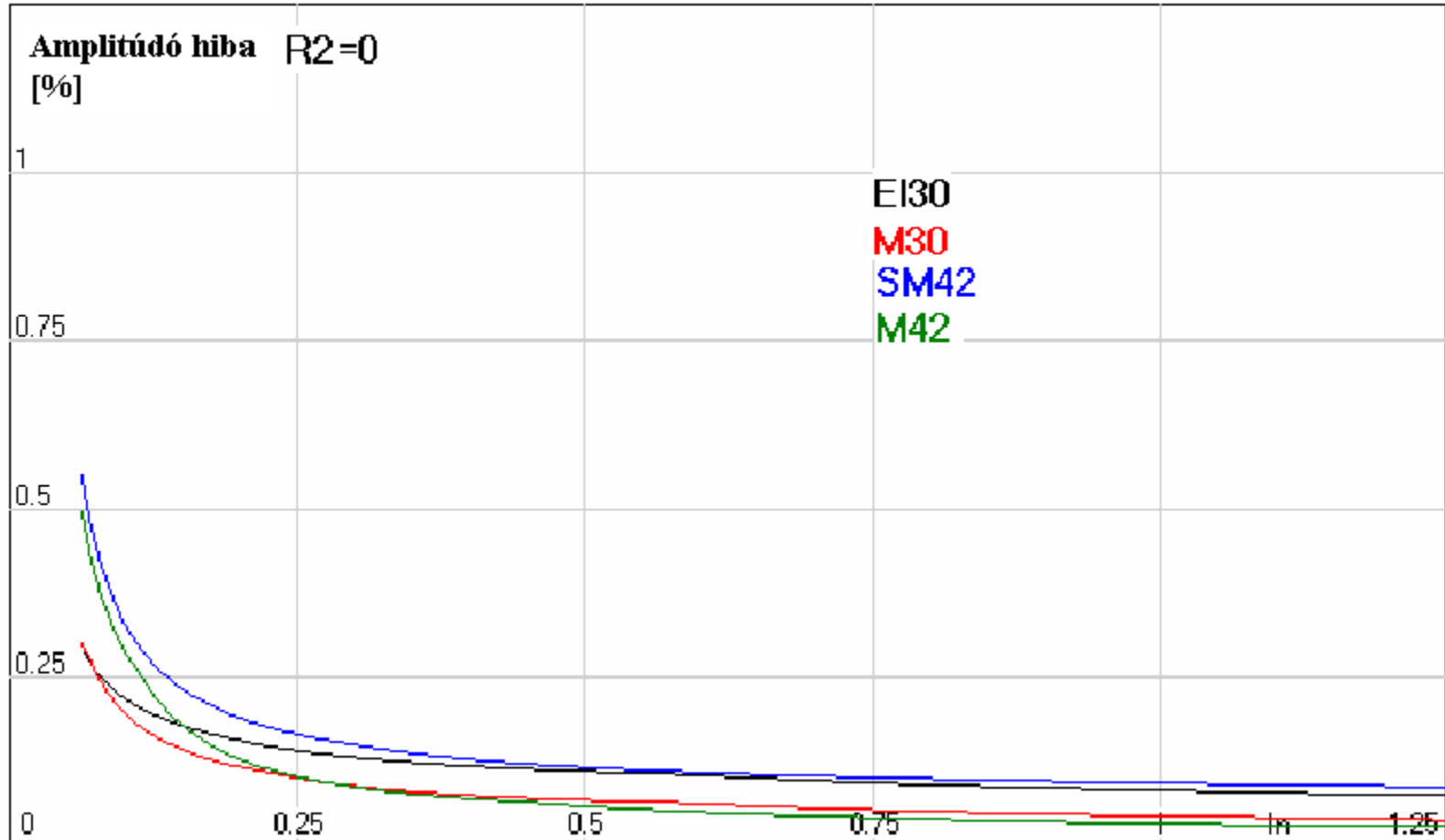
$$Q = \frac{\mu_0 \mu_r}{\rho} \omega \frac{A_v A_t}{L_v L_t} = \mu_r k$$

Mert a permeabilitás jóval kisebb kis kivezérlésnél, mint a névleges értéknél. Kisebb permeabilitás, kisebb jósági tényező.

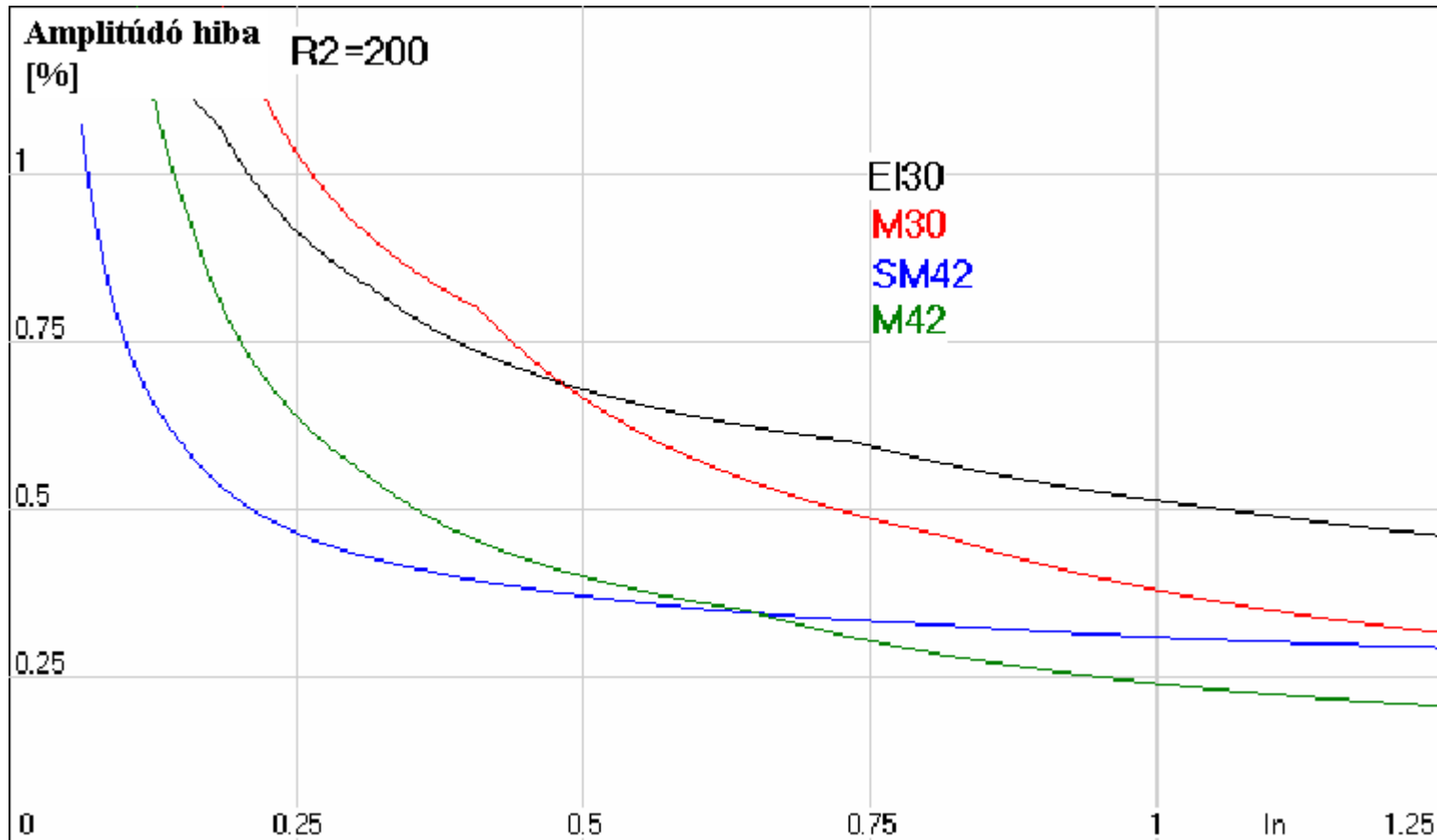
A jósági tényező határozza meg a mérőváltó pontosságát, mert a gerjesztő áram minél kisebb, annál kisebb az általa okozott amplitudó hiba. Tehát a nagy pontosságú mérőváltókat minden esetben célszerű a felső könyökpont környékéig kivezérelni.

Lezáró ellenállás növelése kell a kimenő feszültséghez, de ez is csökkenti a jósági tényezőt (eredő ellenállás = $R_s + R_2$).

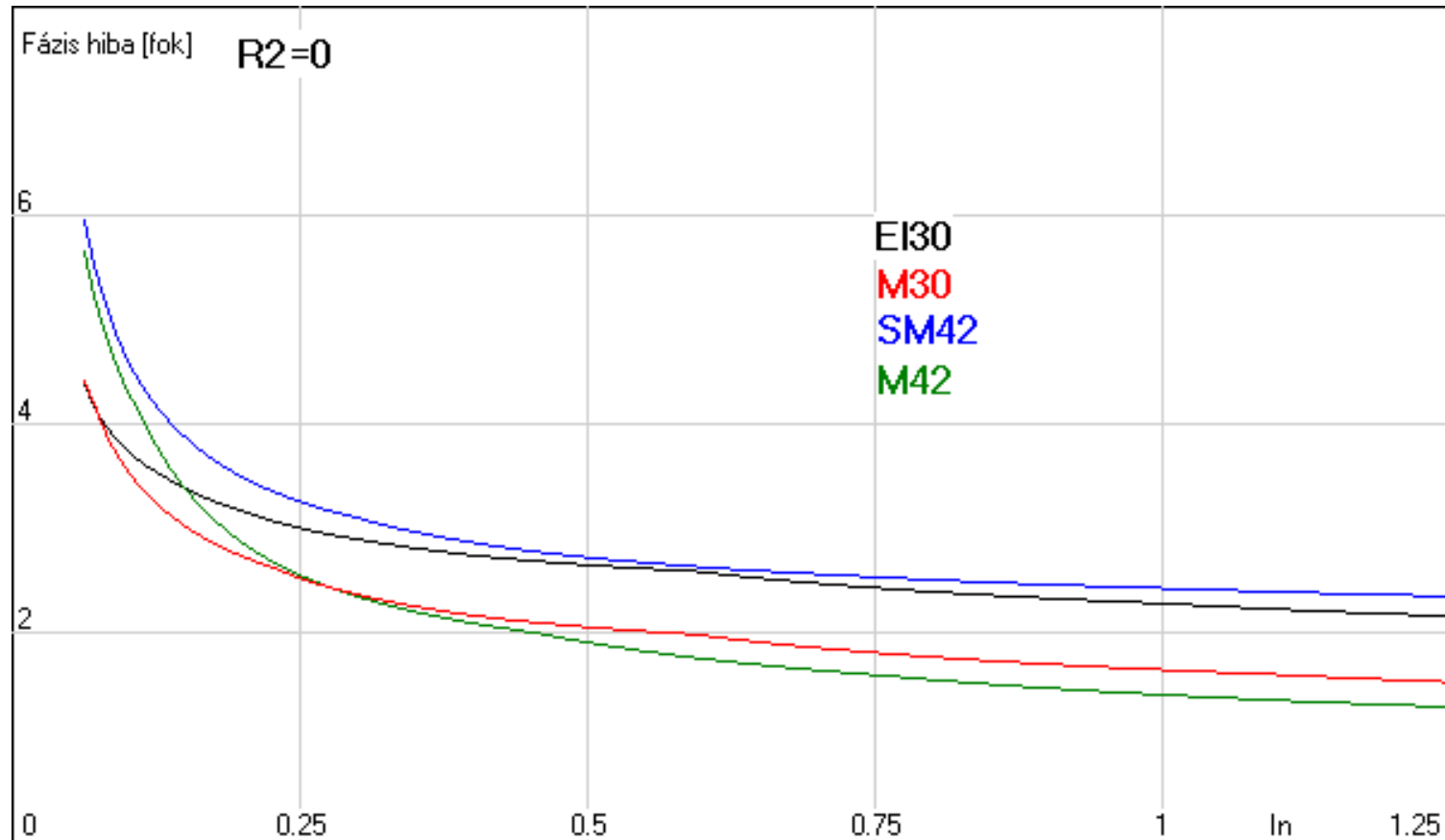
Amplitúdó hiba. Lezáró ellenállás 0 Ohm



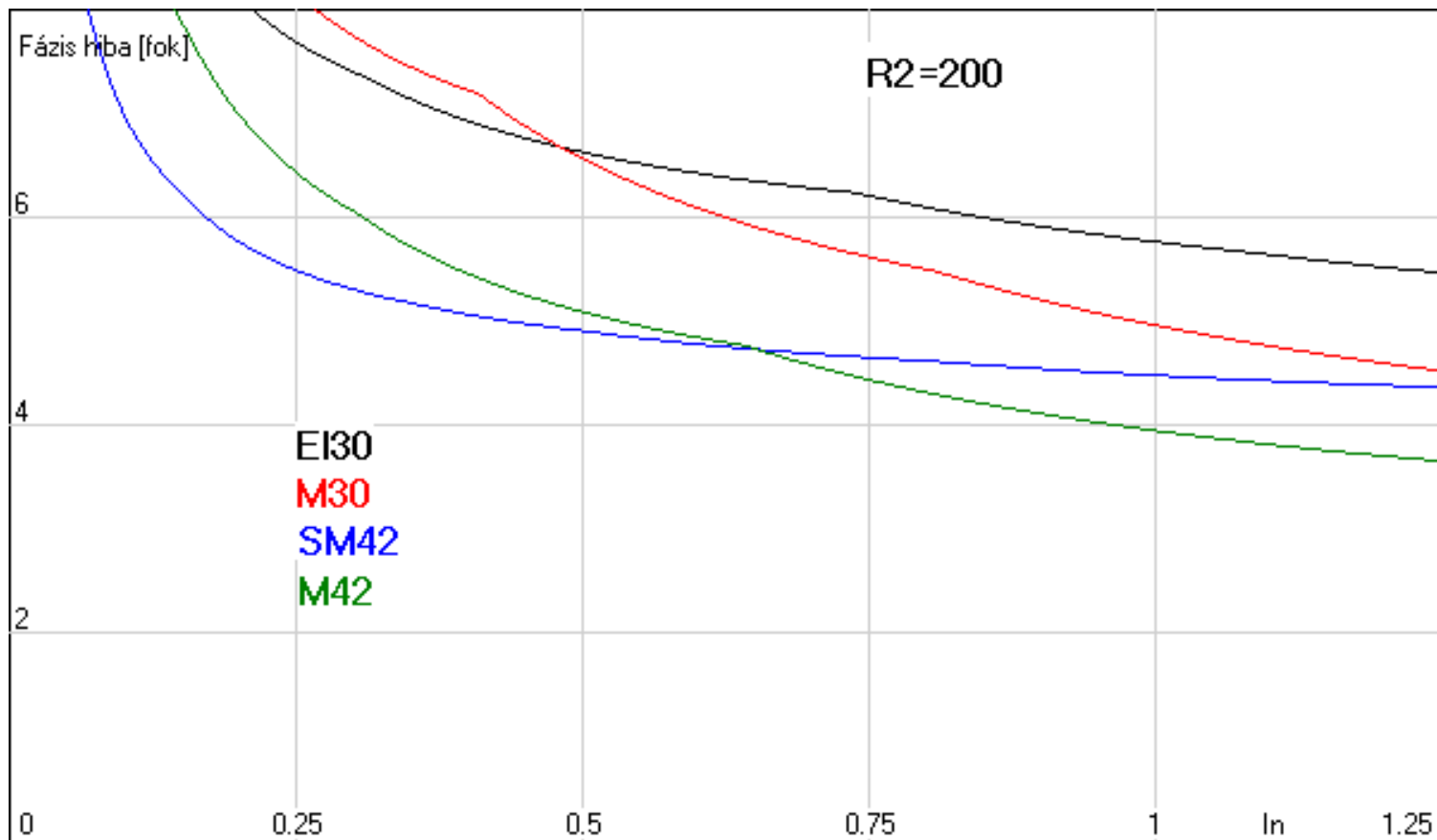
Amplitúdó hiba. Lezáró ellenállás 200 Ohm



Fázis hiba. Lezáró ellenállás 0 Ohm

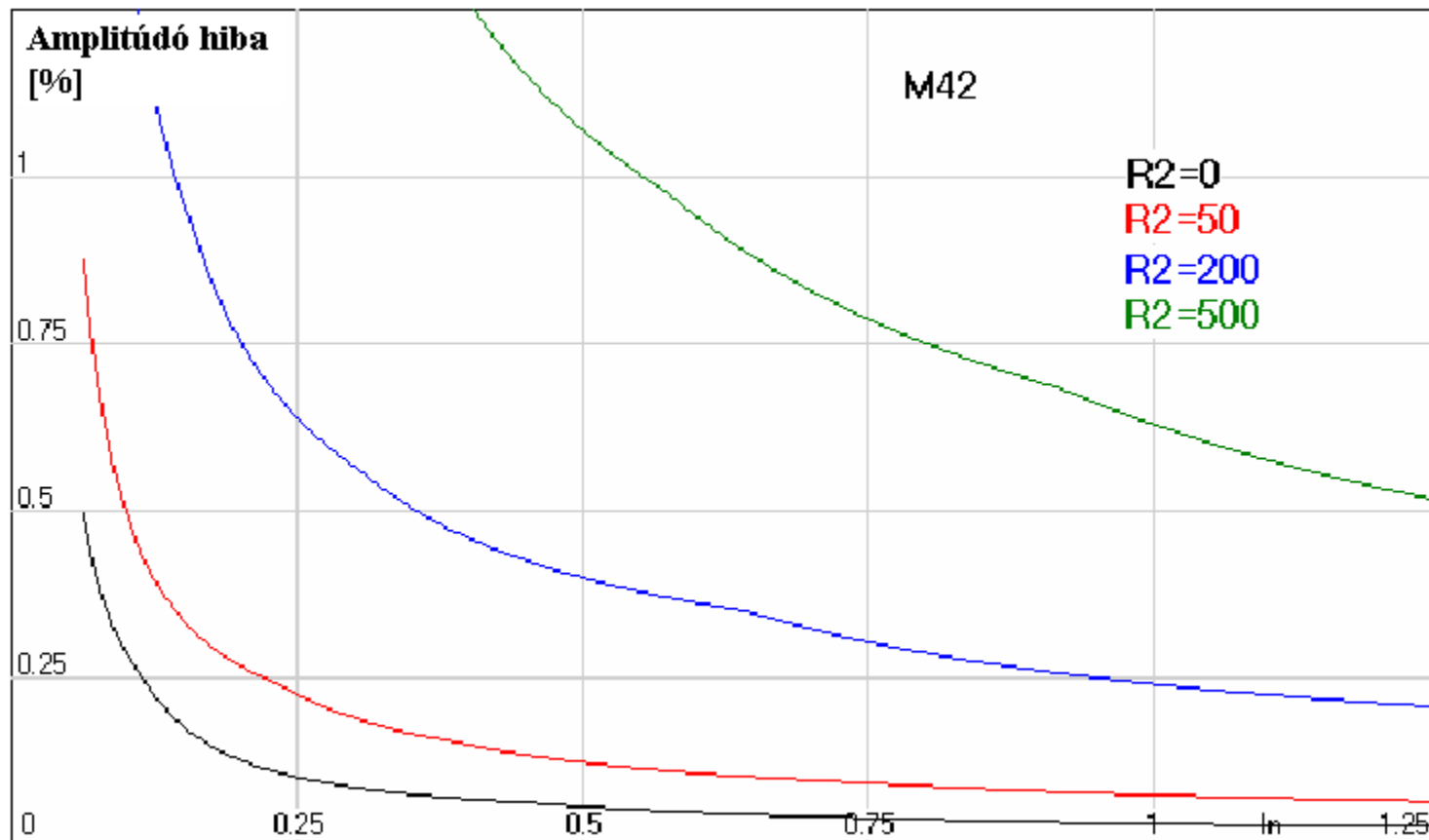


Fázis hiba. Lezáró ellenállás 200 Ohm

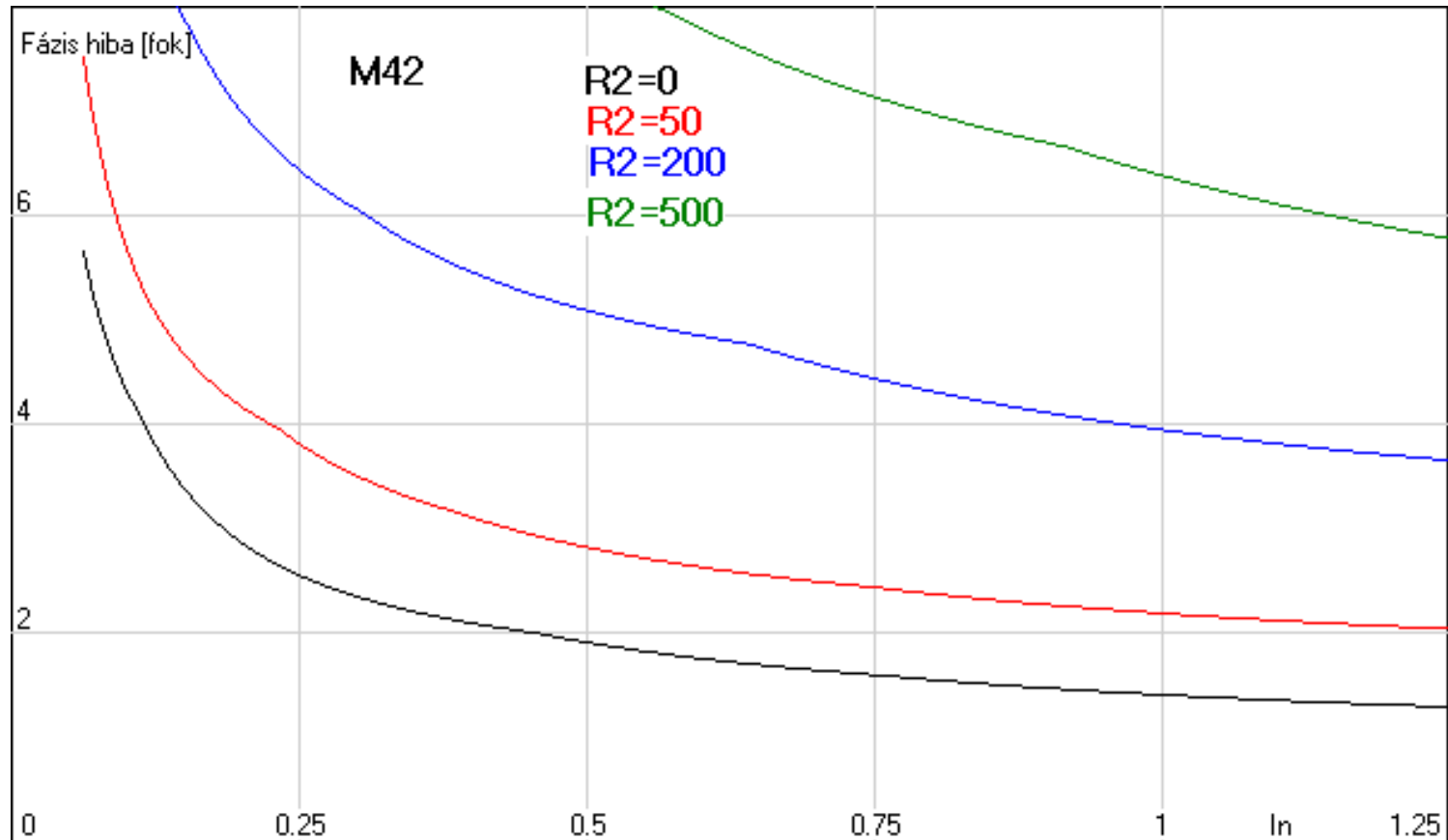


Pontossági osztály	Fázis hiba [perc]			
	5	20	100	120
kivezélés %				
0,1	15	8	5	5
0,2	30	15	10	10
0,5	90	45	30	30
1	180	90	60	60

M42 áramváltó amplitúdó hibája a lezárás függvényében:

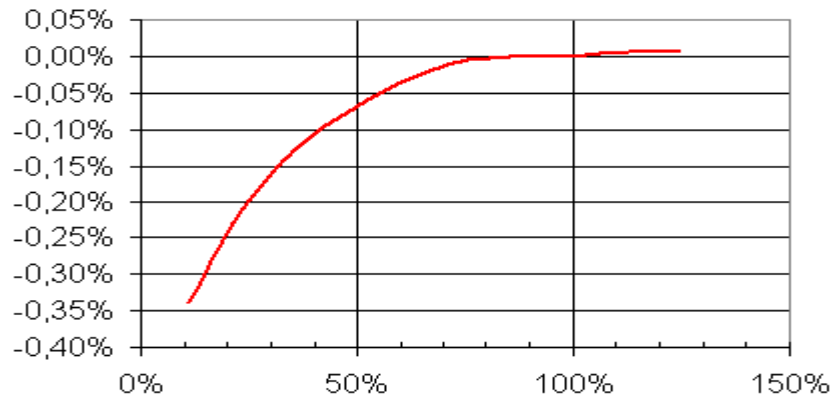


M42 áramváltó fázis hibája a lezárás függvényében:

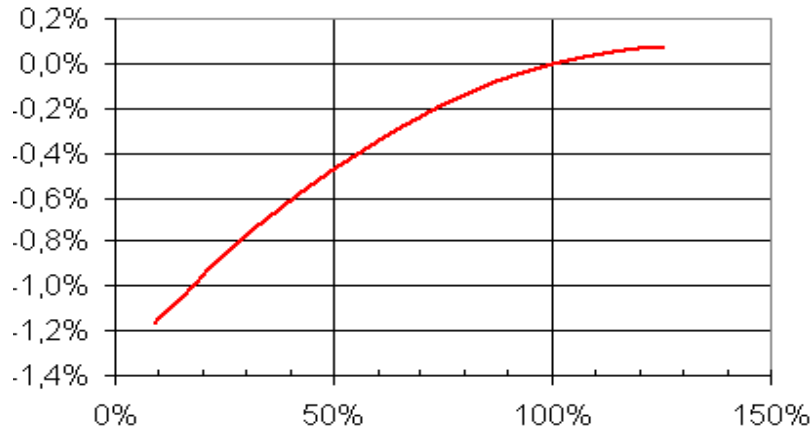
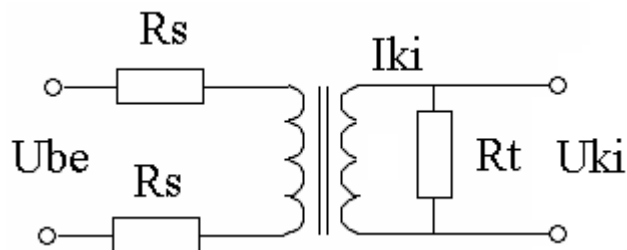


Szekunder feszültségváltók pontossága a kivezérítés függvényében

ETV100/5V M42 magon

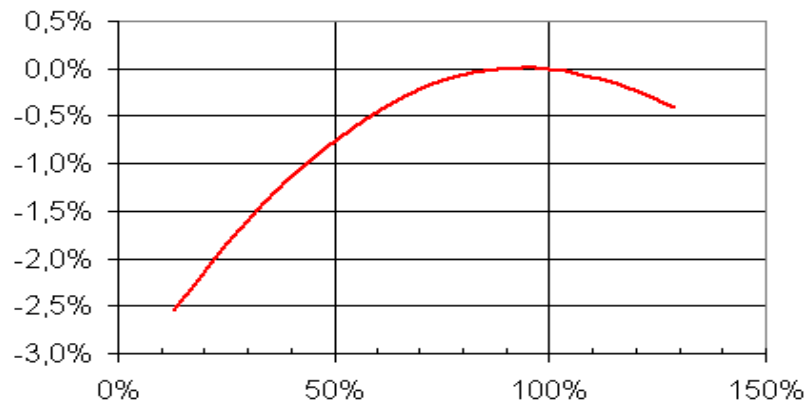


FV 10/10mA M30 magon

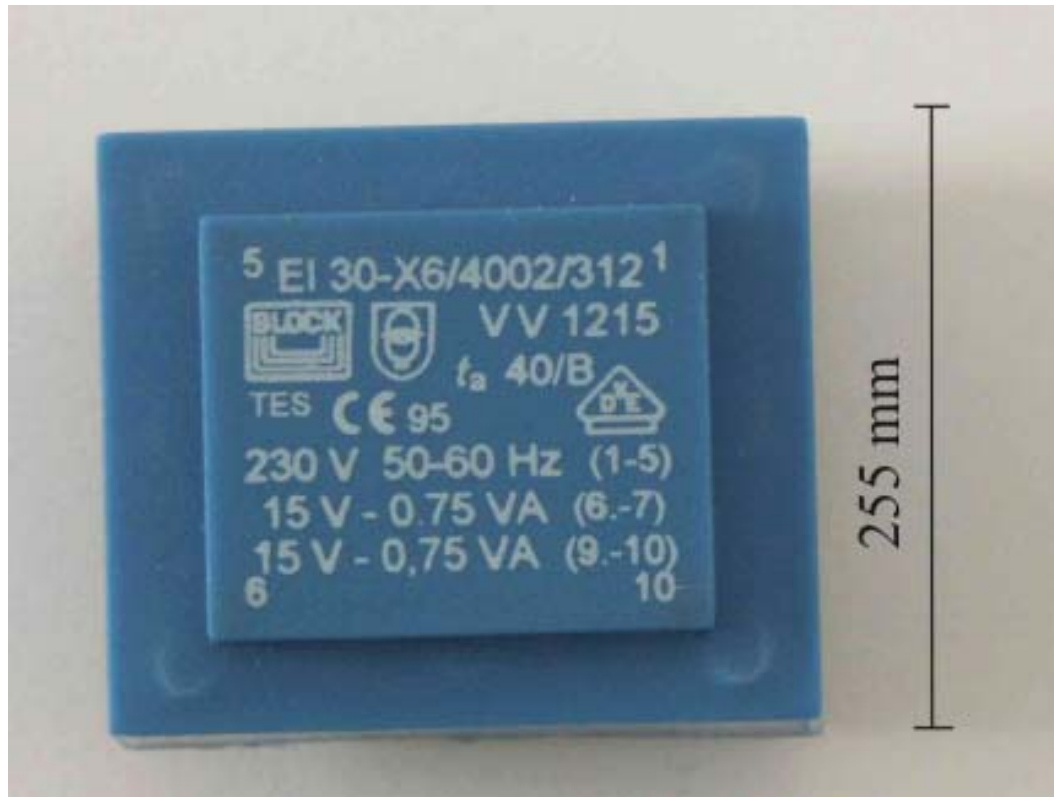


230V/15V trafó 1.5 VA

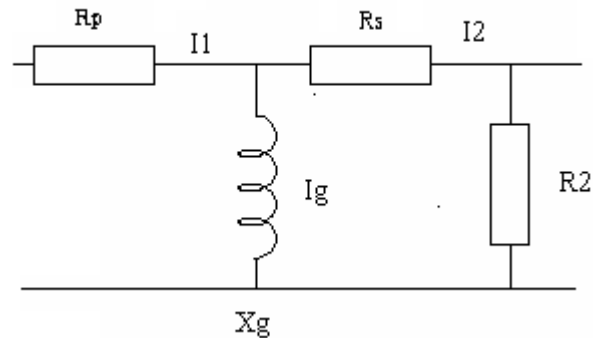
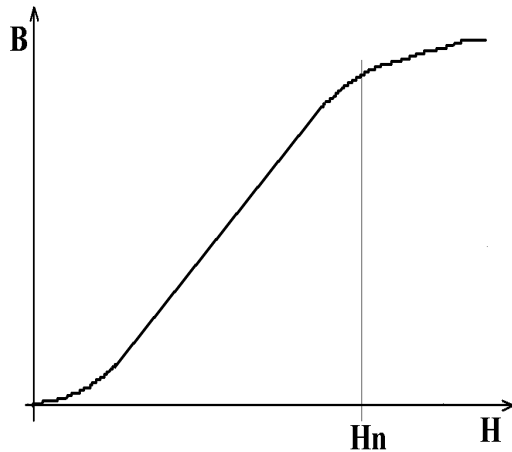
Miért fordul vissza a görbe?



230/2x15 V-1,5 VA-es feszváltó



Miért fordul vissza a görbe?



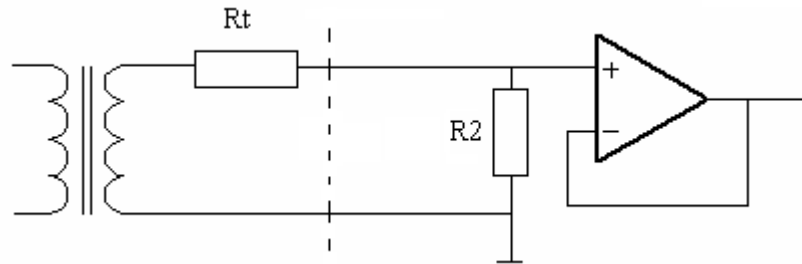
Takarékossági okokból a munkapont a felső könyökpontnál van. Áram növekedés hatására a telítődés miatt X_g és a jósági tényező is csökken, így X_g -n fellépő feszültség is csökken.

A gerjesztést úgy méretezik, hogy R_p -n eső feszültségesést a névleges terhelésnél figyelembe veszik. Gyakorlatilag nincs tartalék. Üresen járó transzformátoron megemelkedik a feszültség és emiatt a gerjesztő áram is ami erőteljes melegedést okoz.

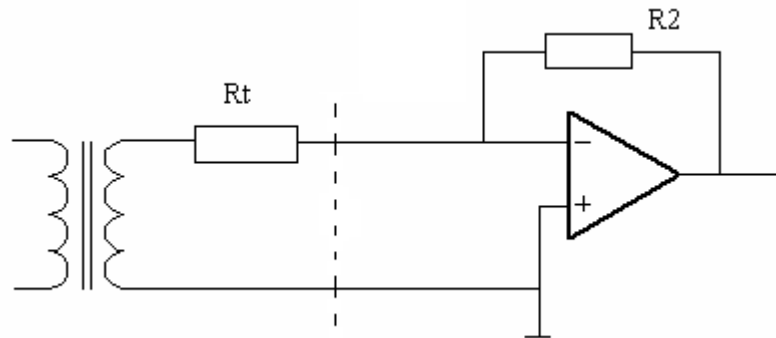
Tartósan névlegesnél magasabb bemeneti feszültség is gondot okoz.

Szekunder áramváltók kapcsolása:

1. Lezáró ellenállás R2:



2. Lezáró ellenállás 0:



Túl áram elleni védelemre figyelni kell!

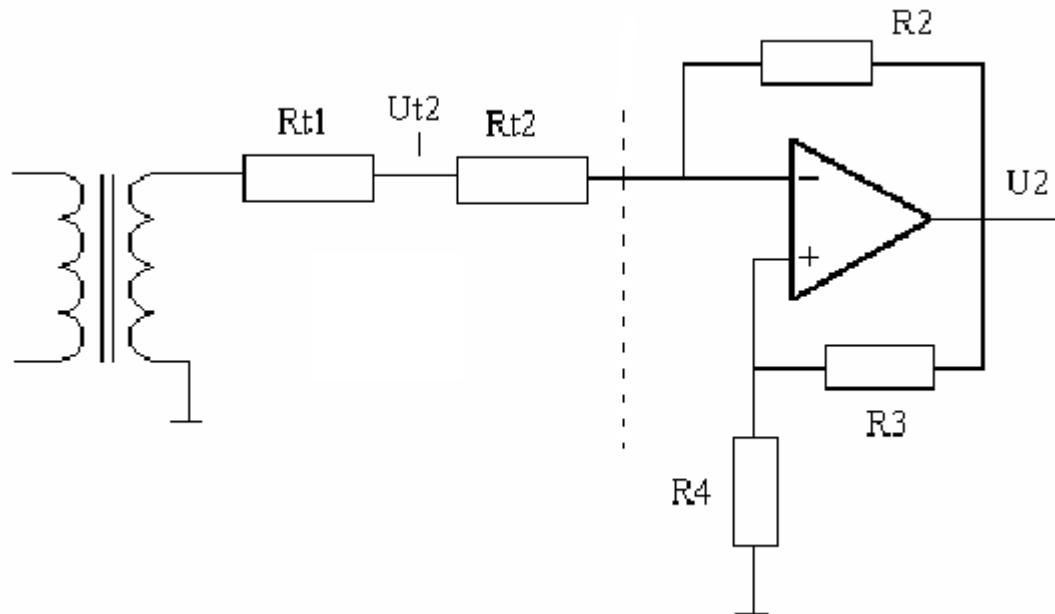
3. Lezáró ellenállás csökkentése pozitív visszacsatolással

Az R_t tekercsellenállást bontsuk két részre

Ha $R_3/R_4=R_2/R_{t2}$ akkor az U_{t2} feszültség=0, ezért a látszólagos tekercsellenállás $R_{t1}<R_t$ lesz

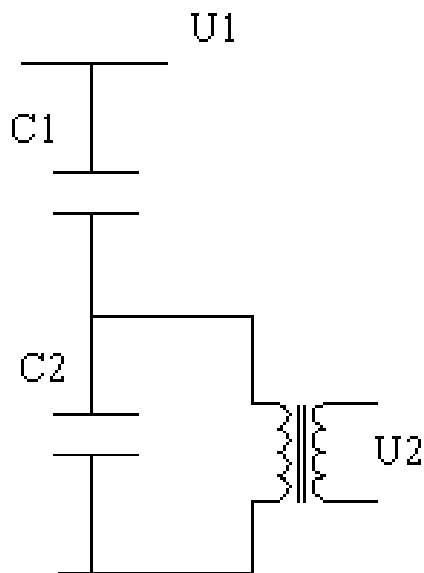
$$U_2 = -i_2(R_{t2} + R_2)$$

A kapcsolás instabil mert R_t értéke hőmérsékletfüggő és ha $R_{t1}<0$ lesz ($R_{t2}>R_t$), akkor a kapcsolás lengeni fog.



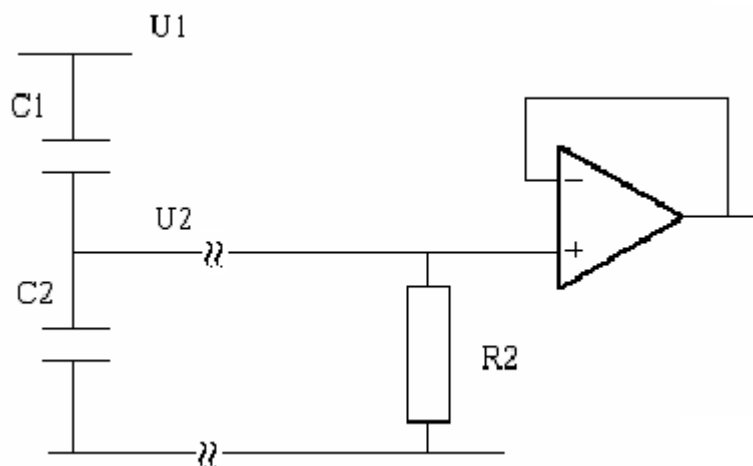
Kombinált feszültségváltó

- igen nagy feszültségek esetén használatos pl. 750 kV
- csak 50 Hz-en pontos (frekvencia függése igen összetett)



Kapacitív osztó és alkalmazása

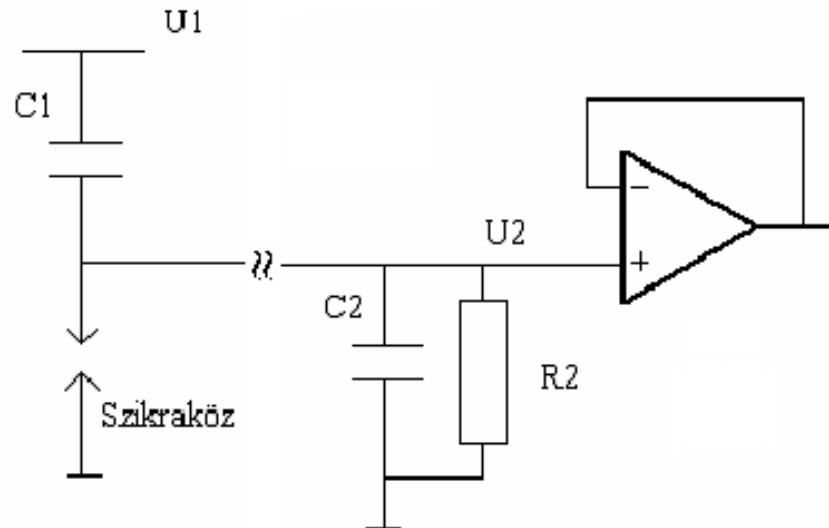
- széles frekvencia tartományban lineáris átvitel
- egyen feszültség mérésére nem alkalmas
- alsó könyökponti frekvencia: $\omega_1 = 1 / R_2 C_2$



Osztás miatt C_2 nagy és mert $\Omega_1 \ll 50$ Hz kell $\rightarrow R_2$ is nagy.
Hosszú vezeték \rightarrow nagy hurok \rightarrow nagy zavarérzékenység!

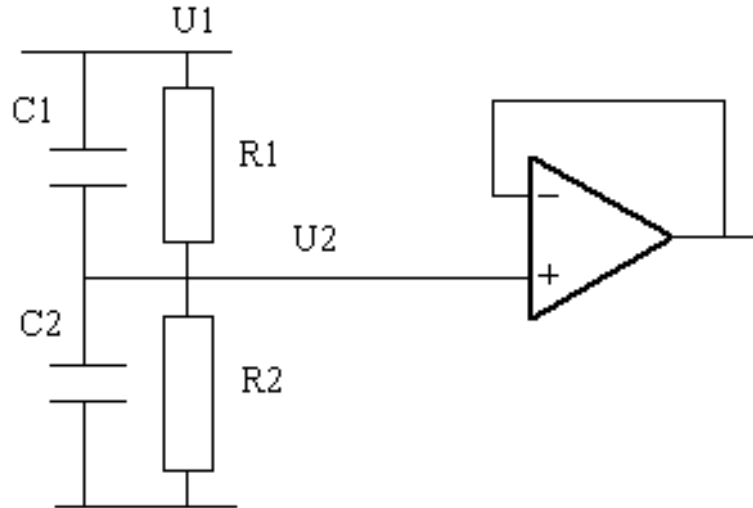
Kapacitív osztó és alkalmazása

- Felső tag beépítve védő szikraközzel a helyszínen
- Alsó tag lezárással a mérőberendezés helyén
- Áramgenerátoros meghajtás (nagy U_1 és Z_{C1})
- Nagy zavarérzékletlenség



Kompenzált osztó

- egyenfeszültség mérésére alkalmas
- egyenletes átvitel feltétele: $R_1 * C_1 = R_2 * C_2$

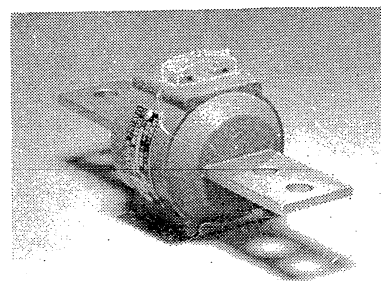
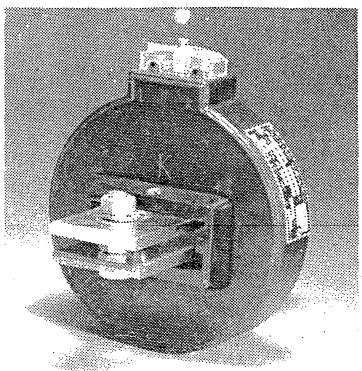
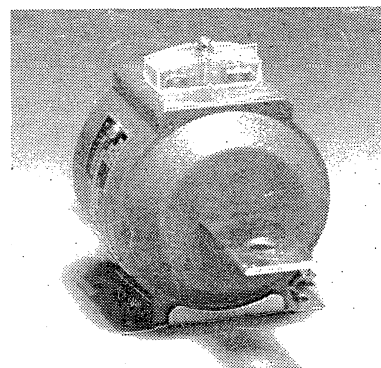
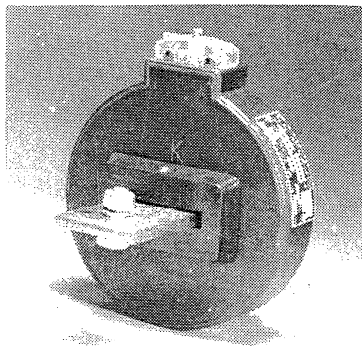
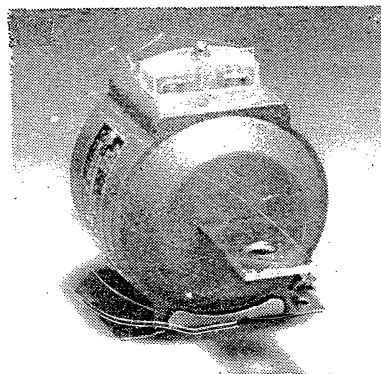
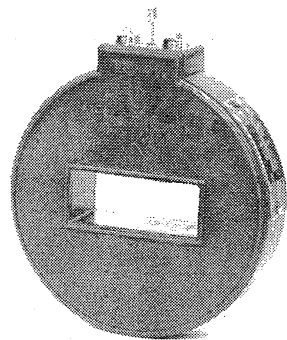


Megvalósítás beszerezhető R és C elemekből sorba kapcsolással

Középfeszültségű kompenzált Kapacitív osztó



Középfeszültségű áramváltók



Kisfeszültségű áramváltók



Sínre szerelhető



Primertekercses



Kábelre húzható



Nyitható

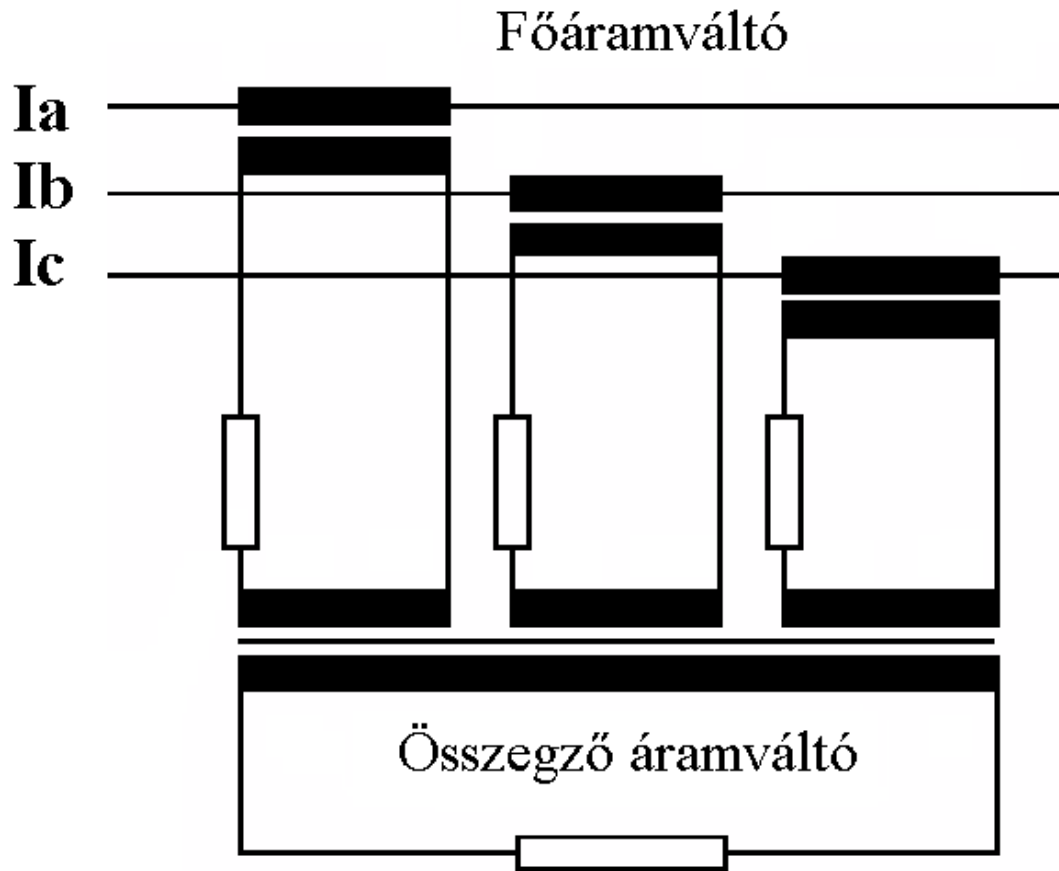


Összegző



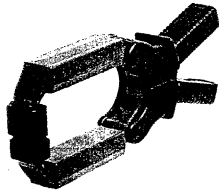
Távadó beépítve 40

Összegző áramváltó



Pl.: zérus sorrendű
áram mérése

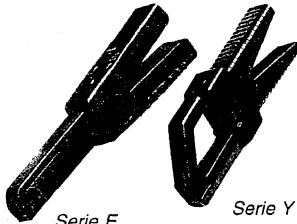
Lakatfogók



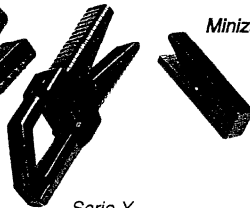
Serie D



Serie B



Serie E



Serie Y

Minizangen

■ 4 neue Formen



Serie K



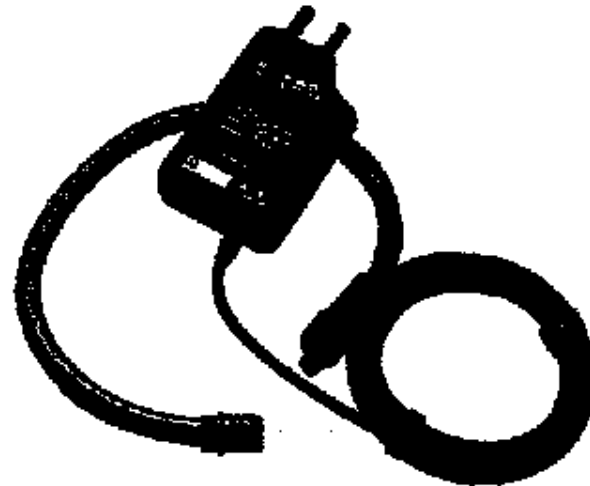
Series PAC



Serie MN

Serie C "100"

Flexibilis nyitható áramváltó



Digitális jelfeldolgozás

Analóg-digitális átalakítók

Irodalom:

www.analog.com

Analóg és digitális jelfeldolgozás összehasonlítása

Analóg módszer

egyszerű áramkör

egyszerűbb feladatok (effektív érték)

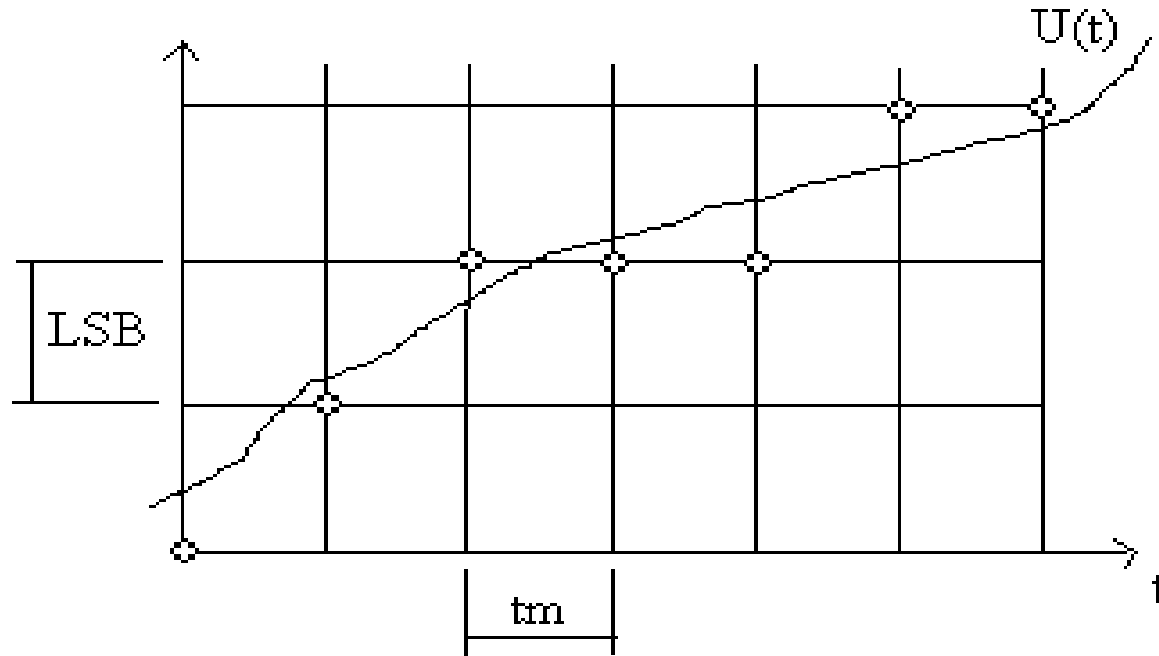
Digitális módszer

összetett áramkör (AD, uP)

összetett feladatok (spektrum)

-	tárolás, feldolgozás, regisztrálás	+
-	feladatok módosítása	+
-	pontosság	+
+	sávszélesség (mintavétel)	-
+	zavarérzékenység	-
-	költség	+
+	tesztelés, hibakeresés	-

Folyamatos időfüggvényből kvantált digitális jelek képzése



Az időben és amplitúdójában folyamatos $U(t)$ függvényből t_m időközönként megjelenő digitális jelek (számok) keletkeznek

A kvantálás jellemzői:

időben: t_m (minta/sec)

amplitúdóban LSB

(kivéve dual-slope elvű átalakítók)

Az AD konverterek jellemző adatai

Mintavételi sebesség:

voltmérők 50 S/sec (Sample/secundum)

hálózati jelekre 50-500 kS/sec

oszcilloszkópok 20 MS/sec - 20 GS/sec

Felbontás: 8-12-16-24 bit (méréstechnikában
leggyakoribb a 12 bit)

Bemeneti jeltartomány: (FS)

monopoláris 0..2.5V, 0..5V, 0..10V

bipoláris $\pm 5V$, $\pm 10V$

Az AD konverterek jellemző adatai II.

Kimeneti kódok:

unipoláris, offset bináris, 1-es komplement, 2-es komplement

Bemeneti csatornaszám 1..8 csatorna

Felbontás: a legkisebb mérhető változás (1 LSB)

$$1 \text{ LSB} = 2 \cdot \text{FS} / 2^n \text{ (n:bitszám, bipoláris)}$$

Mintavételezési hibák:

erősítés (FS), offset, nemlinearitás, hiányzó kód

Fontosabb AD konverter típusok és jellemzői

kettős integráló	3..4.5 digit, lassú, voltmérők
szukcesszív appr.	8-16 bit, közepes sebesség (méréstechnikában)
paralell (flash)	8-12 bit, közepes sebesség, processzorok
szigma-delta	16-24 bit, kis sebesség, nagy felbontás (hangfeldolgozás)

8 bites AD konverter kimeneti kódok

Analóg jel	unipoláris	offs. bináris	1-es kompl.	2-es kompl.
+FS	1111.1111	-	-	-
+FS-LSB	1111.1110	1111.1111	0111.1111	0111.1111
1 LSB	0000.0001	1000.0001	0000.0001	0000.0001
+0	0000.0000	1000.0000	0000.0000	0000.0000
-0	-	1000.0000	1111.1111	0000.0000
-1 LSB	-	0111.1111	1111.1110	1111.1111
-FS+LSB	-	0000.0001	1000.0000	1000.0001
-FS	-	0000.0000	-	1000.0000

AD konverterek felbontása

RESOLUTION N	2^N	VOLTAGE (10V FS)	ppm FS	% FS	dB FS
2-bit	4	2.5 V	250,000	25	-12
4-bit	16	625 mV	62,500	6.25	-24
6-bit	64	156 mV	15,625	1.56	-36
8-bit	256	39.1 mV	3,906	0.39	-48
10-bit	1,024	9.77 mV (10 mV)	977	0.098	-60
12-bit	4,096	2.44 mV	244	0.024	-72
14-bit	16,384	610 μ V	61	0.0061	-84
16-bit	65,536	153 μ V	15	0.0015	-96
18-bit	262,144	38 μ V	4	0.0004	-108
20-bit	1,048,576	9.54 μ V (10 μ V)	1	0.0001	-120
22-bit	4,194,304	2.38 μ V	0.24	0.000024	-132
24-bit	16,777,216	596 nV*	0.06	0.000006	-144

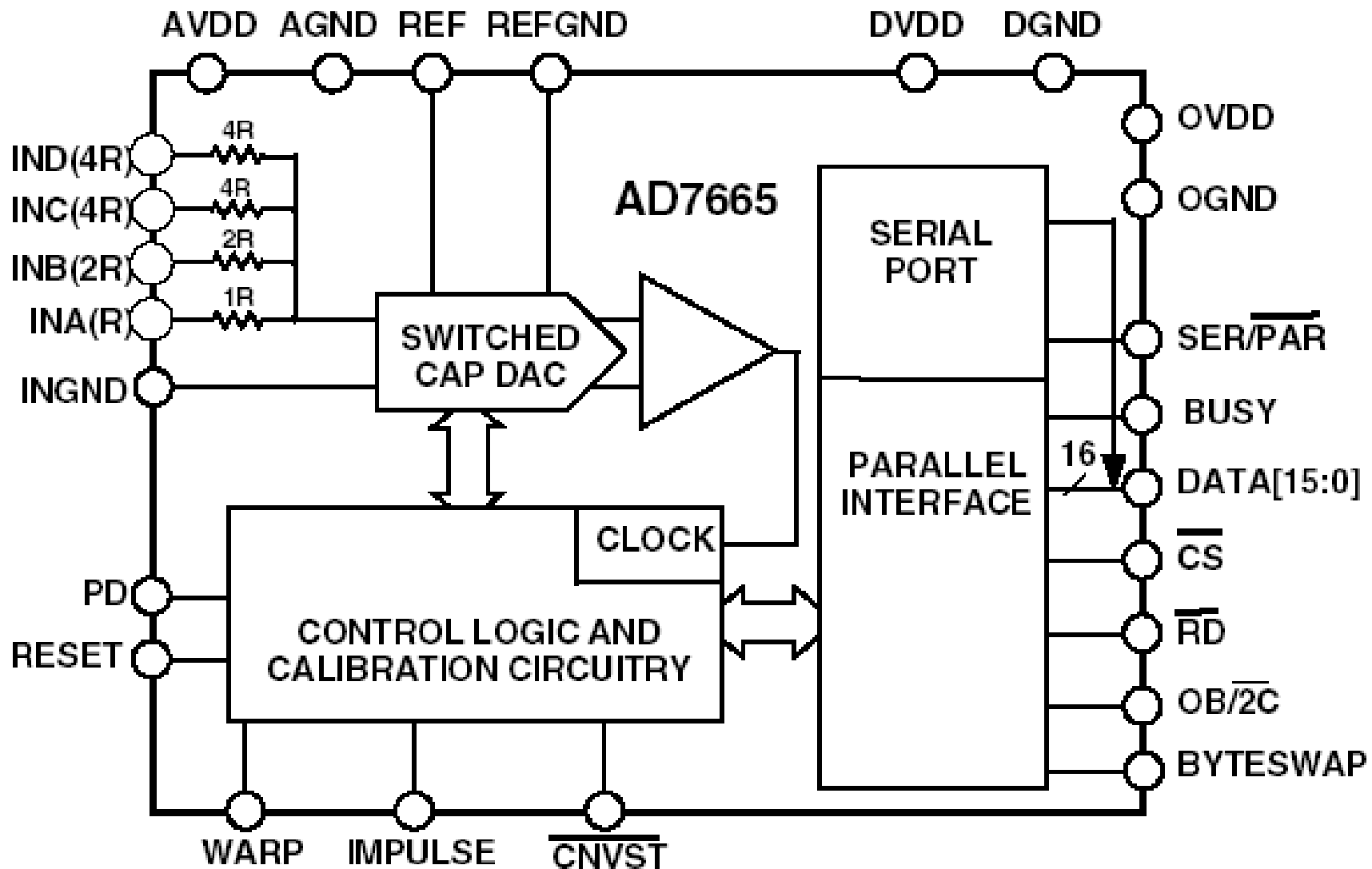
*600nV is the Johnson Noise in a 10kHz BW of a 2.2k Ω Resistor @ 25°C

Néhány AD konverter típus jellemzői

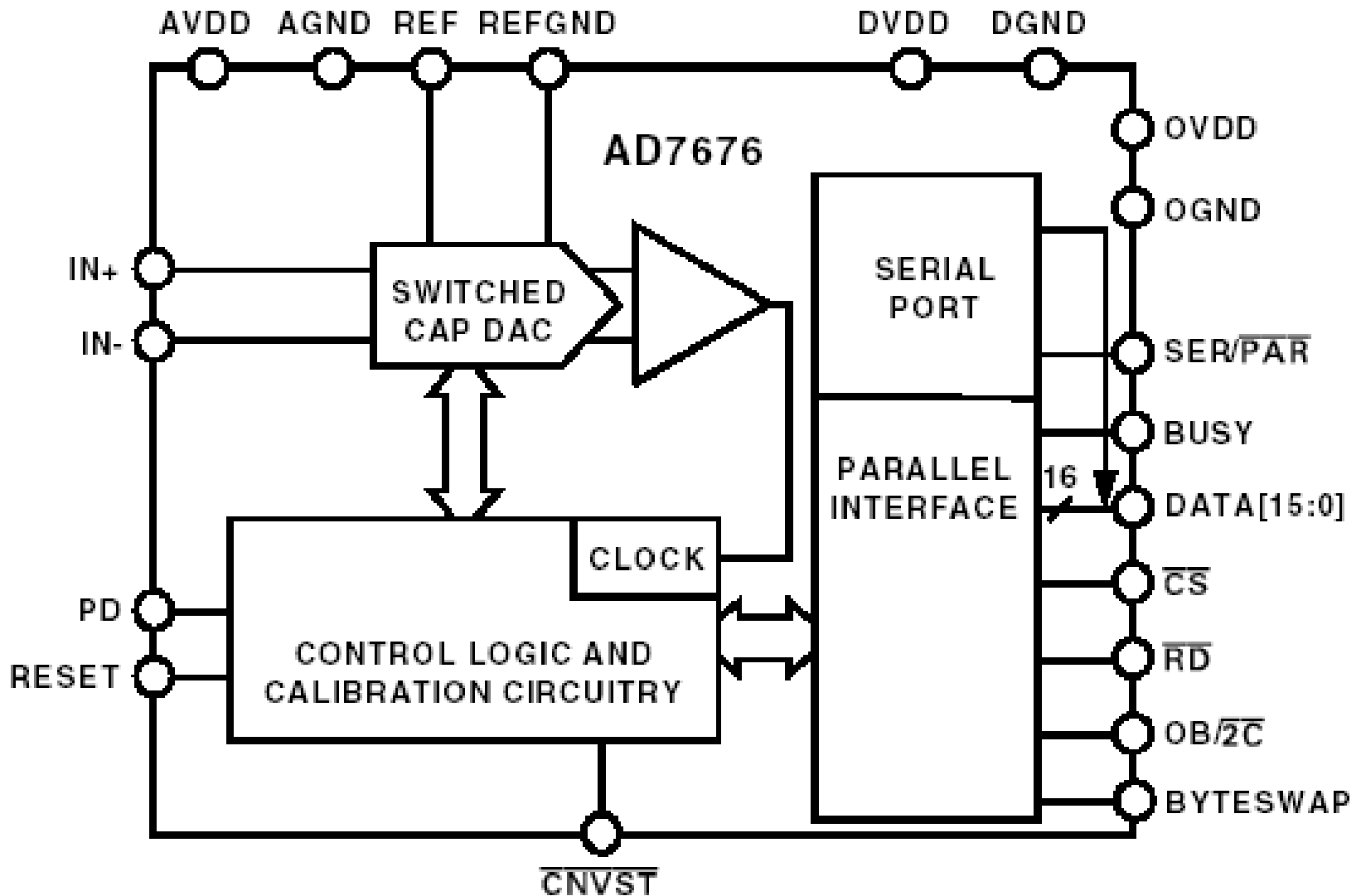
RESOLUTION / CONVERSION TIME COMPARISON FOR REPRESENTATIVE SINGLE-SUPPLY SAR ADCs

	RESOLUTION	SAMPLING RATE	POWER	CHANNELS
AD7472	12-BITS	1.5MSPS	9mW	1
AD7891	12-BITS	500kSPS	85mW	8
AD7858/59	12-BITS	200kSPS	20mW	8
AD7887/88	12-BITS	125kSPS	3.5mW	8
AD7856/57	14-BITS	285kSPS	60mW	8
AD7660	16-BITS	100kSPS	15mW	1
AD974	16-BITS	200kSPS	120mW	4
AD7664	16-BITS	570kSPS	150mW	1

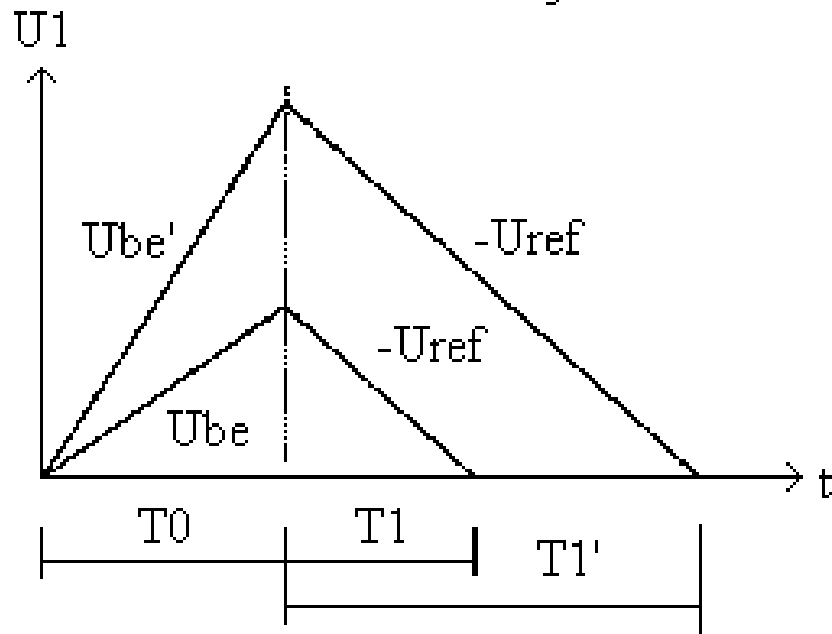
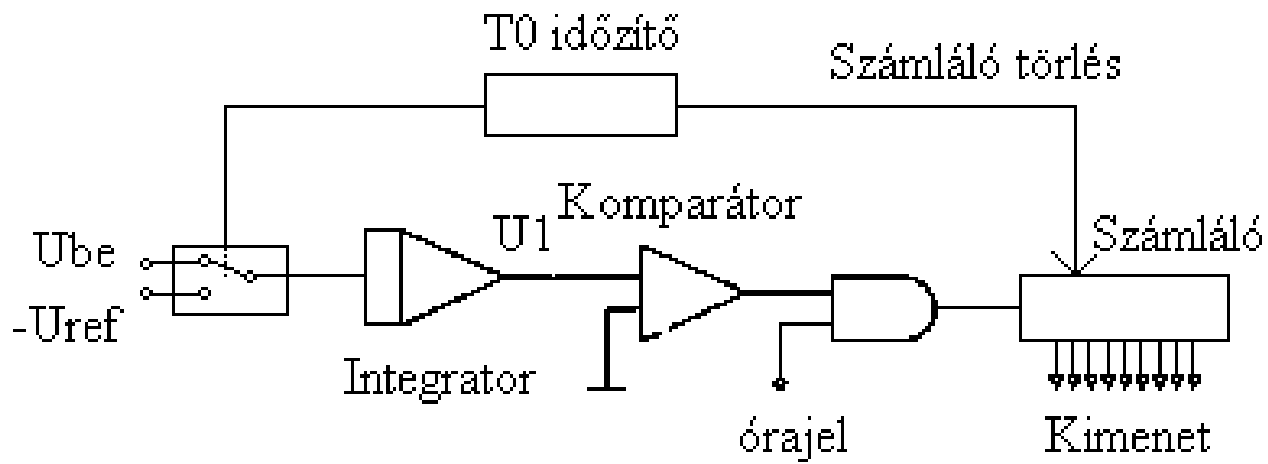
AD 7665 16 bites 500ks/sec AD konverter



AD7676 16 bites 500ks/sec AD konverter $\pm 2.5V$ diff. bemenet

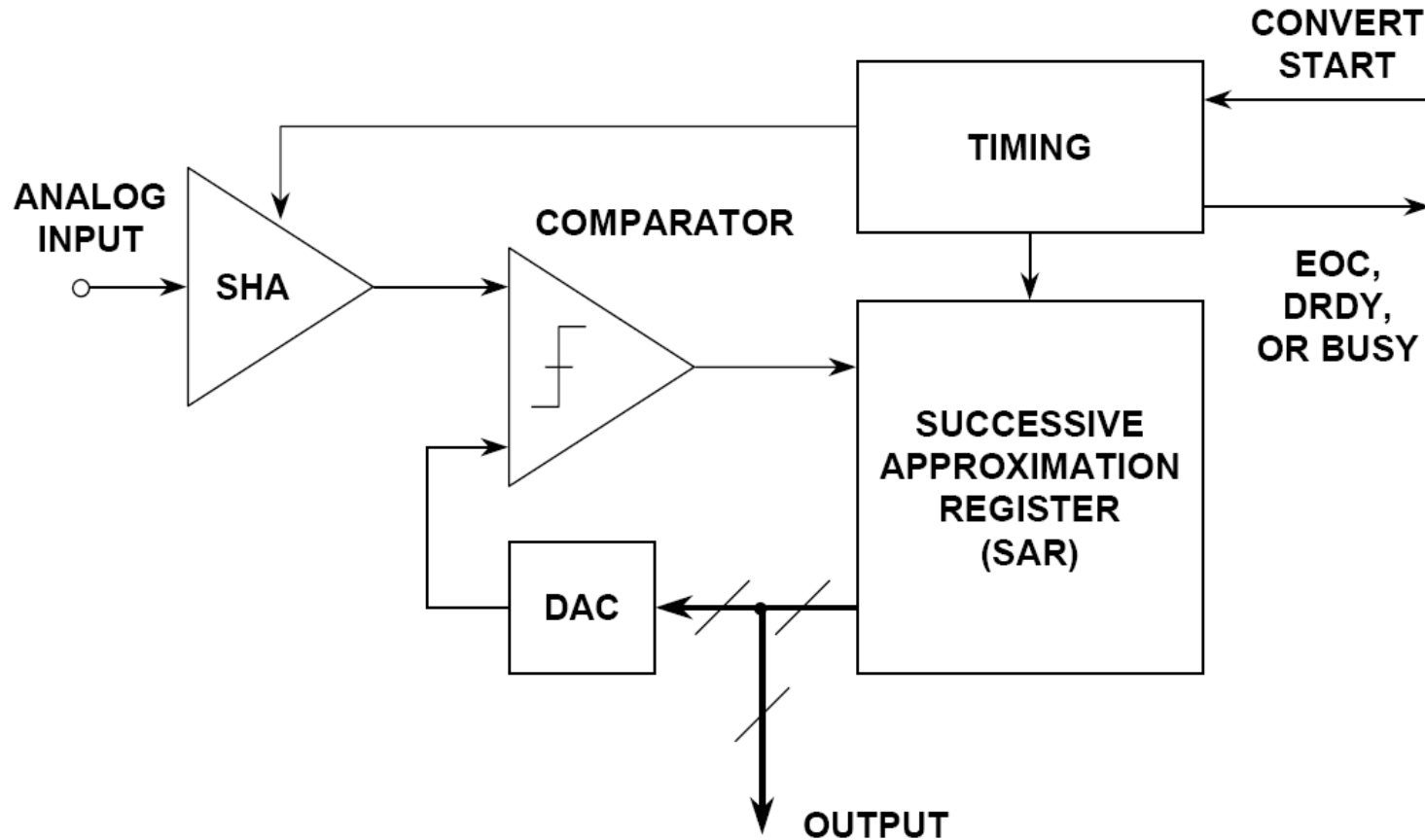


Kettős integrálós AD konverter



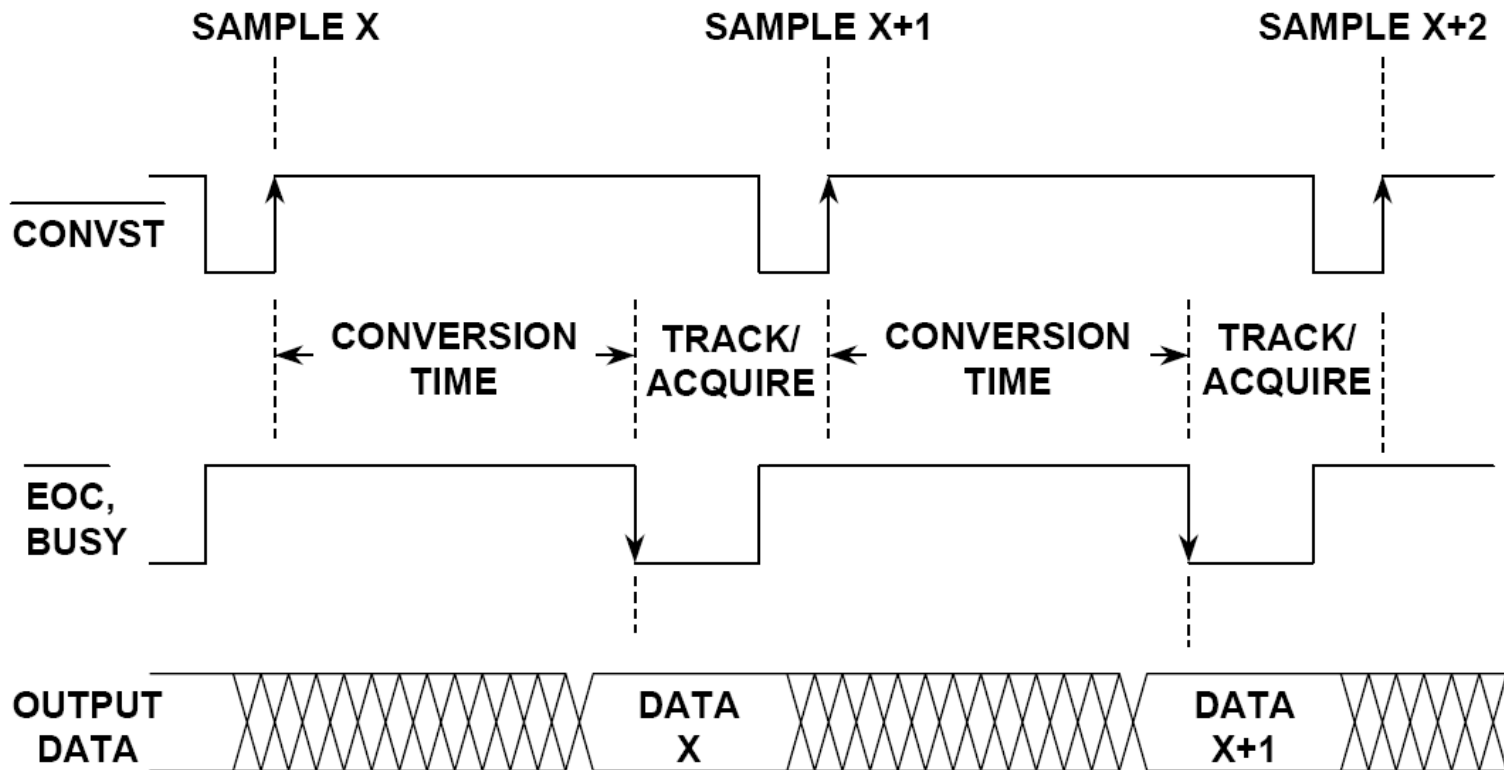
Szukcesszív approximációs (SAR) AD konverter

SUCCESSIVE APPROXIMATION ADC



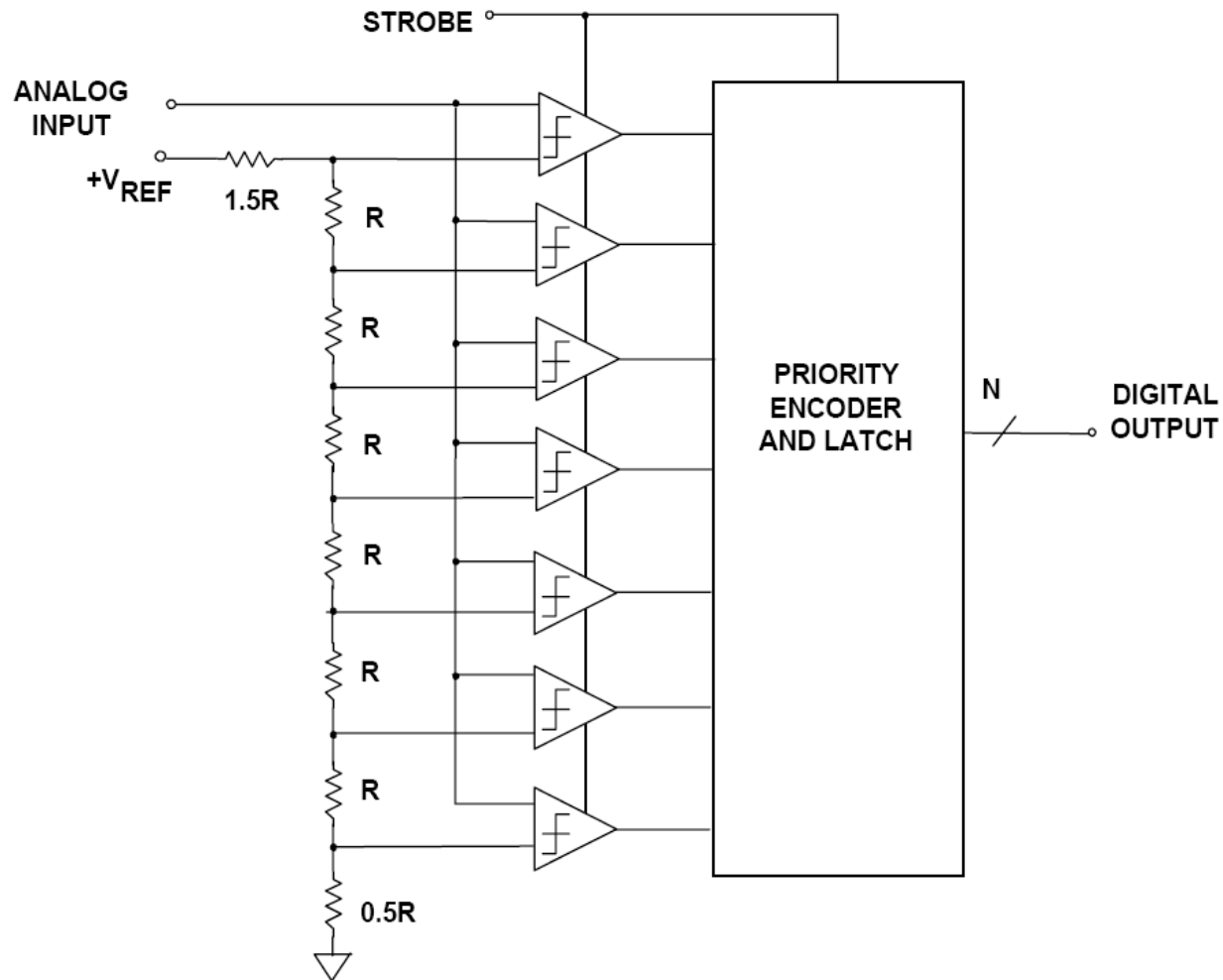
Tipikus SAR AD konverter időzítés

TYPICAL SAR ADC TIMING



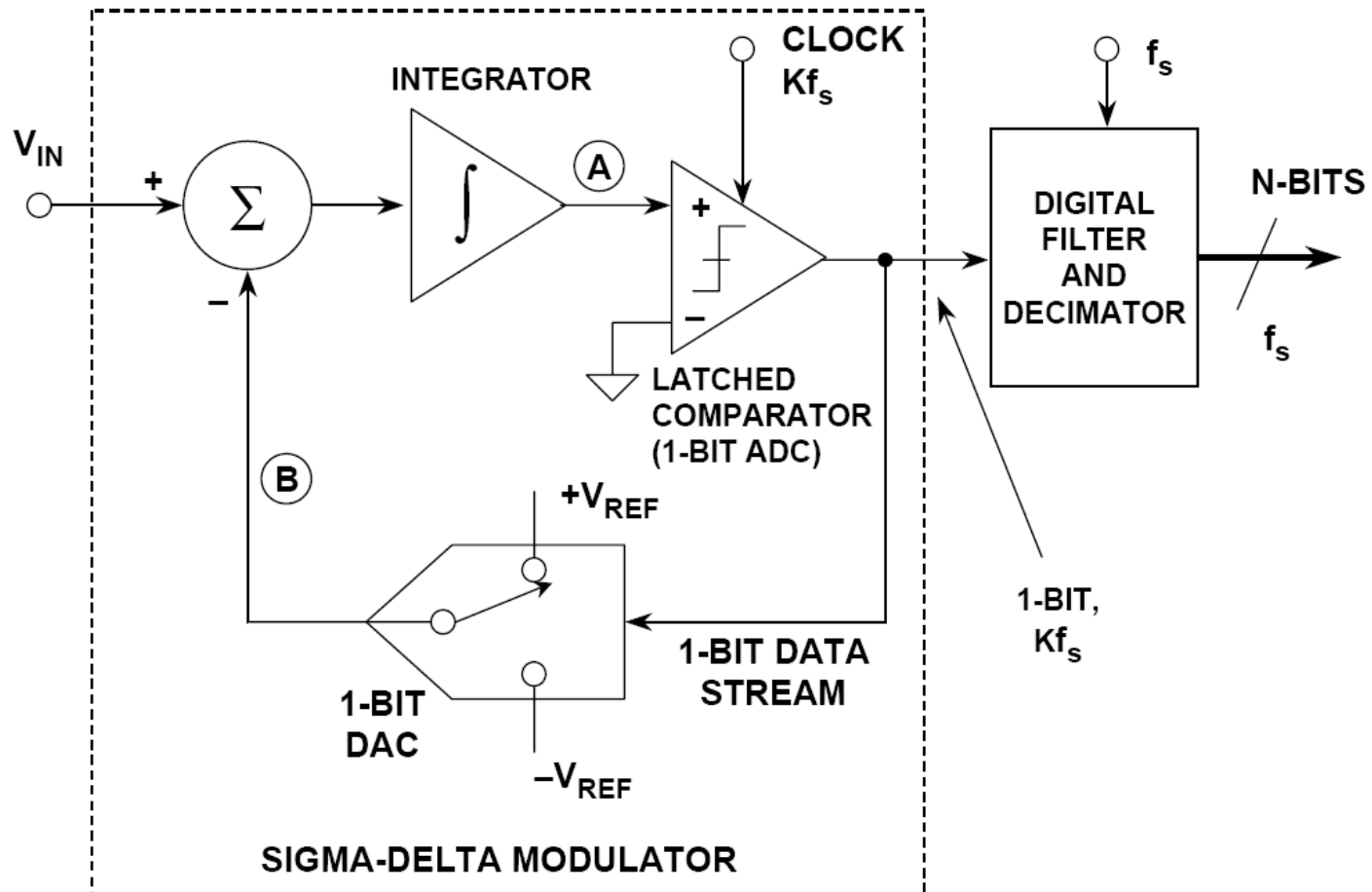
Paralell (Flash) AD konverter

FLASH OR PARALLEL ADC



1 bites szigma-delta konverter

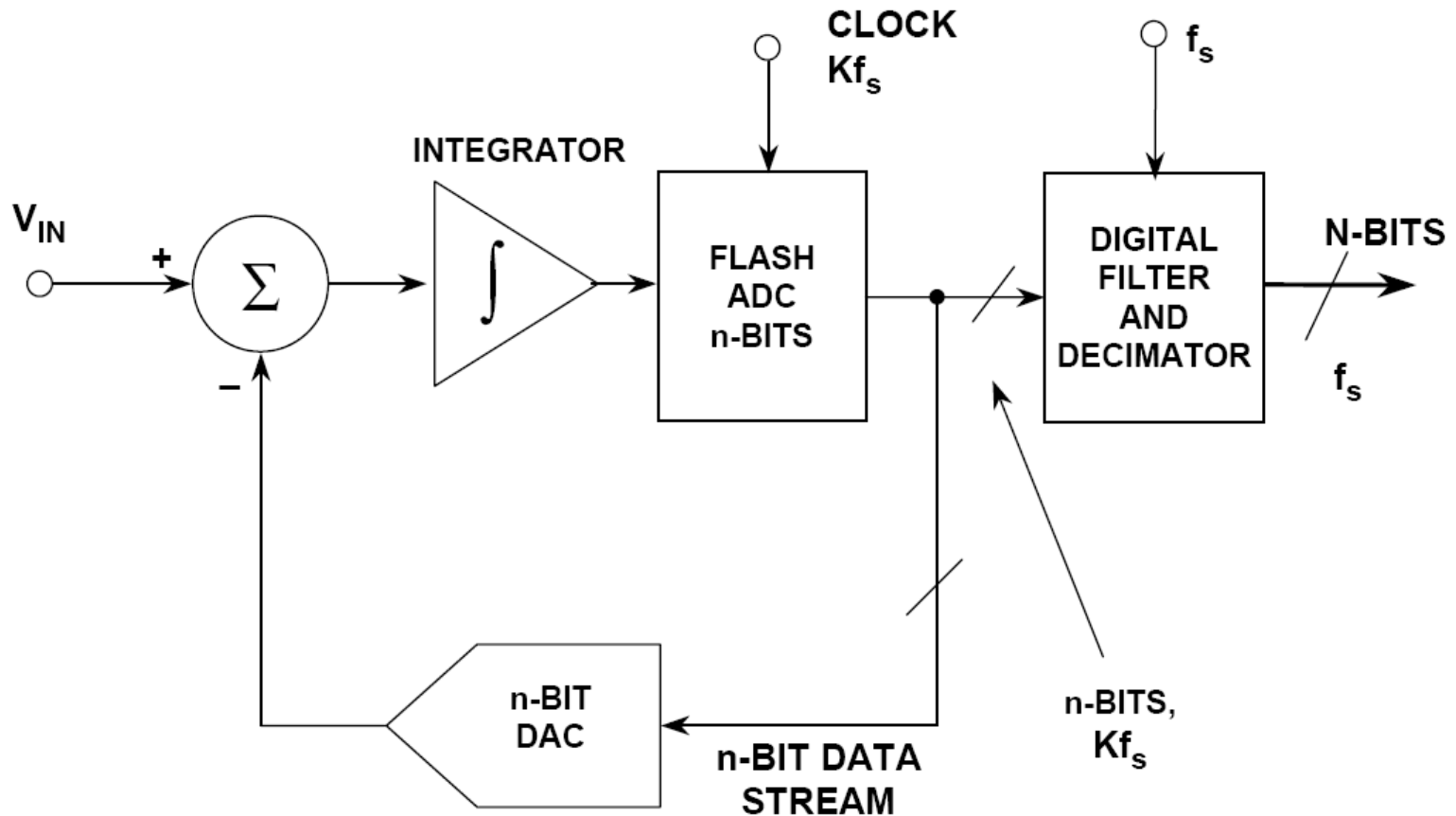
FIRST-ORDER SIGMA-DELTA ADC



Mi lesz a kimenő jel 0 V bemenő jel esetén?

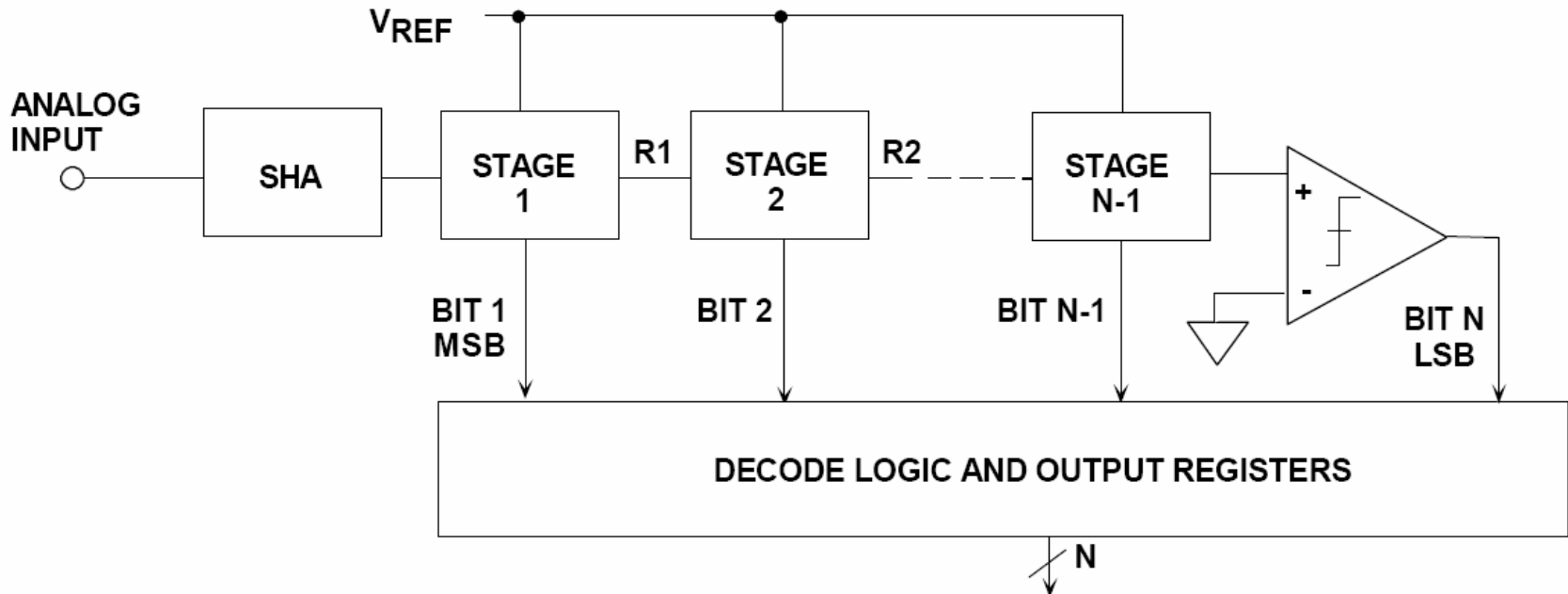
n bites szigma-delta konverter

MULTI-BIT SIGMA-DELTA ADC



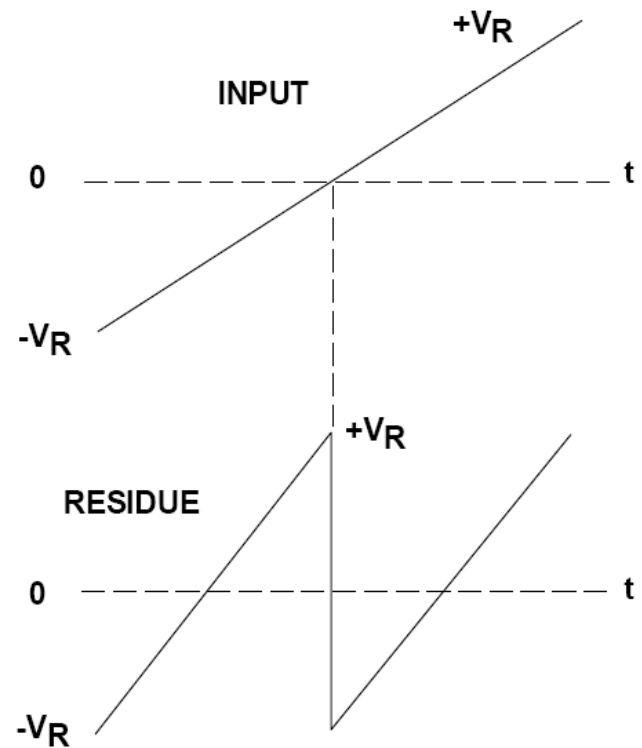
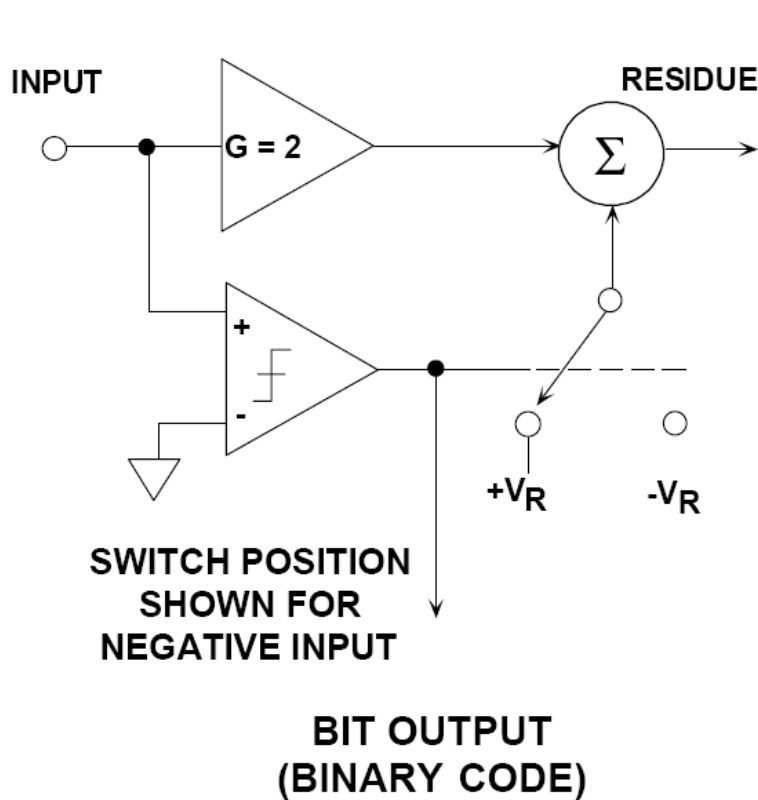
Soros elrendezésű konverterek

BIT-PER-STAGE, SERIAL, OR RIPPLE ADC



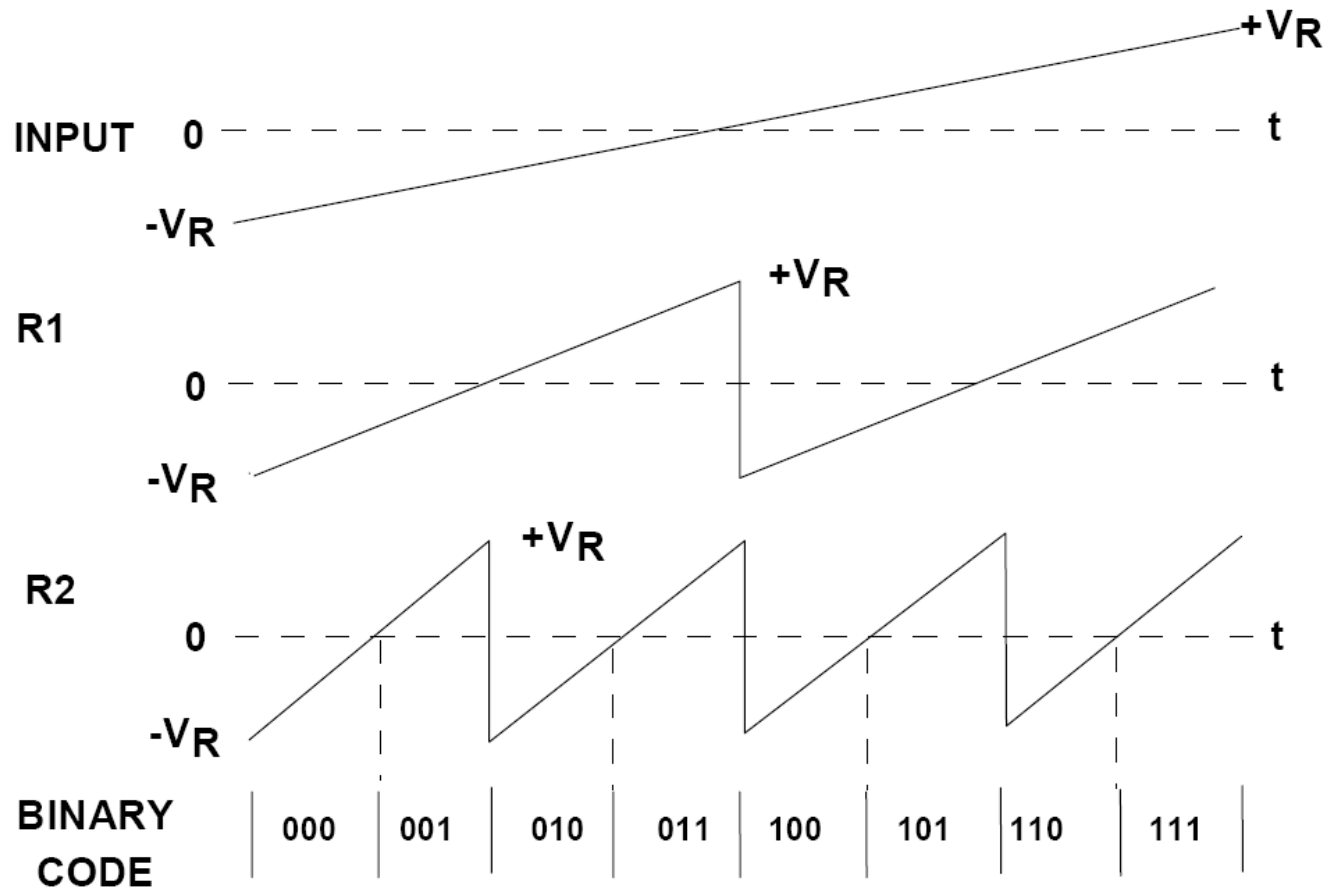
Egy fokozat kialakítása és működési elve

SINGLE-STAGE OF BINARY ADC



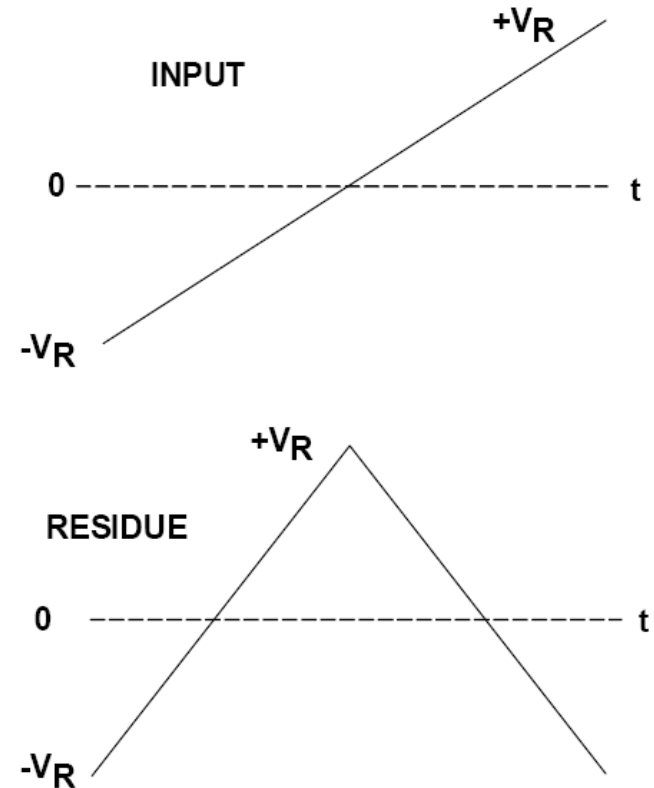
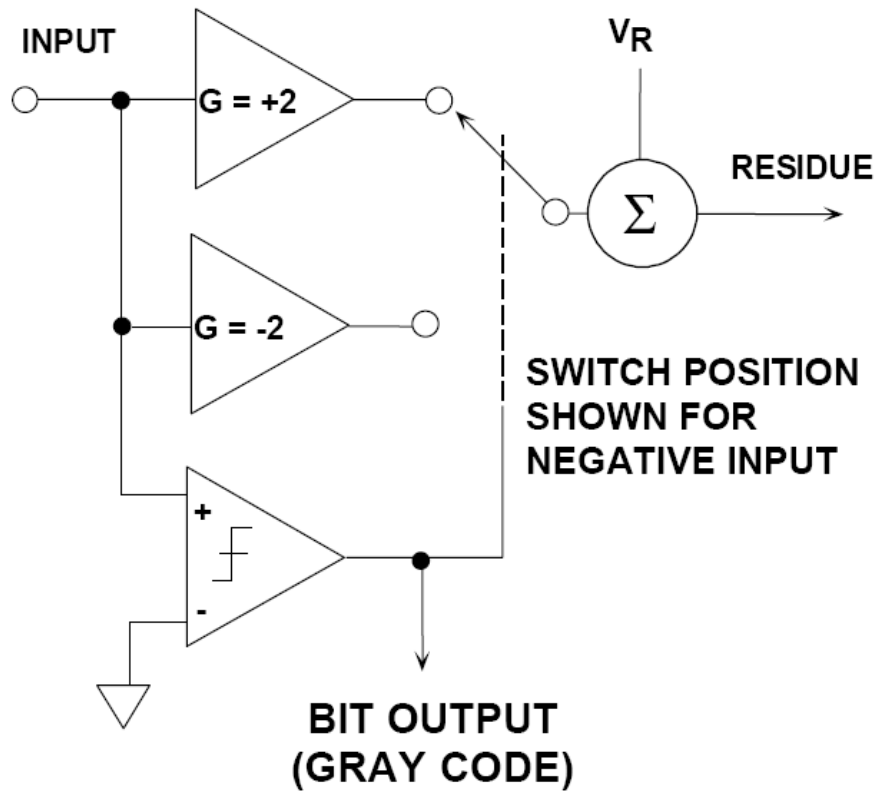
3 bites fokozat működési elve és a kimeneti kód

INPUT AND RESIDUE WAVEFORMS OF 3-BIT BINARY RIPPLE ADC



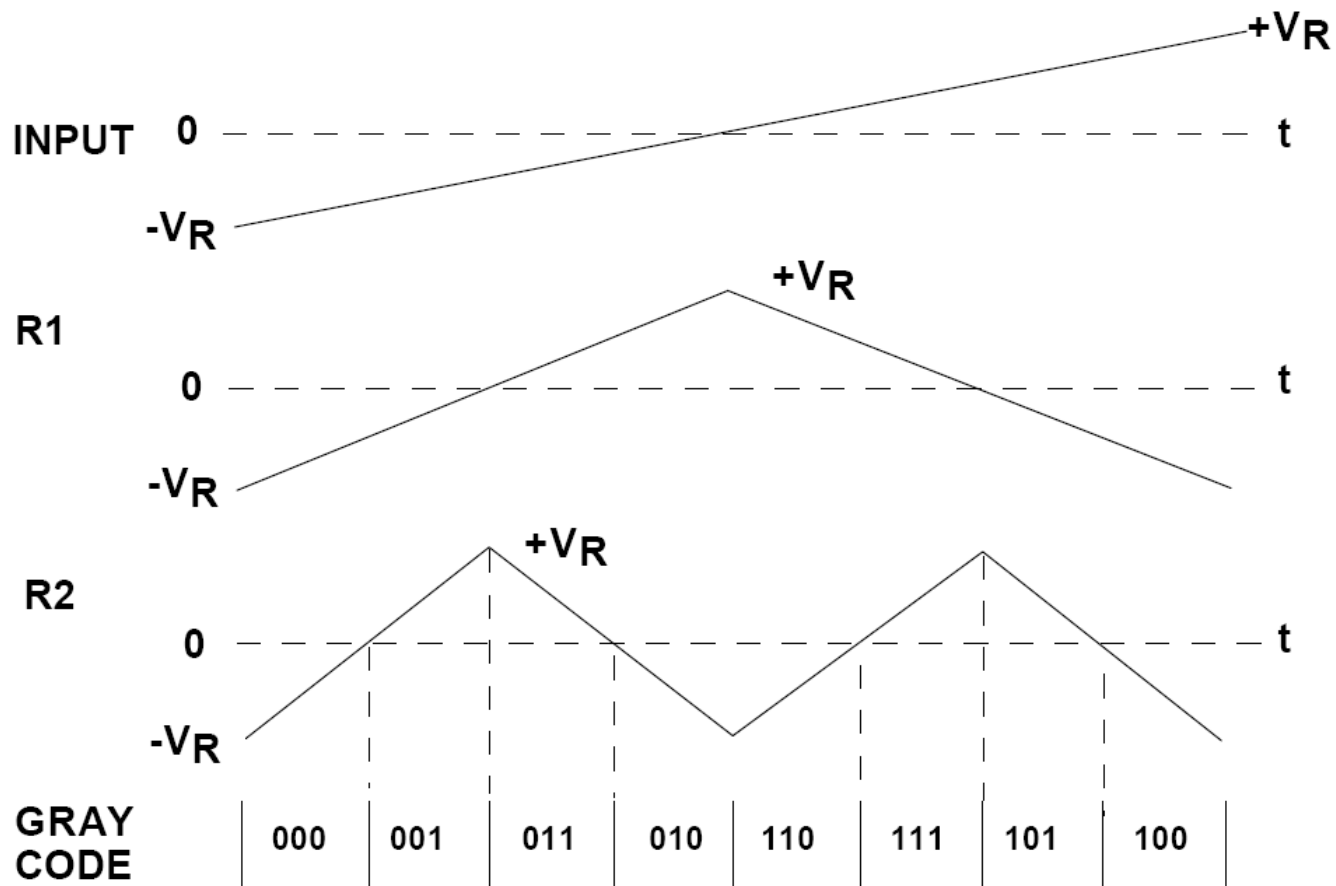
Egy fokozat másik megvalósítása

MagAmp STAGE FUNCTIONAL EQUIVALENT CIRCUIT



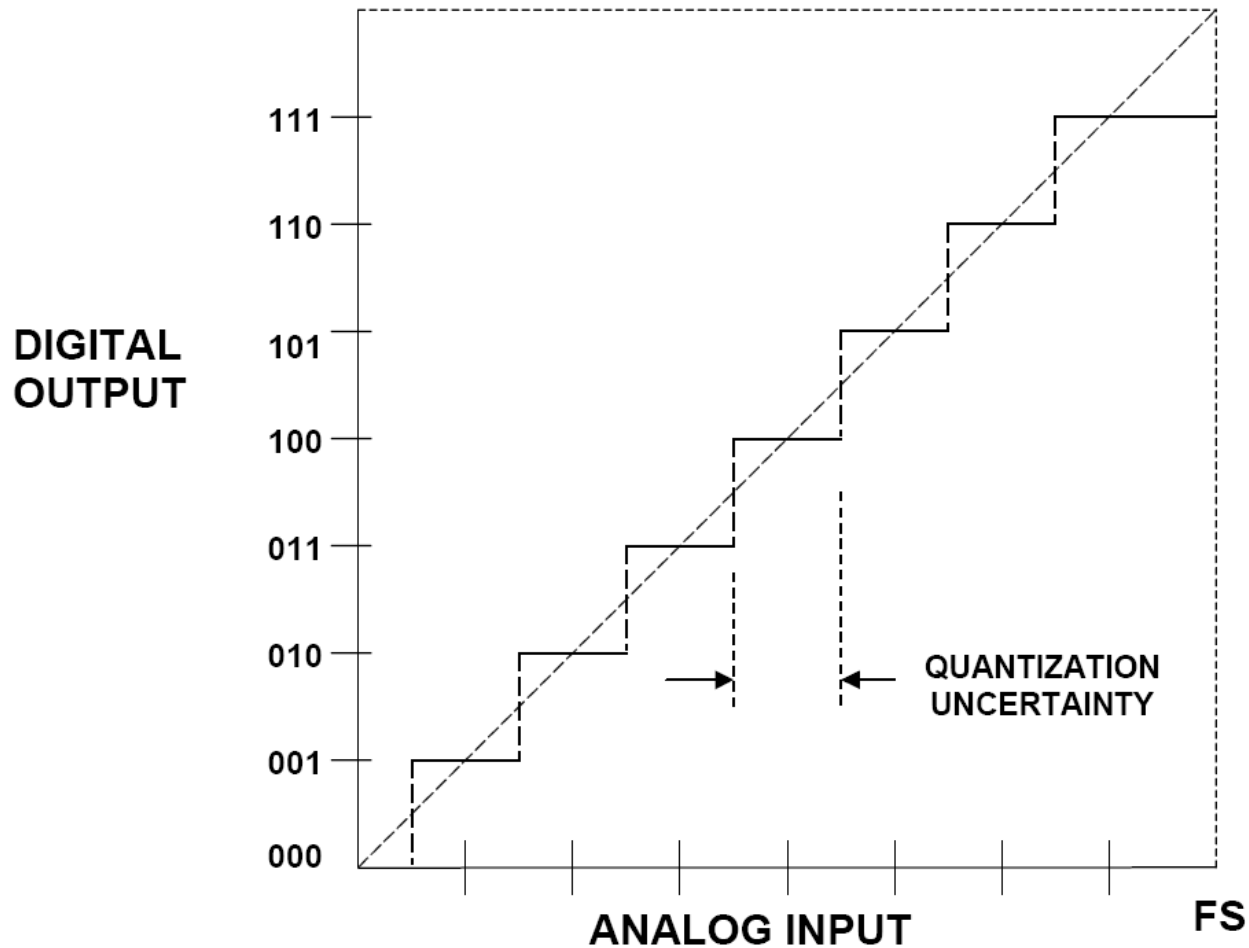
3 bites fokozat működési elve és a kimeneti kód

INPUT AND RESIDUE WAVEFORMS FOR 3-BIT MagAmp ADC



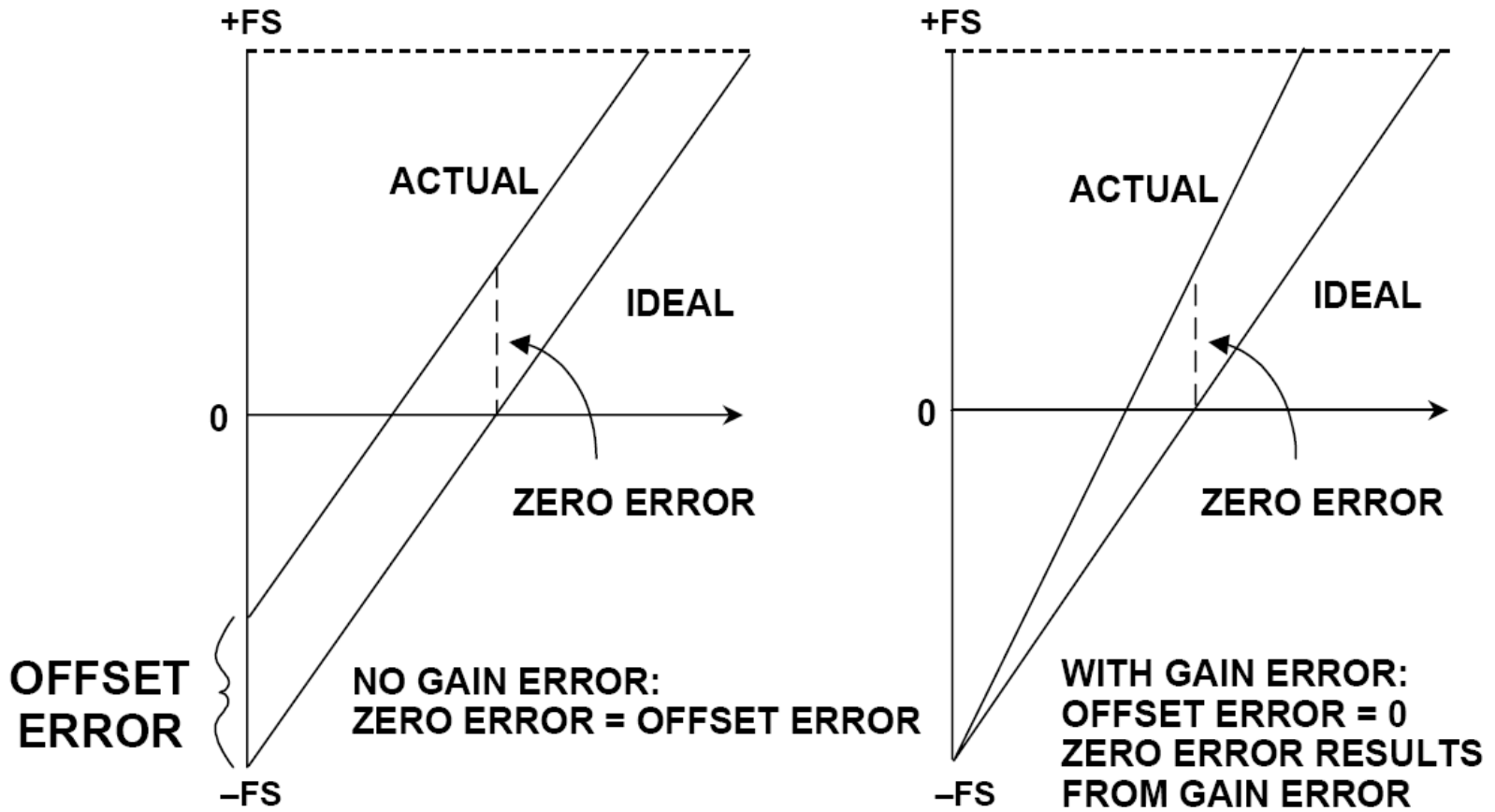
3 bites AD ideális átviteli függvénye

TRANSFER FUNCTION FOR IDEAL 3-BIT ADC



AD offset és erősítési hiba

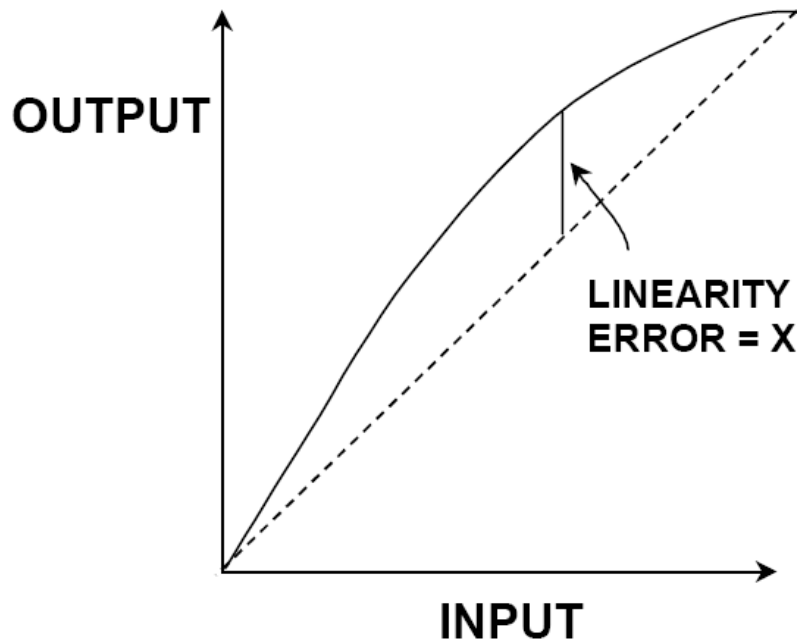
CONVERTER OFFSET AND GAIN ERROR



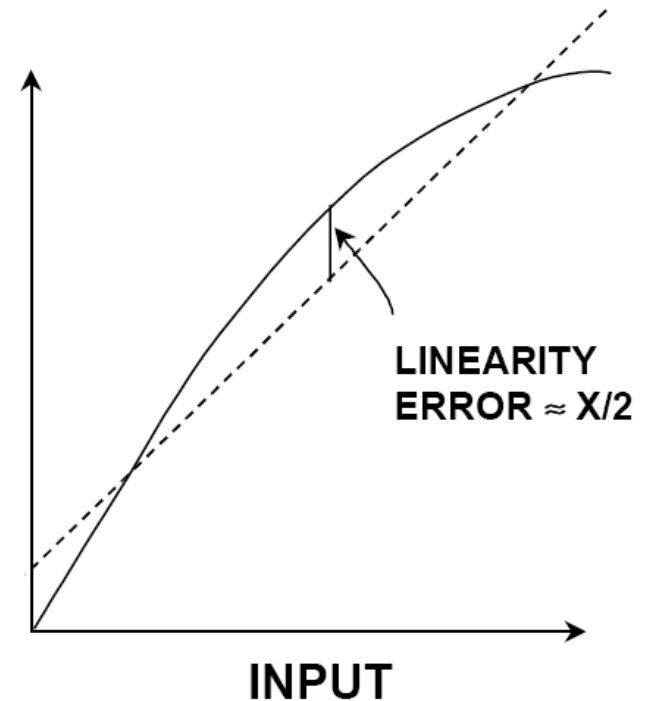
Linearitás hiba

METHOD OF MEASURING INTEGRAL LINEARITY ERRORS

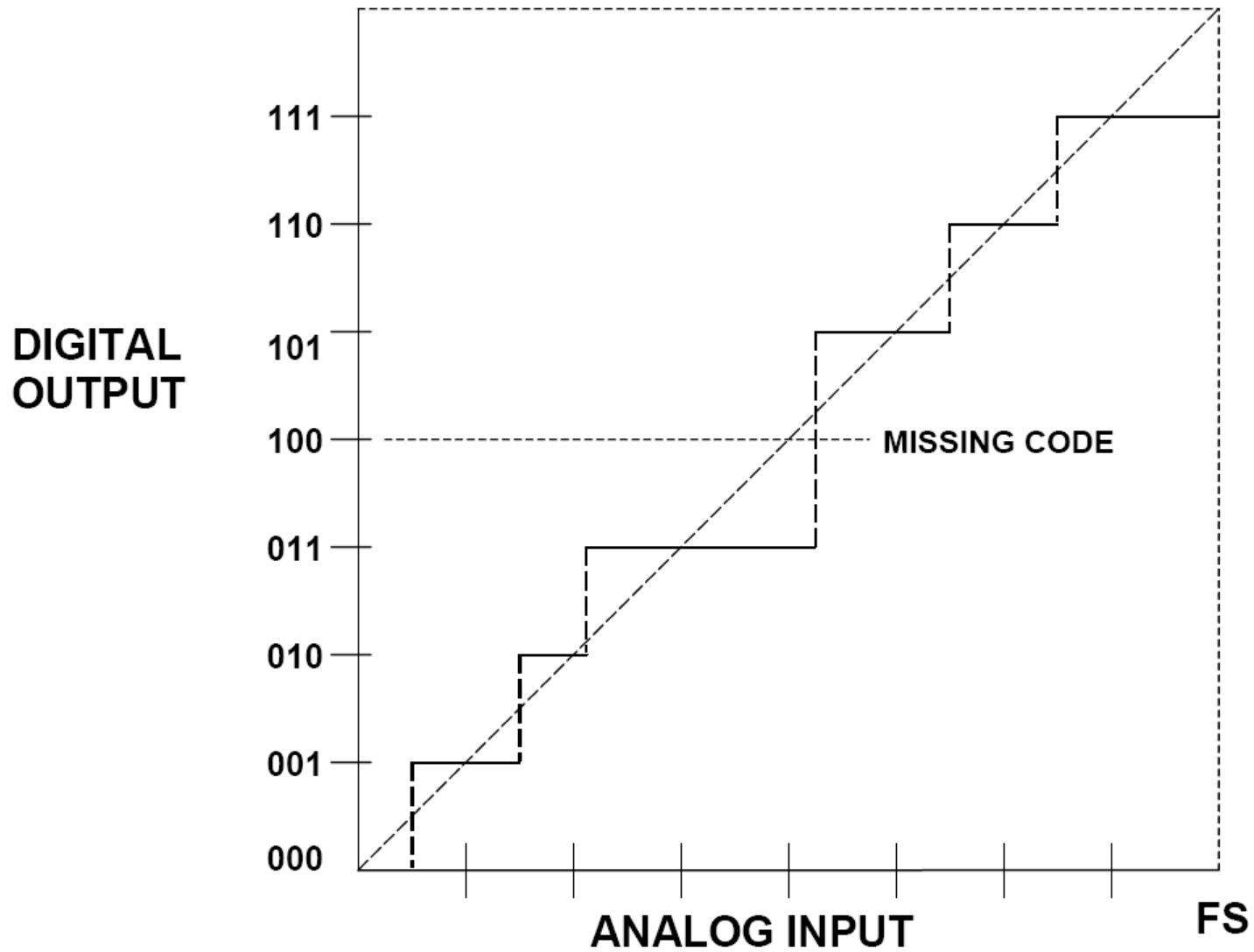
END POINT METHOD



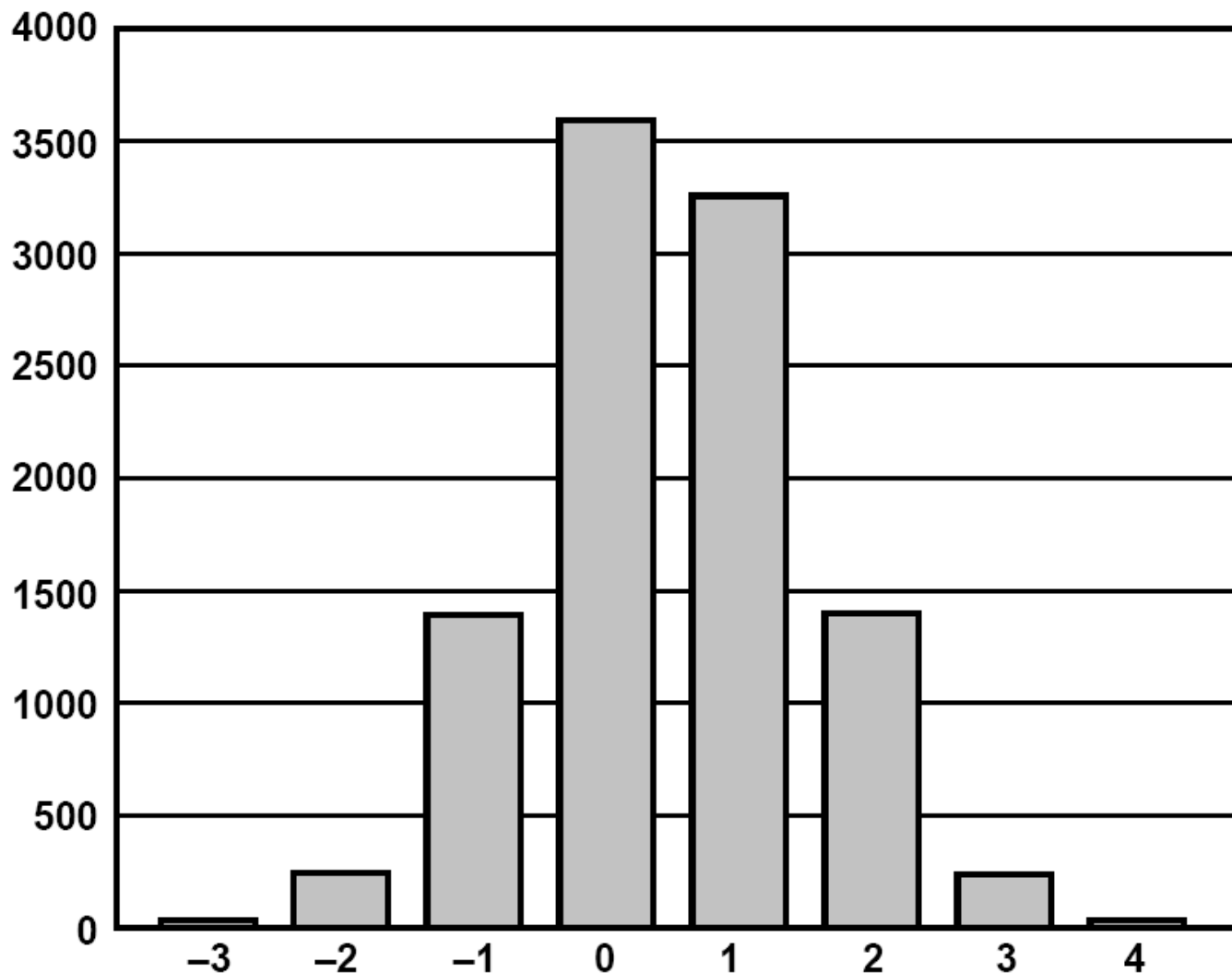
BEST STRAIGHT LINE METHOD



Hiányzó kód



AD976 típusú konverter hisztogramja



Histogram of 10,000 Conversions of a DC Input 27

Mintavevő-tartó áramkörök

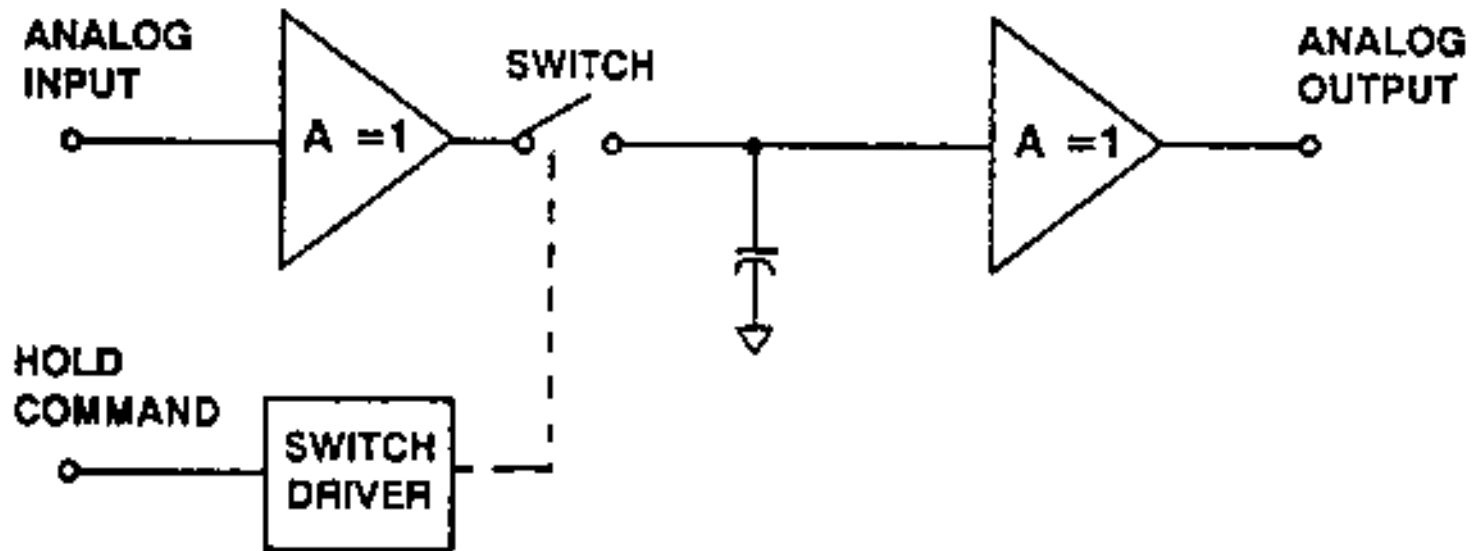
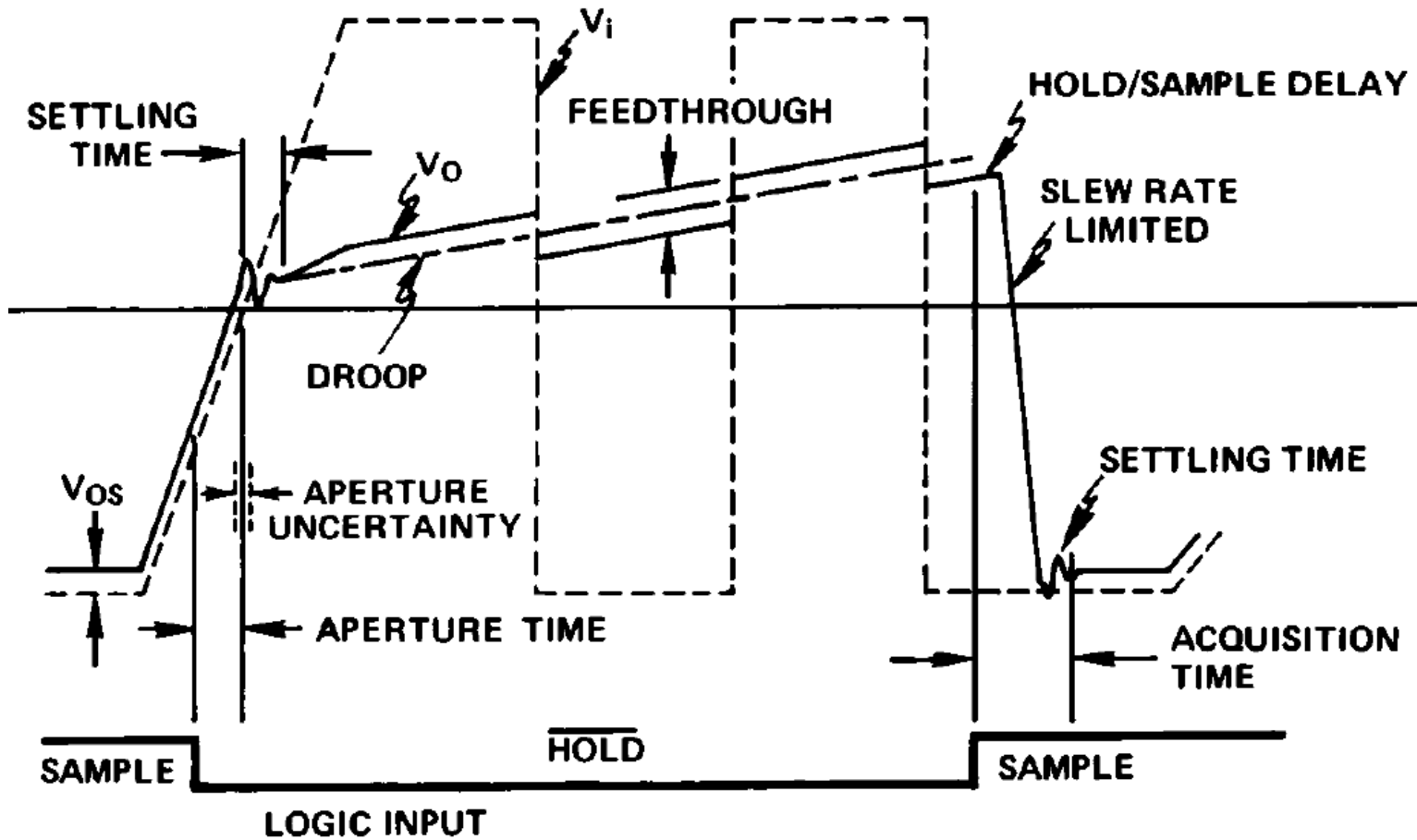


Figure 1. Basic Sample-and-Hold Amplifier

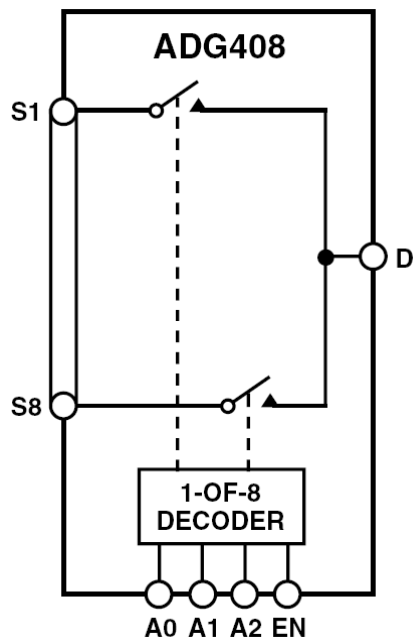
SH erősítők jellemző adatai



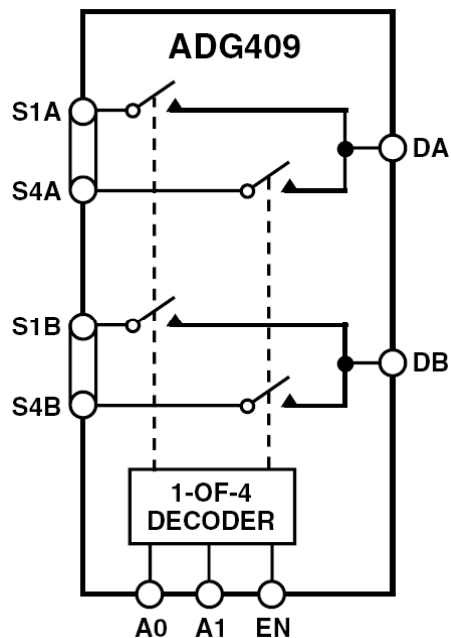
Sample Hold áramkör típusok

típus		alkalmazás	pontosság	Taq
			%	usec
AD386	SH	nagyon pontos	0.0007	4.5
AD389	SH	nagy felbontás	0.003	2.5
AD684	SH	gyors	0.01	1
HTS 0025	TH	igen gyors	0.01	0.025
AD582	SH	alacsony költs.	0.1	6

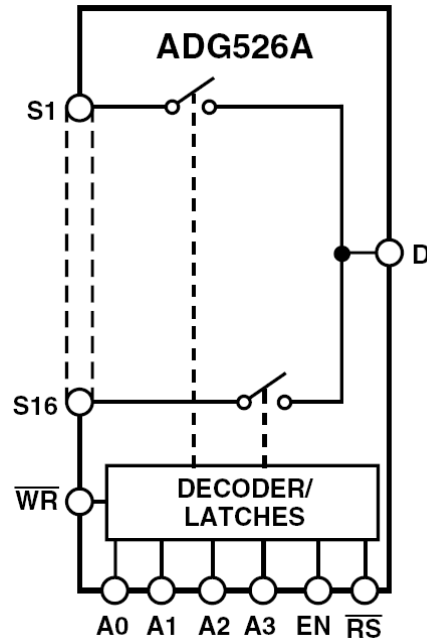
Analóg multiplexer típusok



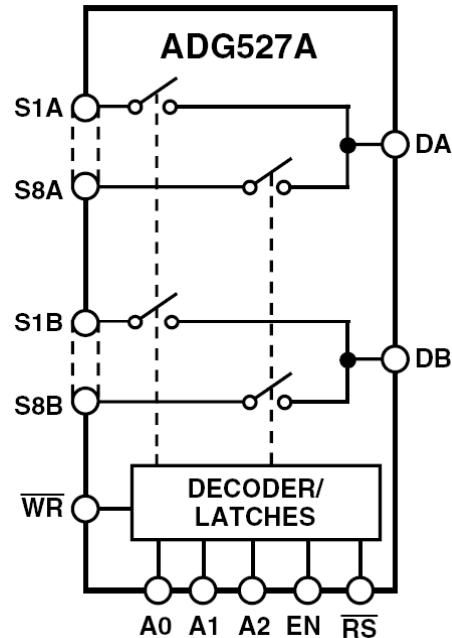
Szimpla
8 csatornás



dual
4 csatornás



szimpla
16 csatornás
+ lach



dual
8 csatornás
+lach

Analóg multiplexerek jellemző adatai:

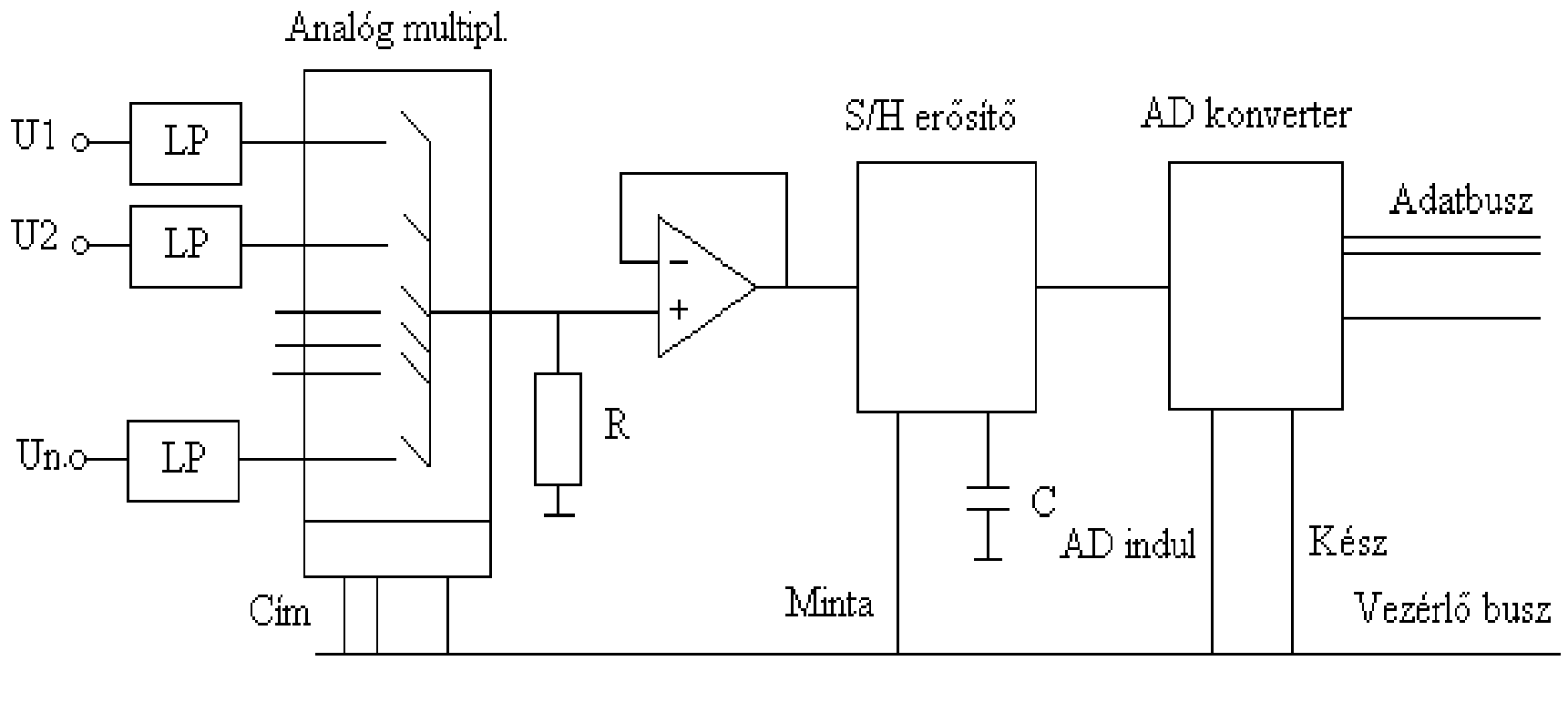
csatornaszám:	2,4,8,16
csatorna ellenállás:	50 .. 2000 Ohm
ellenállás szórása	5 .. 20%
kapcsolási idő:	0.1 .. 5 usec
maradékáram:	0.2 ..2 nA

Itt is nagyon fontos a beállási idő és a pontosság mint a SH áramköröknél

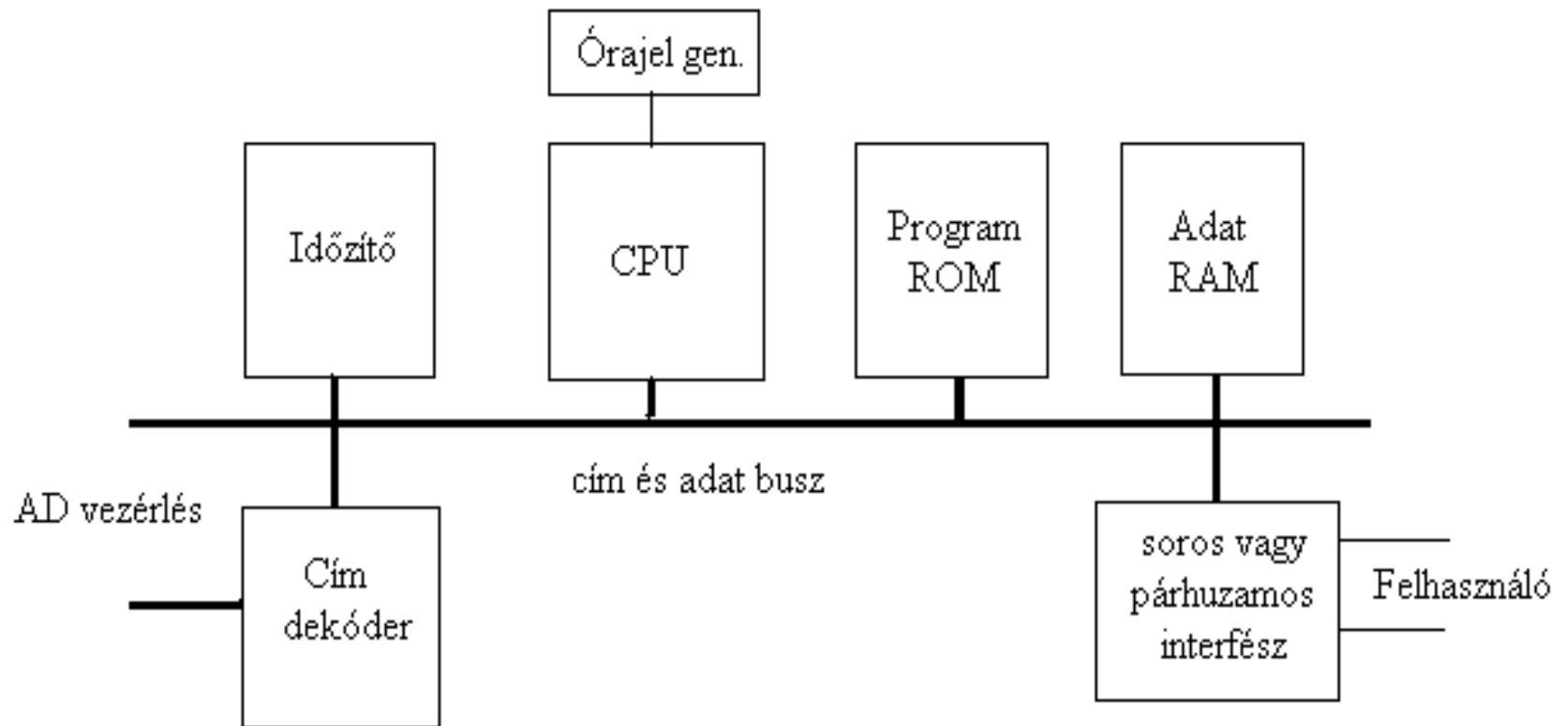
Analóg multiplexer típusok

típus	csatorna	csat. ell.	eltérés	Id off	ton
		Ohm	%	nA	nsec
ADG506	16 s	280	5	1	200
ADG507	8 d	280	5	1	200
ADG508	8 s	280	5	1	200
MCP508	8 s	1800		0,5	3500
ADG509	4 d	300	5	1	200

Többcsatornás mintavevő áramkör kialakítása



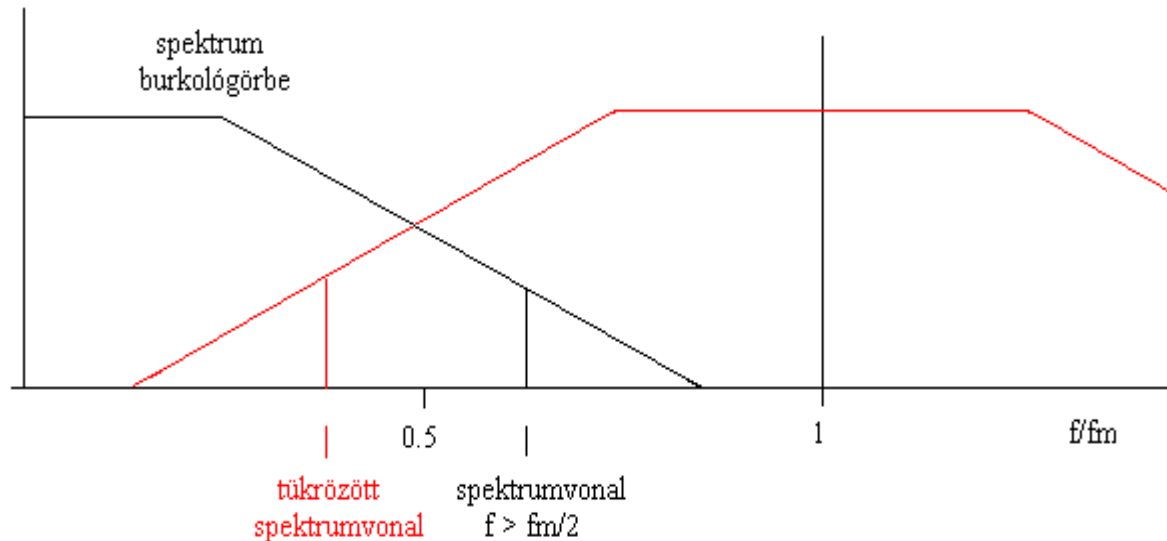
Többcsatornás mintavevő áramkör vezérlése mikroprocesszorral



Aliasing jelenség

Ha a bemenő jel tartalmaz az $f_m/2$ -nél nagyobb frekvenciájú összetevőket, akkor a spektrumban megjelennek hamis vonalak.

Ha $f > f_m/2$, akkor az $f_m - f$ frekvencián is kapunk spektrum vonalat (az $f_m/2$ frekvenciára tükrözve)

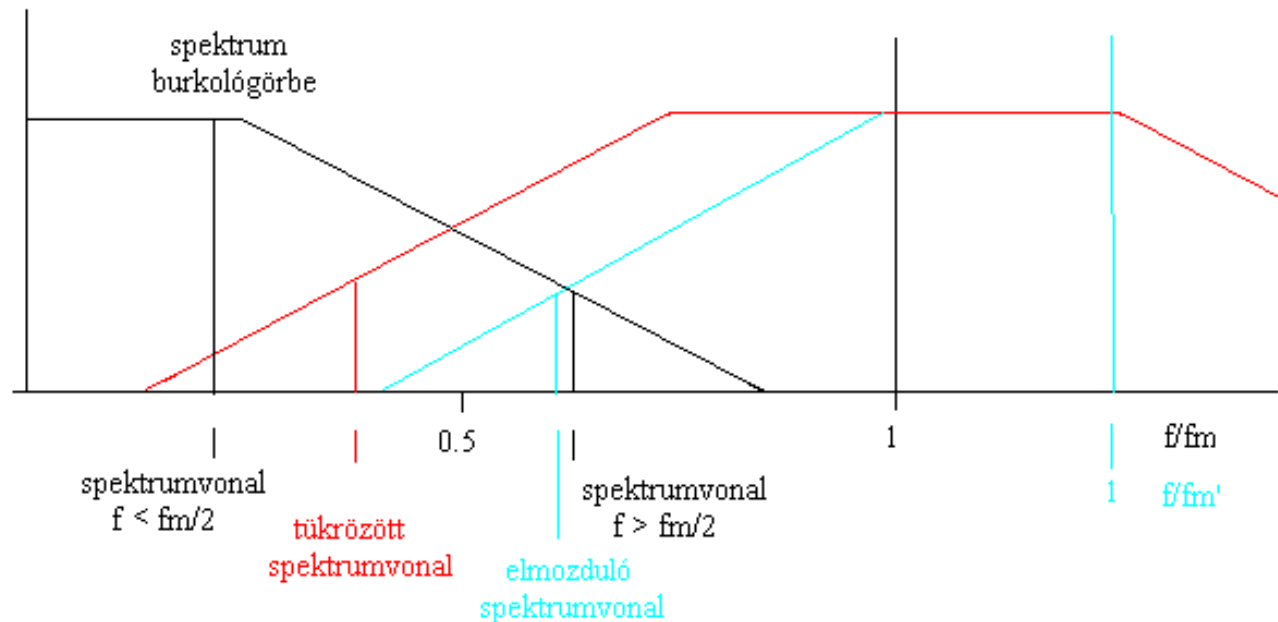


Aliasing jelenség korlátozása I.

1. Nagyobb mintavételi frekvencia alkalmazása, ha lehetséges.

A mintavételi frekvencia változtatásával megállapítható, hogy jelen vannak-e hibás spektrumvonalak.

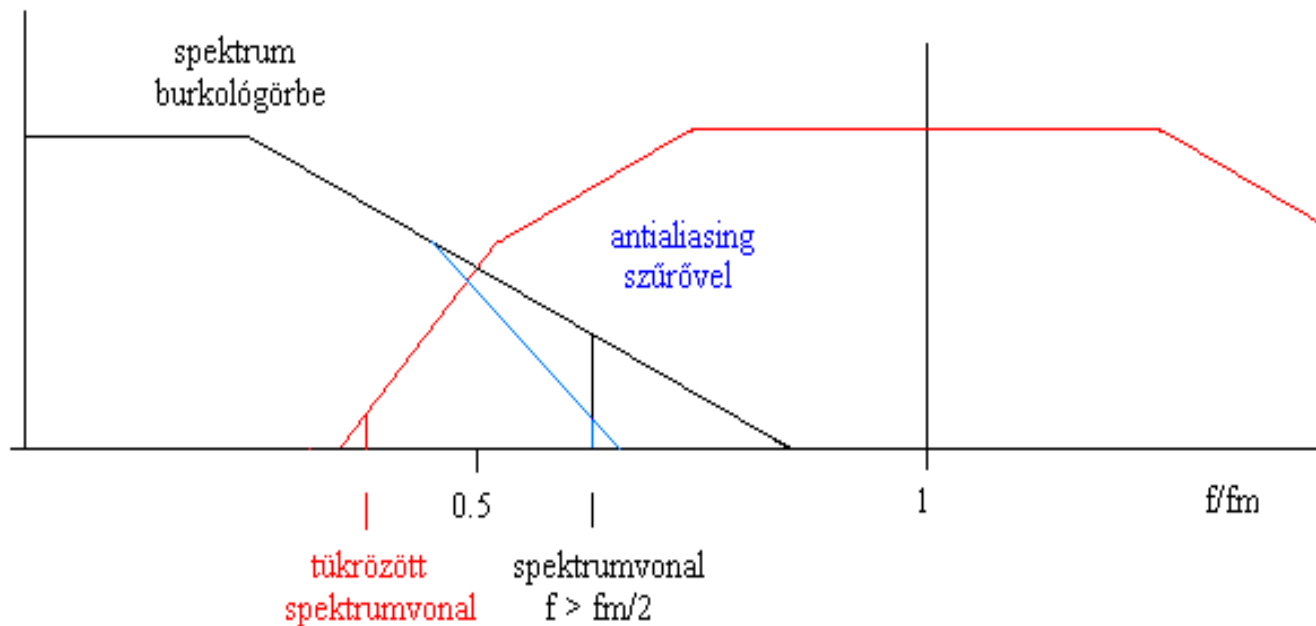
Pl. ha f_m és $2f_m$ frekvenciával is mintavételezünk akkor a hibás vonal $f_m - f$ frekvenciáról $2f_m - f \neq f_m - f$ helyre vándorol.



Aliasing jelenség korlátozása II.

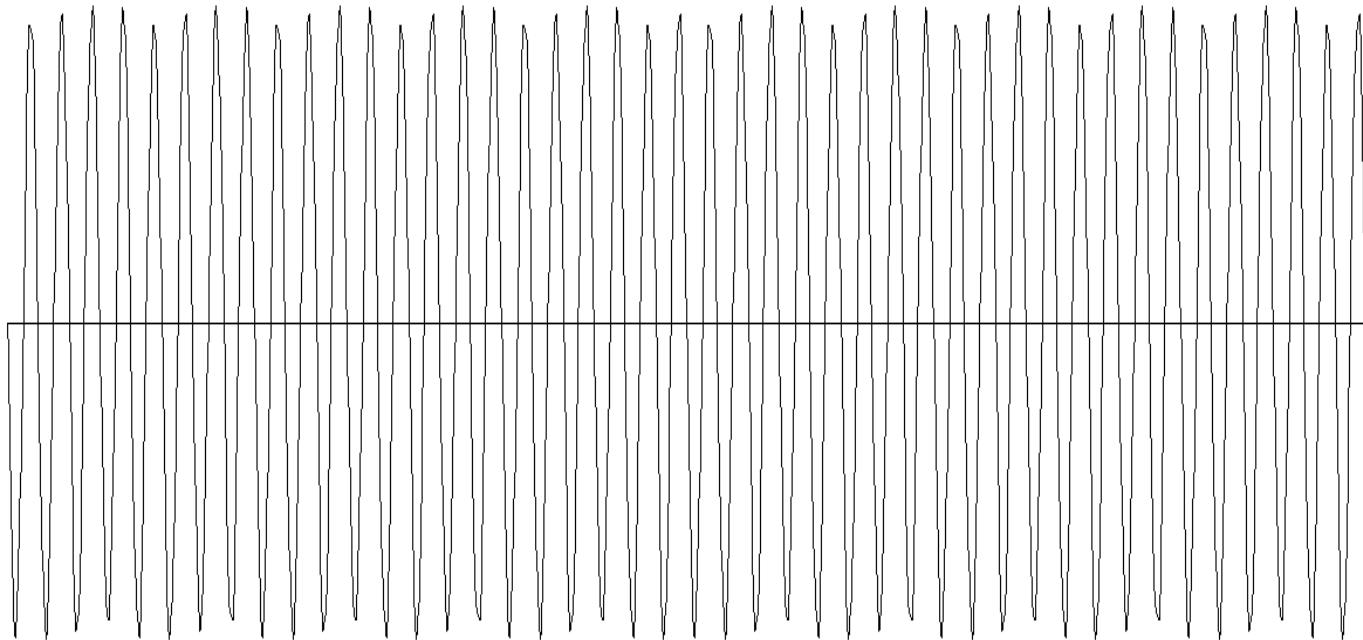
2. Az $f_m/2$ -nél nagyobb összetevőket a bemenő jelből eltávolítjuk aluláteresztő szűrővel (antialiasing szűrő).

Ez a szűrő természetesen csak az analóg oldalon, az AD konverter előtt lehet, és általában magas fokszámú szűrőket szokás alkalmazni.



Mintavételi frekvencia megválasztása

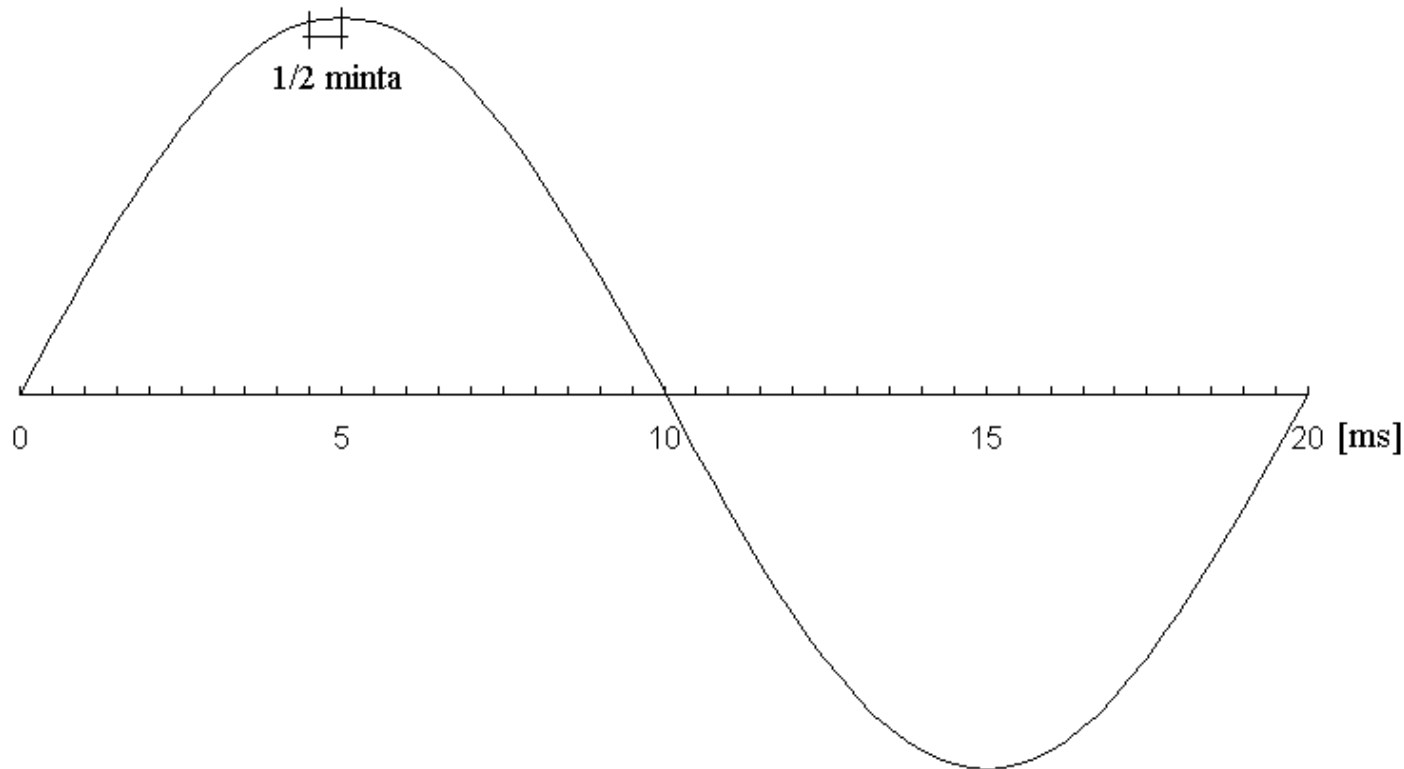
1 kHz-cel mintavételeztünk hálózati jelet és az alábbi ábrát kaptuk



Mi a jelenség neve és magyarázata?

Mekkora hibát okozhat, hogyan lehet elkerülni?

A lebegés jelenségének magyarázata és nagysága



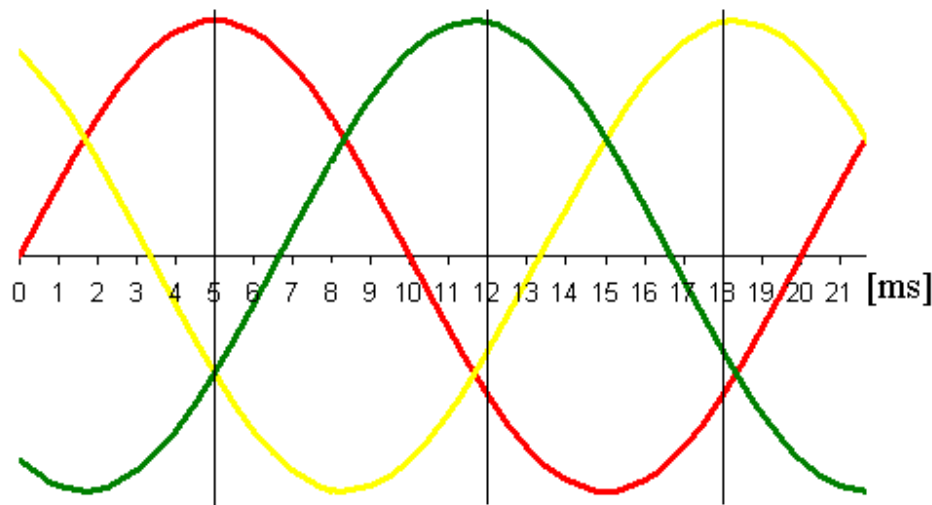
1 kHz = 20 minta/periódus = 18° /minta \rightarrow max. eltérés $\pm 9^\circ$

Ez **1,23 %** amplitúdó hibát eredményez a csúcserék közelében!!!

Felbontás hibája: 8 bit – 0,39%, 10 bit – 0,098%, 12 bit – 0,024%

Megoldás?

3 fázisú jelek mintavételezésének speciális esetei



Mintavételi frekvencia 1 kHz			
minta	5	12	18
a (piros)	1,000	-0,588	-0,588
b (sárga)	-0,500	-0,407	0,995
c (zöld)	-0,500	0,995	-0,407

Szinkron mintavételezés

Mintavételi frekvencia helyes megválasztása:

$$f = 2 * 3 * f_0 * n \quad n = 1, 2, 3, \dots$$

Változó f_0 mellett a mintavételi frekvencia fenti algoritmus szerinti megválasztása további problémát jelent.

Számábrázolás és alap algoritmusok mikroprocesszoros rendszerekben

Negatív számok ábrázolása I.

•*Előjel-abszolútértékes ábrázolás*

A negatív számok ábrázolásának ez a legegyszerűbb módja, hiszen egyszerűen 1-est írunk az előjel helyére.

Például: $N = -6 = 10000110$

Ezt az ábrázolásmódot BCD számok esetén, illetve a lebegőpontos számábrázolásban használják.

•*Inverz kódú (1-es komplement) ábrázolás*

Ez az ábrázolásmód is nagyon egyszerű. Az előjel helyére 1-et írunk a többi bitet pedig invertáljuk. Az előző példára alkalmazva:

$N = -6 = 11111001$

Az inverz kódú ábrázolást egyes lyukszalag vezérlésű szerszámgépekben alkalmazzák.

Negatív számok ábrázolása II.

- Mind az előjel-abszolútértékes mind az inverz kódú ábrázolás nagy hibája, hogy problematikus az összeadás és a kivonás műveletének elvégzése illetve az eredmény értelmezése.
- Mindkét ábrázolásmódban pozitív és negatív számként is ábrázolhatjuk a zérust. Ez sajnos azt jelenti, hogy nincs egy egyértelmű megfeleltetés a szimbólumsorozatok és az általuk ábrázolt szám között.

Ezeket a hibákat küszöböli ki a 2-es komplement ábrázolás.

A kettes komplement kódban ábrázolt bináris szám a legnagyobb helyértékű bitjét (az előjelbitet) negatív súlyozásúnak kell tekinteni. Egy bináris szám kettes komplementjét úgy képezzük, hogy a szám inverzéhez hozzáadunk 1-et.

2-es komplementes számábrázolás I.

- Negatív számok esetén MSB=1
- Műveletek helyes eredményt adnak amíg a 2-es komplementes körben maradunk

Honnan tudja a processzor, hogy 2-es komplementesben dolgozunk?

Mi történik összeadás és kivonás során?

2-es komplementes kódú összeadáskor, ha mindkét operandus negatív, mindig *keletkezik* átvitel, különböző előjelű operandusok esetén *keletkezhethet*, pozitív operandusok esetén *nem keletkezik* átvitel.

2-es komplement számábrázolás II.

Túlcsordulás?

$70h + 70h = E0h$ (0111 0000 + 0111 0000 = 1110 0000)

pozitív számok esetén igaz, de 2-es komplementben nem, mert két pozitív szám összege nem lehet negatív!

Honnan tudom, hogy ez nem jó? Kell-e vizsgálnom, hogy történt aritmetikai túlcsordulás vagy nem? Rá jövök erre valahogyan vagy nem?

Vagy eleve tudom, hogy nem kerülhet elő olyan szám, hogy ez előforduljon, vagy $AC(flag) = 1$ ha 2-es komplement eredmény hibás!

Vegyünk helyette 7F-t (telítésbe vittük) ez is hiba, de kisebb hiba mintha nem csinálnánk semmit, és az eredmény negatív.

A két **pozitív** operandus összege lehet **negatív**, és két **negatív** operandus összege lehet **pozitív**!

Számábrázolási módok összehasonlítása:

DEC	Unipoláris		Előjel absz. ért.		1-es komp.		2-es komp.	
	B	H	B	H	B	H	B	H
255	1111.1111	FF	-	-	-	-	-	-
127	0111.1111	7F	0111.1111	7F	0111.1111	7F	0111.1111	7F
+1	0000.0001	01	0000.0001	01	0000.0001	01	0000.0001	01
+0	0000.0000	00	0000.0000	00	0000.0000	00	0000.0000	00
-0	-	-	1000.0000	80	1111.1111	FF	0000.0000	00
-1	-	-	1000.0001	11	1111.1110	FE	1111.1111	FF
-127	-	-	1111.1111	FF	1000.0000	80	1000.0001	81
-128	-	-	-	-	-	-	1000.0000	80

Tört számok ábrázolása

- Hogyan ábrázolunk tört számot?
- Aritmetika nem tudja, programozó dönti el, hogy hova képzeli a tizedes pontot, és ez alapján számol.
- Összeadás kivonásnál nincs gond
- Szorzás esetén összeszorozom egész számként, és megfelelő helyre leteszem a tizedespontot hasonlóan a tízes számrendszerben papíron történő szorzáshoz

1.15-ös számábrázolás

DSP-k használják szorzás megkönnyítésére

16 bitből van 1 db előjel (MSB bit) és 15 db törtrész. Ha 0 van elől akkor pozitív a szám, ha 1 akkor negatív (előjel abszolút-értékes).

1	15
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

 -1 től +1-ig ábrázol (-1 LSB)

Szorzás után ha vissza akarunk térni a 16 bites számábrázolásra, akkor a tizedespontot helyre kell tenni.

A DSP-k ezt automatikusan megteszik, ha megmondom, hogy 1.15-ös ábrázolást használlok.

Létezik 32.0 ábrázolás is, amely 16 egészből és 16 törtrészből áll.

Lebegőpontos tárolási forma

A lebegőpontos számtárolási formánál a számok tárolása hatványkitevős formában, mégpedig többnyire nullára normalizált alakban történik. A lebegőpontos adattárolási forma mindig egy számpár, mantissza ($1 <$) és a karakterisztika (2^n) együttes tárolását és kezelését jelenti, mégpedig önmagukban fixpontosan, azaz a mantisszát fixpontos törtszámként, a karakterisztikát fixpontos egészszámként kezelve. A kitevő ábrázolása általában offset bináris (eltolt nullapontú, azaz egyszerű lineáris transzformációval a legkisebb -legnegatívabb- kitevőt zérussá transzformáljuk.) Így a kitevő mindig pozitív, ami nagyon egyszerűvé teszi két lebegőpontos szám összehasonlítását.

- Nagy a dinamika, nincs túl és alulcsordulás
- Számítás és futási idő igényes
- Összeadás a nehezebb, nem a szorzás,
- Szorzás után az eredményt normalizálni kell, hogy a mantissza elférjen az ábrázolási tartományban.

PIC 18-as sorozatra néhány egyszerűbb utasítása

f (adat memória) vagy w (munkaregiszter)

- Egyszerűbb byte utasítások:

f vagy w lehet =f , (f \leftrightarrow w, f \leftrightarrow f)

f vagy w lehet =f aritmetika művelet w-vel

w lehet =w aritmetikai művelet k konstanssal (konstans műveletek csak w-n keresztül érhetőek el!)

- Egyszerűbb bit műveletek

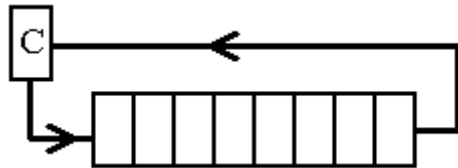
f,b (bármely címen bármely bit) =törlés, 1-be állítás,
komplement képzés, feltételes ugrások,

Példák egyszerűbb műveletekre

- f-regben 8 bites adatnak 2-vel való osztása



1-el jobbra tolom, de mi kerül az első helyre?



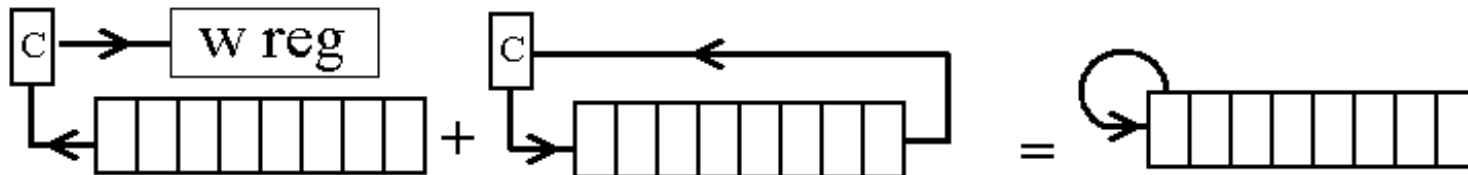
Pozitív számok esetén 0. Forgatás C-n keresztül. Első lépés C törlése.

Mit csinálunk előjeles szám esetén?

Pl.: Megvizsgáljuk az előjel bitet, és elágazás alapján állítom be C-t.

Egyszerűbb megoldás mindössze két lépésben:

1. balra forgatás C-n keresztül **w regbe**. (f értéke megmarad! Előjel bit tartalma bekerül C regbe)
2. Jobbra forgatás C-n keresztül f-be. (C-ből előjel bit vissza helyére)



Előjel kiterjesztés I.

8 bites 2-es komplement szám 16-32 bites rendszerben

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0							
---	--	--	--	--	--	--	--

ha pozitív szám volt, csupa 0-ás

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

1							
---	--	--	--	--	--	--	--

ha negatív szám volt, csupa 1-es

Pl.: 4 byte-os összegző tár esetén ha 2 byte után abba hagyom az összeadást, akkor negatív szám esetén az átvitel minden esetben elvész, tehát hibás eredményt fogok kapni! Ilyenkor feltétlen ki kell egészíteni.

Megoldás:

- 7. bit vizsgálat, és ennek függvényében elágazás, hogy wreg-be 0 vagy 1 kerül, majd wreg-et hozzáadom a tárhoz.

Előjel kiterjesztés II.

Carry vizsgálata:

add wreg, w

Az adatot a wreg-be teszem, hogy az eredeti megmaradjon, majd hozzáadom önmagához. (megcsinálható, mert wreg-nek címe van, memóriába ágyazott, nem hardver tár. A 16-os sorozatnál még nem lehetett megcsinálni, a 18-asnál már igen!)

Ha pozitív szám volt (MSB=0), akkor nem keletkezik átvitel, ha negatív szám volt (MSB=1) akkor minden esetben keletkezik átvitel és ennek függvényében elágazás, hogy wreg-be 0 vagy 1 kerül, majd wreg-et megfelelő helyre hozzáadom a tárhoz.

Szorzás konstanssal

2 hatványaival egyszerű szorozni, az csak balra siftelés Carry-n keresztül. (Carry-t előtte töröljük)

Mi van, ha nem kettő hatványával kell szorozni? Pl. 17-el?

Megszorozzuk 16-al és még hozzá adjuk egyszer.

Minden olyan számmal ami kevés 1-est tartalmaz bináris formában, azzal viszonylag gyorsan lehet szorozni. De mit csinálunk pl. 63-al való szorzás esetében

2 hatványaival szorzunk és utána összeadjuk.

$$X * 2^5 + X * 2^4 + X * 2^3 + X * 2^2 + X = 21 * X$$

Az assembler program esetében a futási idő kritikus lehet, ilyenkor használjuk ezt a szorzási eljárást, mert ez biztosan gyorsabb, mint bármely más algoritmus.

Osztás konstanssal

PIC-nek nincs osztás utasításuk. 8051-es sorozatnak van 8 bites előjel nélküli osztása két munka regiszterben, osztandó és osztó, eredmény ugyan ide hányados és maradékként.

DSP-knek nincs közvetlen 16 bite-s osztás utasításuk, 17 óraütem alatt csinálják meg.

2 hatványaival egyszerű osztani de egyéb számmal? Pl.: 13-al?

Átalakítjuk szorzássá. Szorzunk $1/13$ -al $=0,07693$, vagy ha pontosabban kell akkor $256/13=19.69$ -el, vagy $10000h/13=5041,1$ -el- Ez már igen pontos megoldás.

Természetesen az eredményt az eredeti értéknek megfelelően megszorozzuk, azaz elsifteljük.

Szorzás tetszőleges számmal

$$[f_{1+0} \ f_{1+1} \ f_{1+2} \ f_{1+3}] = [f_{20+} \ f_{2+1}] * [f_{3+0} \ f_{3+1}]$$

The diagram shows a 4-byte number $[f_{1+0} \ f_{1+1} \ f_{1+2} \ f_{1+3}]$ on the left, which is equal to the product of two 2-byte numbers $[f_{20+} \ f_{2+1}]$ and $[f_{3+0} \ f_{3+1}]$. Each number is represented by a horizontal bar divided into four segments.

Felbontás 1 byte-os rész szorzásokra:

- 1, $f_{1+2} \ f_{1+3}$

Diagram 1 shows the multiplication of the high two bytes of the first number (f_{1+2} and f_{1+3}) by the high two bytes of the second number (f_{2+1} and f_{3+1}). The result is a 4-byte number where the high two bytes are shaded.
- 2, $f_{1+0} \ f_{1+1}$

Diagram 2 shows the multiplication of the low two bytes of the first number (f_{1+0} and f_{1+1}) by the high two bytes of the second number (f_{2+0} and f_{3+0}). The result is a 4-byte number where the low two bytes are shaded.
- 3, $f_{1+1} \ f_{1+2}$

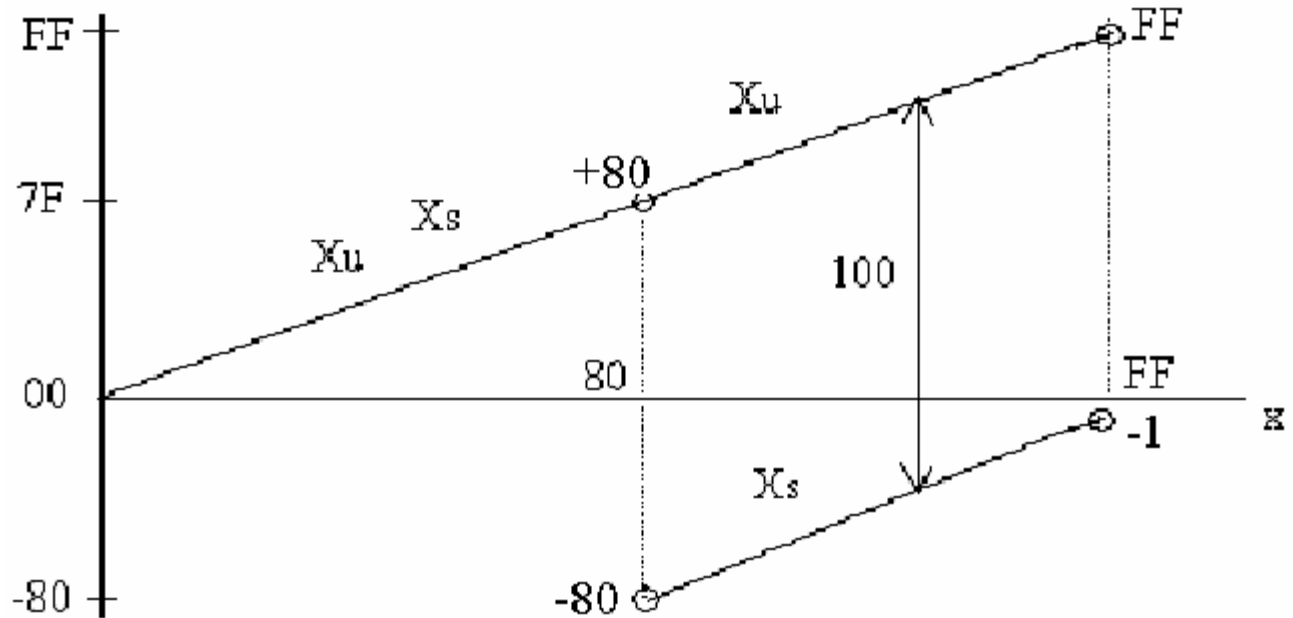
Diagram 3 shows the multiplication of the high two bytes of the first number (f_{1+1} and f_{1+2}) by the low two bytes of the second number (f_{2+1} and f_{3+0}). The result is a 4-byte number where the high two bytes are shaded.
- 4, $f_{1+1} \ f_{1+2}$

Diagram 4 shows the multiplication of the low two bytes of the first number (f_{1+1} and f_{1+2}) by the low two bytes of the second number (f_{2+0} and f_{3+1}). The result is a 4-byte number where the low two bytes are shaded.

A szorzat a négy érték összege lesz. Természetesen az első két tagot nem adjuk össze, csak a byte-kat helyre tesszük.

Előjeles szorzás (8 biten)

	unipoláris	2-es kompl.
+FS	1111.1111	-
+FS-LSB	1111.1110	0111.1111
1 LSB	0000.0001	0000.0001
0	-	0000.0000
-1 LSB	-	1111.1111
-FS+LSB	-	1000.0001
-FS	-	1000.0000



Kettes komplementum szám értelmezése, ha $x=a$ byte értéke:

$X_s = X_u$ ha $x < 80h$ azaz pozitív, és

$X_s = X_u - 100h$ ha $x \geq 80h$ azaz negatív

1. Legyen $X_s < 0$ és $Y_s \geq 0$, ekkor

$$M = X_s * Y_s = (X_u - 100h) * Y_u = X_u * Y_u - 100h * Y_u$$

2. Legyen $X_s \geq 0$ és $Y_s < 0$, ekkor

$$M = X_s * Y_s = X_u * (Y_u - 100h) = X_u * Y_u - 100h * X_u$$

3. Legyen $X_s < 0$ és $Y_s < 0$, ekkor

$$\begin{aligned} M &= X_s * Y_s = (X_u - 100h) * (Y_u - 100h) = \\ &= X_u * Y_u - 100h * Y_u - 100h * X_u + 10000h \end{aligned}$$

Az előjeles szorzás ($X_s * Y_s$) algoritmus:

($M_u = X_u * Y_u$ az előjel nélküli szorzás)

ha $X < 0$ akkor $M_s = M_u - 100h * Y_s$

ha $Y < 0$ akkor $M_s = M_u - 100h * X_s$

ha $X < 0$ és $Y < 0$ akkor $M_s = M_u - 100h * Y_s - 100h * X_s$

A $10000h$ hozzáadása elhagyható, mert kilóg a számtartományból!!

Osztás tetszőleges számmal

- Legegyszerűbb megoldás a többszöri kivonás, leállok ha az eredmény negatív (ezért csökkenthetem is 1-el az eredményt)
 - lassú
 - 0-val való „osztás” végtelen ciklust eredményez. (A ciklus előtt meg kell vizsgálni, hogy az osztó értéke 0-e. Ilyenkor valami elágazást csinálunk a programba, pl. az eredménybe az ábrázolható legnagyobb számot tesszük, csupa 1-el feltöltöm)
- Gyorsabb megoldás „hagyományos kézi osztás”
 - Nincs 0-val való osztás problémája
 - Ciklusszervezéssel nincs végtelen tizedes tört probléma
 - Ha osztandó $>$ osztó, akkor $\text{osztó} = 256 * \text{osztó}$

Mikroprocesszorok

1. Mikroprocesszor
2. Mikrokontroller
3. Digitális szignál processzor

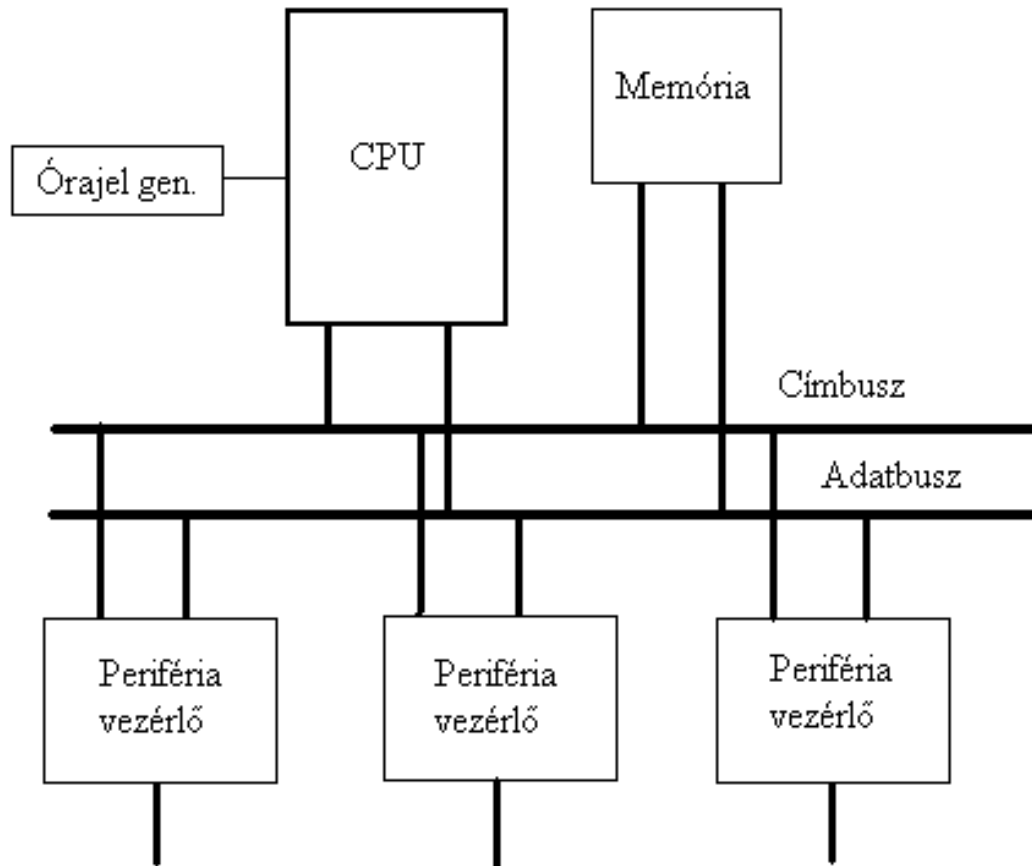
Irodalom:

www.microchip.com

www.chipcad.hu

Dr. Kónya László: PIC mikrovezérlők alkalmazástechnikája

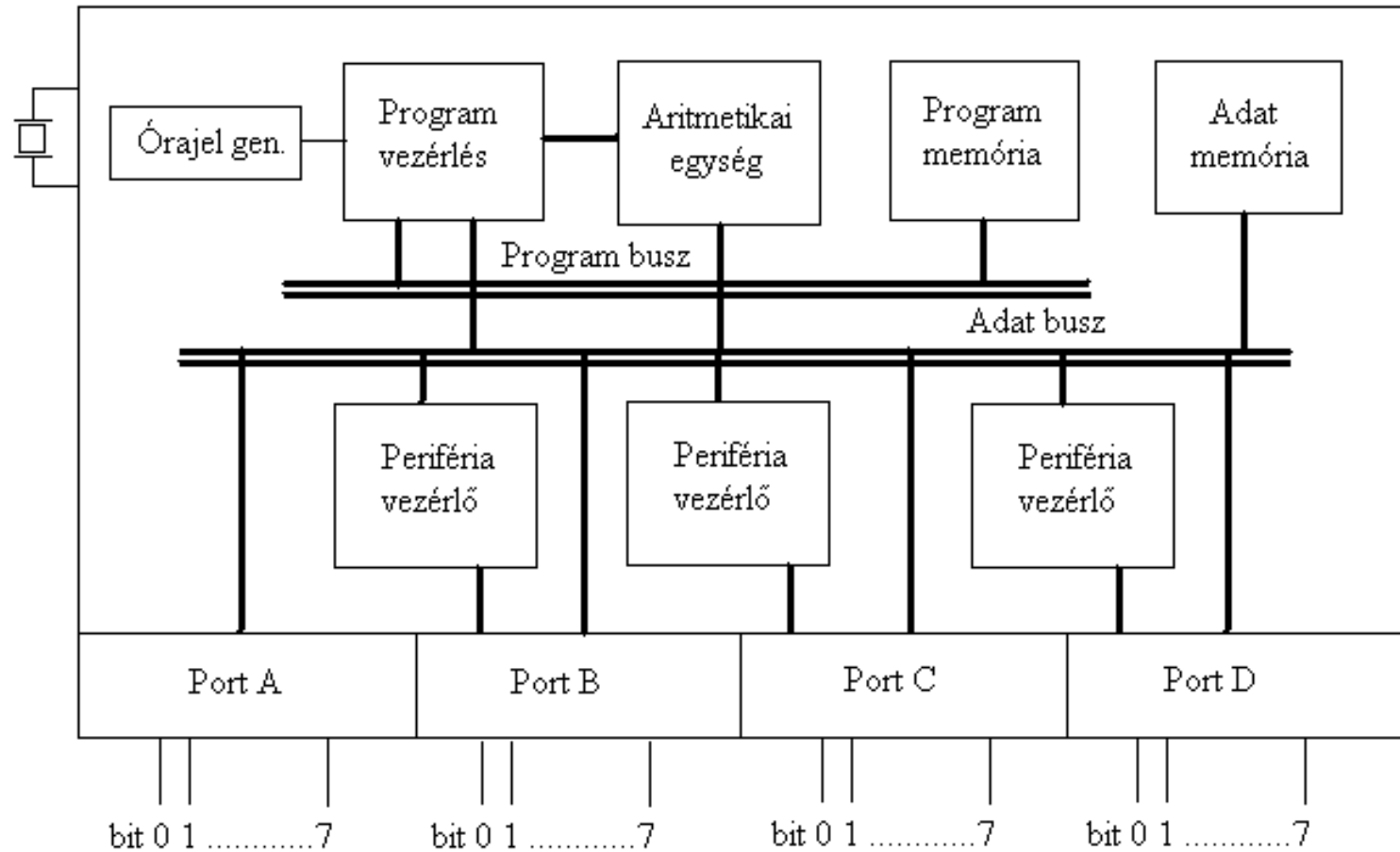
Mikroprocesszoros rendszer sémája



1. Hagyományos mikroprocesszor

- közös adat és program busz
- közös adat és program memória
- külön periféria áramkörök (timer, interr. vezérlés, portok)
- külső kapcsolat a buszrendszeren keresztül (adat, cím ...)
- komplex utasításkészlet - **CISC** (**C**omplex **I**nstruction **S**et **C**omputer - Komplex Utasításkészletű Számítógép)
- alkalmazás: kis és nagy teljesítményű számítógépek
- pl. 8086, Z80, 286, 386, 486, pentium ...
- 8 .. 64 bites busz
- 4MHz ..2 GHz... órajel

2. Mikrokontroller felépítése



2. Mikrokontrollerek

- külön adat és program buszok amelyek nincsenek kivezetve (Harvard architektúra)
- külön adat és program memória (nem keverhető!)
- beépített periféria áramkörök (timer, interr. vezérlés, portok)
- külső kapcsolat a portokon keresztül
- egyszerű utasításkészlet - **RISC** (**R**educed **I**nstruction **S**et **C**omputer - Csökkentett Utasításkészletű Számítógép)
- Minden utasítás (gyakorlatilag) azonos idő alatt fut le
- alkalmazás: kis és közepes teljesítményű mérési és vezérlési feladatok
- pl. 8048, 8051, 80552, 8196, PIC16xx, PIC17xx, PIC18xx
- 8 .. 16 bites busz
- 1MHz ..40 MHz órajel

Mikrokontrollerek beépített periféria elemei

A perifériák a környezettel a portokon keresztül tartanak kapcsolatot (a portoknak két funkciója van).

A periféria elemek vezérlő regiszterei az adat memóriában vannak beágyazva.

1. Timer

1..4 db timer egy tokban

8..16 bites számlálás

programozható előosztó (intervallum növelés céljából)

két alapfunkció az impulzus számlálás vagy időzítés

2. Timerhez kapcsolódó funkciók

esemény időpontjának mérése (capture)

meghatározott időintervallum előállítása (compare)

periodikus időintervallumok előállítása (pl. AD konverter)

impulzusmoduláció előállítása (PWM)

3. Soros interfész

aszinkron soros IF (pl.: RS232)

szinkron IF

I2C interfész

CAN, USB...

4. AD konverter (típustól függően)

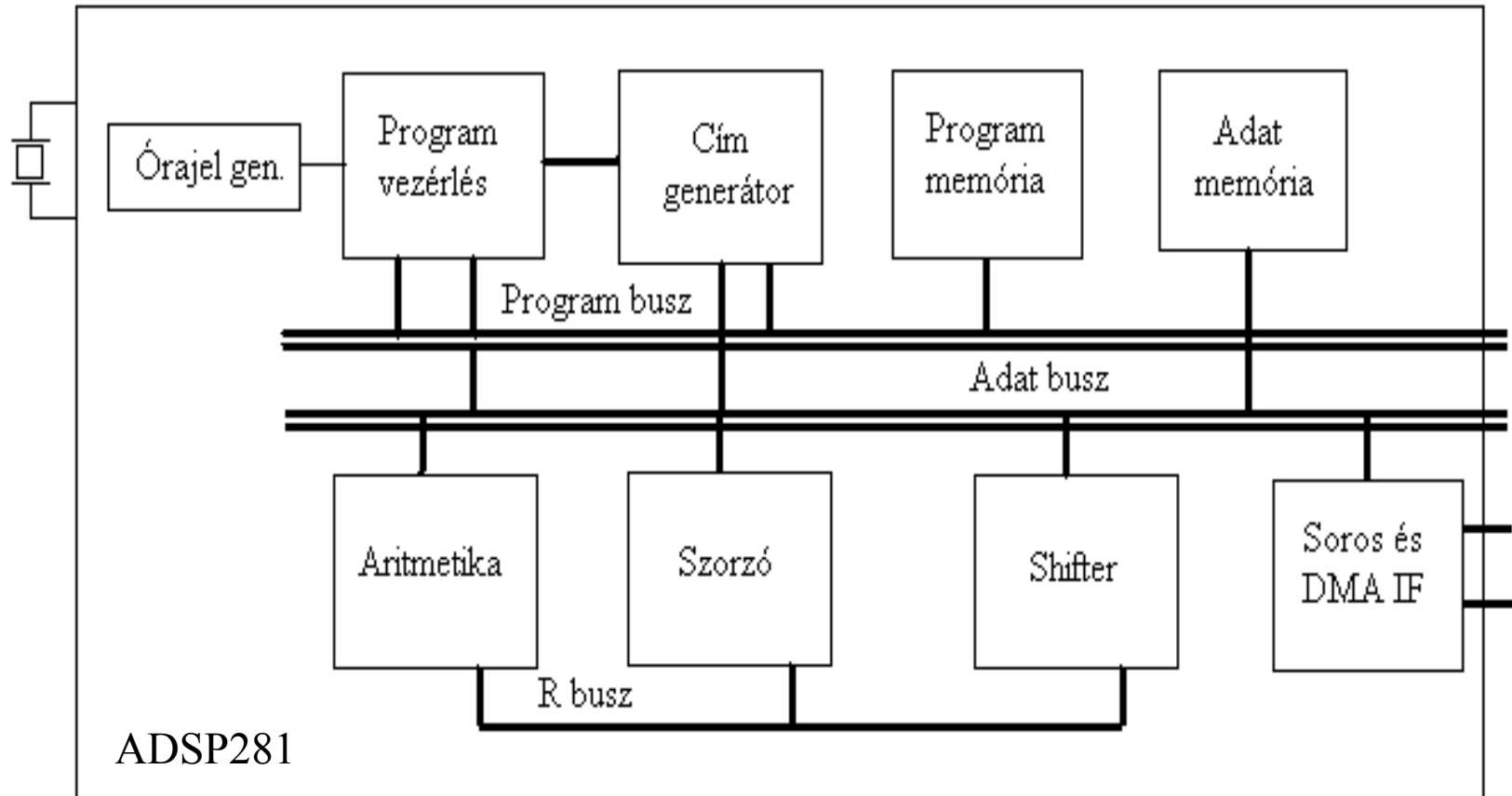
8-10-12 bit

4 ...16 bemenő csatorna (beépített MUX)

Mikrokontrollerek beépített eszközei

1. Órajel generátor és reset áramkör
2. Megszakítás vezérlő
3. Watchdog rendszer
4. EEPROM (nem felejtő) memória (típustól függően)
5. Tápfeszültség detektor (LVD)

3. Digitális szignál processzor (DSP) felépítése



3. Digitális szignál processzor (DSP)

- külön adat és program buszok amelyek nincsenek kivezelve (Harvard architektúra)
- külön adat és program memória (nem keverhető!)
- kevés beépített periféria áramkör (timer, soros IF)
- külső kapcsolat a buszrendszeren és soros porton keresztül
- egyszerű (RISC) utasításkészlet
- fejlett aritmetikai egység (40 bites szorzó, shifter)
- fejlett indirekt címzési módok (ciklikus táruk....)
- alkalmazás: közepes és nagy teljesítményű mérési és vezérlési feladatok
- nagy tömegű számítási feladatok gyors elvégzése (FFT)
- pl. ADSP2181... sok cég gyártja már
- 16 .. 64 bites busz
- 20 MHz ..400 MHz órajel

Microchip PIC sorozatú mikrokontrollerek

Típus	PIC16	PIC17	PIC18
Tokozás, lábszám	14-40	40-80	28-80
RAM [byte]	25-368	232-454	512-1536
Program memória	512-8k	2k-16k	8k-16k
Program memória	12 bit	16 bit	16 bit
Utasítások száma	35	58	77
Sebesség MIPS	2,5	8	10
HW szorzó	nem	igen	igen
AD konverter	változó	nincs	8*10 bit

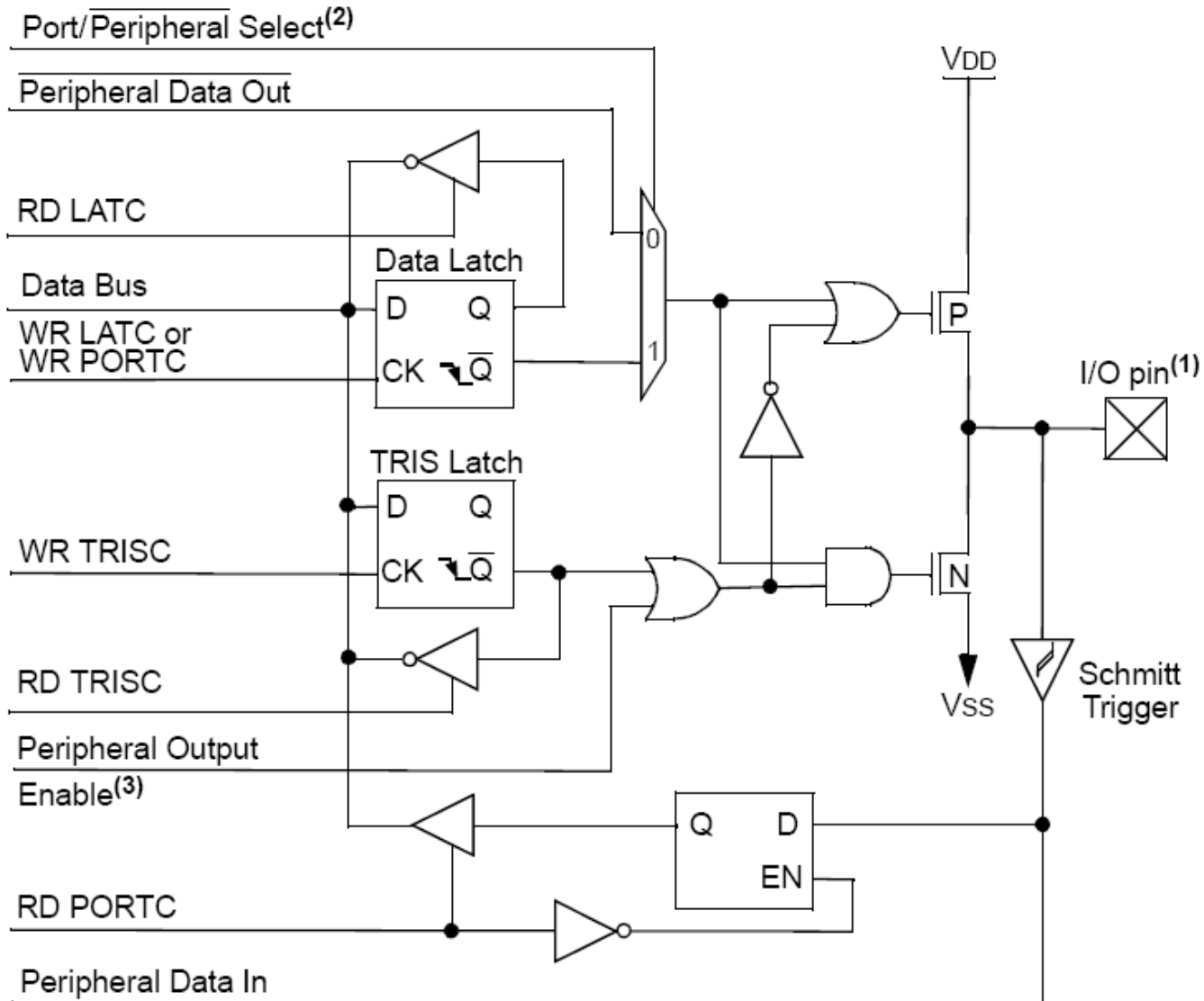
PIC18F4xx mikrokontroller portjainak másodlagos funkciói:

	0	1	2	3	4	5	6	7
port A	AN0	AN1	AN2	AN3	T0CKI	AN4/LVDIN	OSC2	-
port B	INT0	INT1	INT2	CCP2	Nincs másodlagos funkciója			
port C	T1CKI	CCP2	CCP1	SCL/SCK	SDA/SDI	SDO	TX	RX
port D	PSP0	PSP1	PSP2	PSP3	PSP4	PSP5	PSP6	PSP7
port E	AN5/RD	AN6/WR	AN7/CS	-	-	-	-	-

A port csak 7, E port csak 3 bites

AN0..7:	analóg bemenetek
INT0..2:	interrupt
RX,TX:	aszinkron soros interface
SCK, SDI, SDO:	szinkron soros interface
SCL, SDA:	I2C soros interface
T0CKI, T1CKI:	timer clock input
CCP1, CCP2:	Capture input, Compare, PWM output
PSP0..7, RD, WR,CS:	parallel slave port
LVDIN:	low voltage input

A „C” port felépítése PIC18F4xx



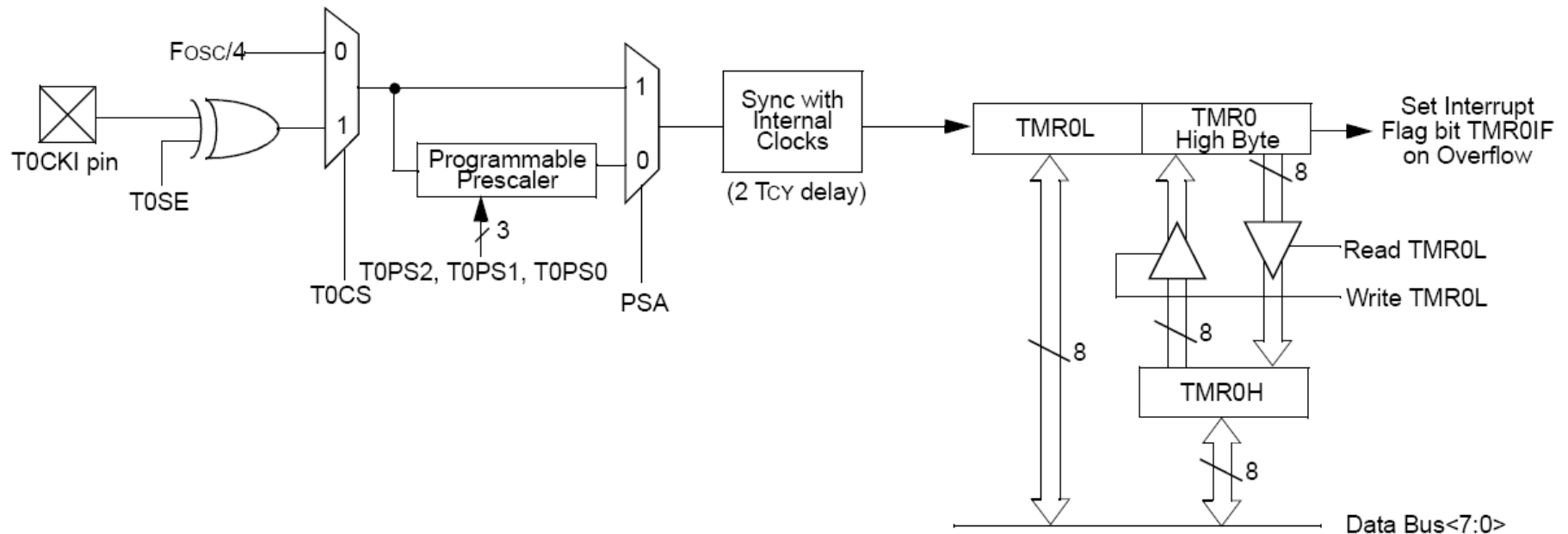
T0 16 bites timer felépítése

időzítő vagy számláló üzemmód

programozható előosztó

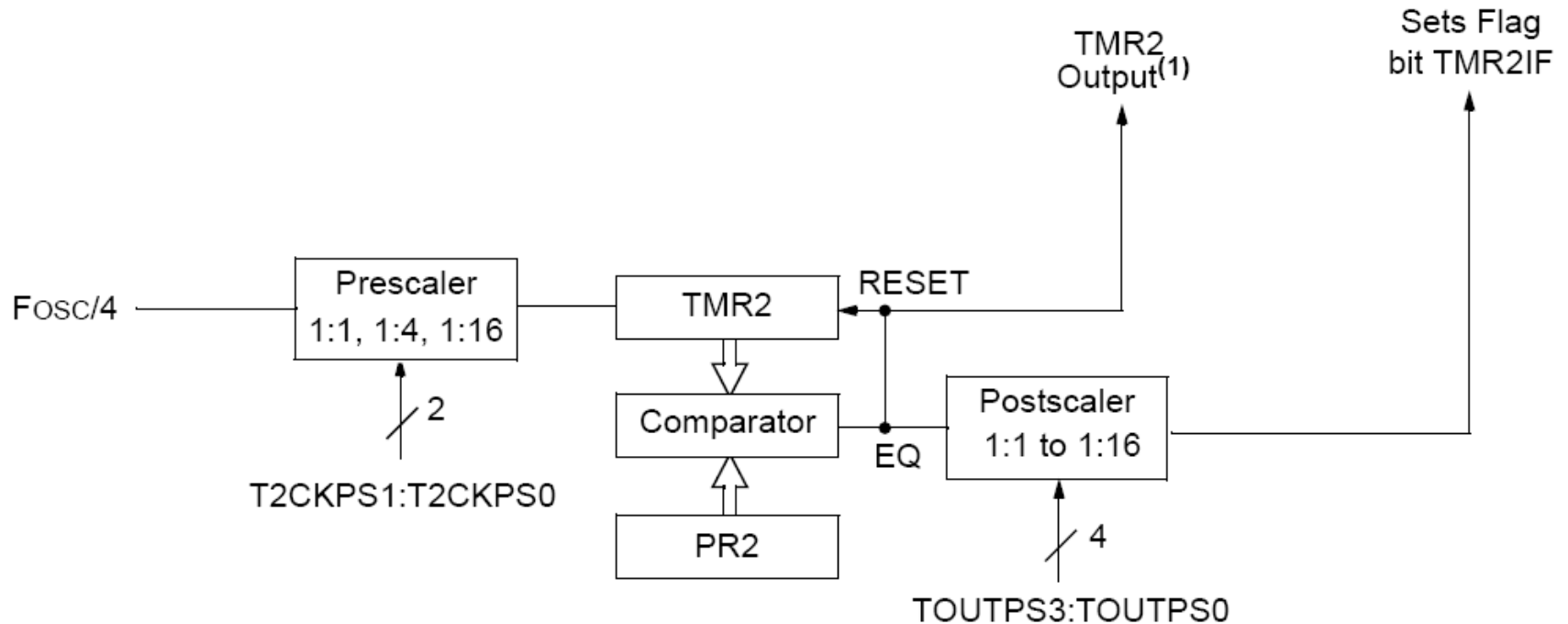
16 bites írás és olvasás

túlcsordulásakor megszakítás kérés



T2 timer felépítése

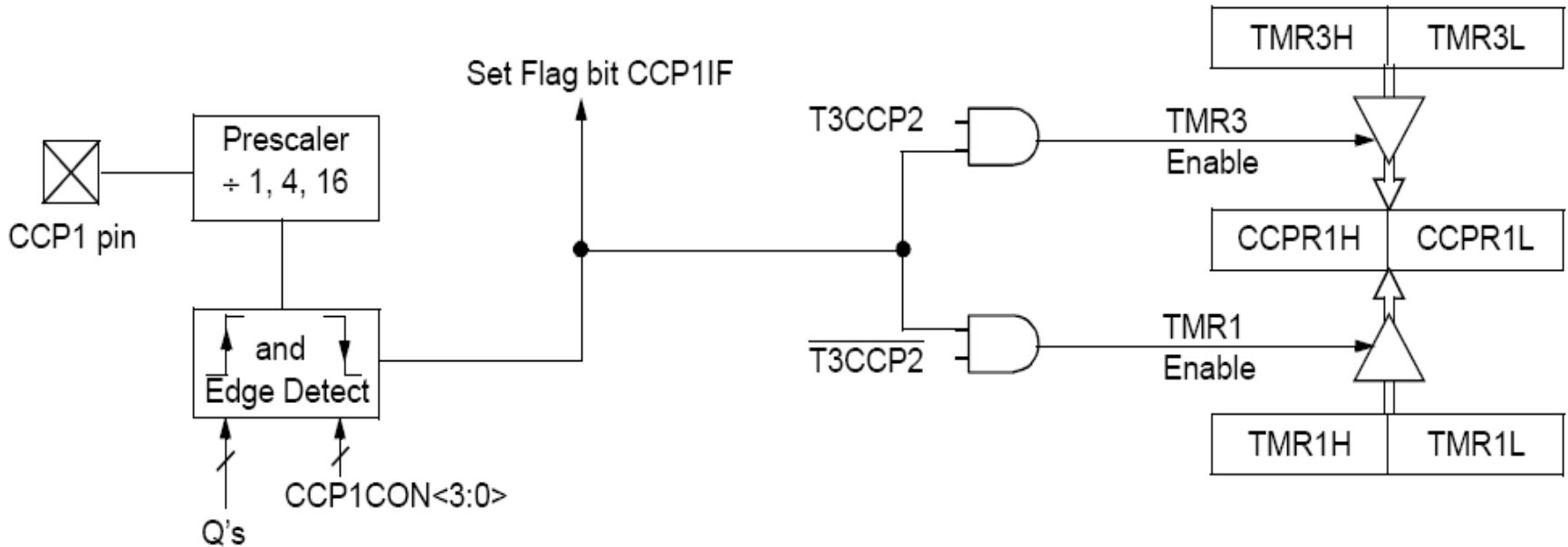
programozható előosztó és eseményszámláló
programozható periodikus ismétlődés (PR2)



PIC18 Capture funkció

Alkalmazás: külső esemény pontos idejének rögzítése

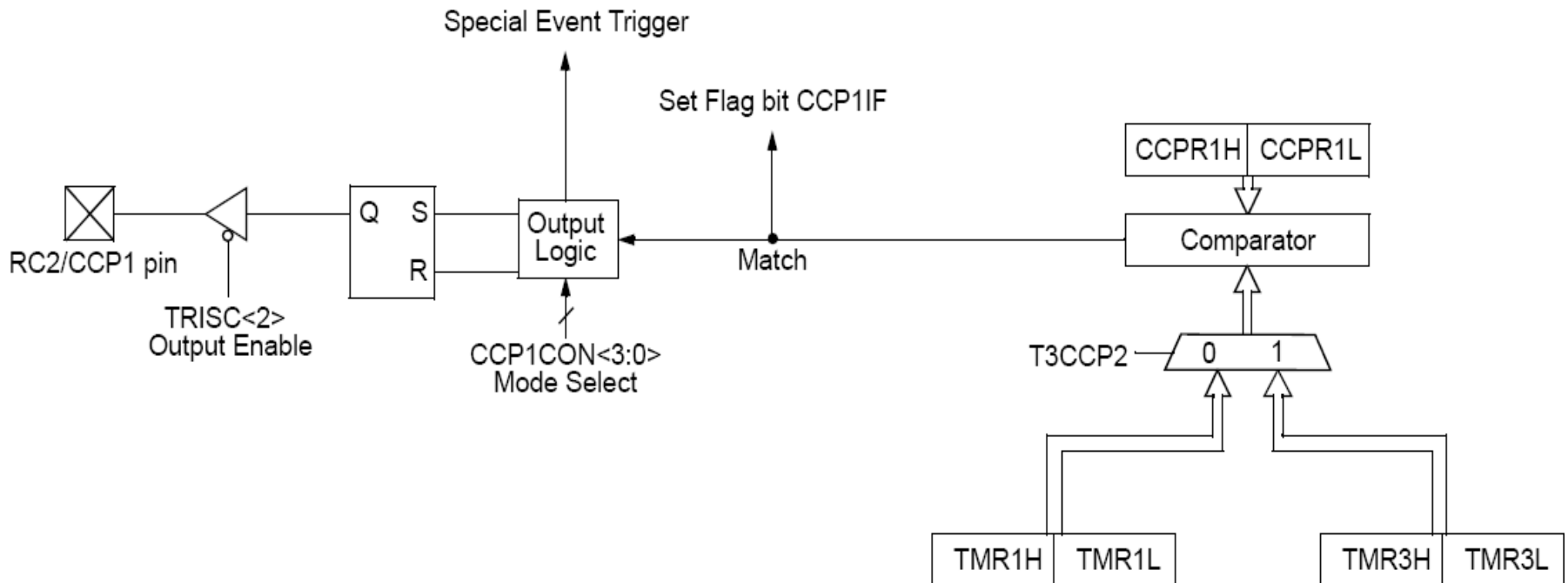
Megvalósítás T1 vagy T3 timer felhasználásával



Komparátor üzemmód

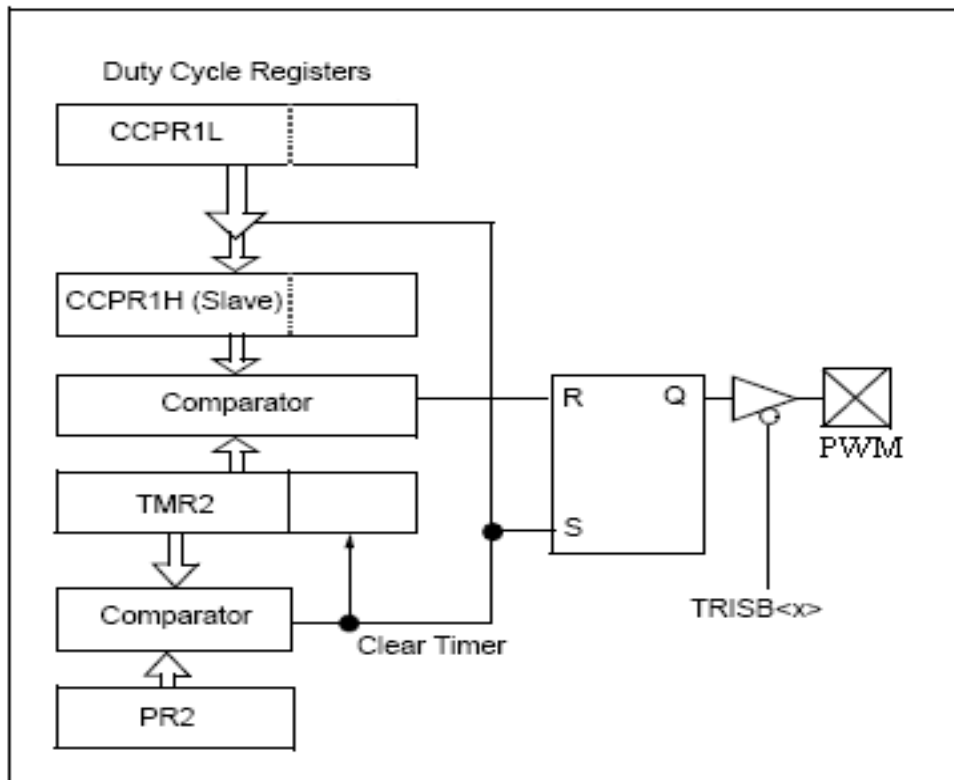
Alkalmazás: esemény előállítása előre meghatározott időpontban

Megvalósítás T1 vagy T3 timer felhasználásával

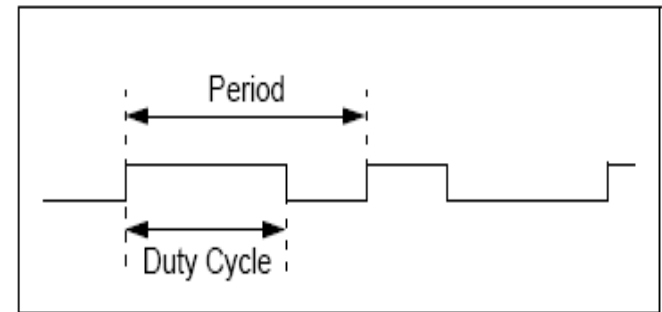


Impulzus szélesség modulált (PWM) kimenet:

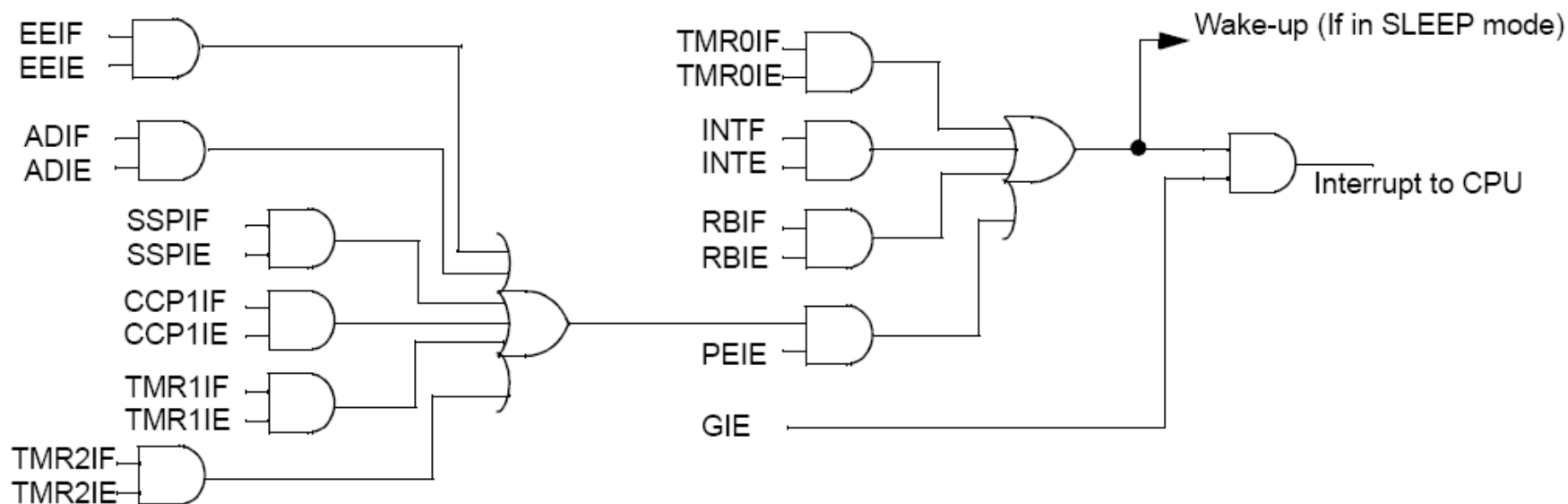
Alkalmazás: pl. aluláteresztő szűrővel (RC tag) változtatható egyenfeszültség előállítása (egyszerű DA konverter).



Kimenő jel



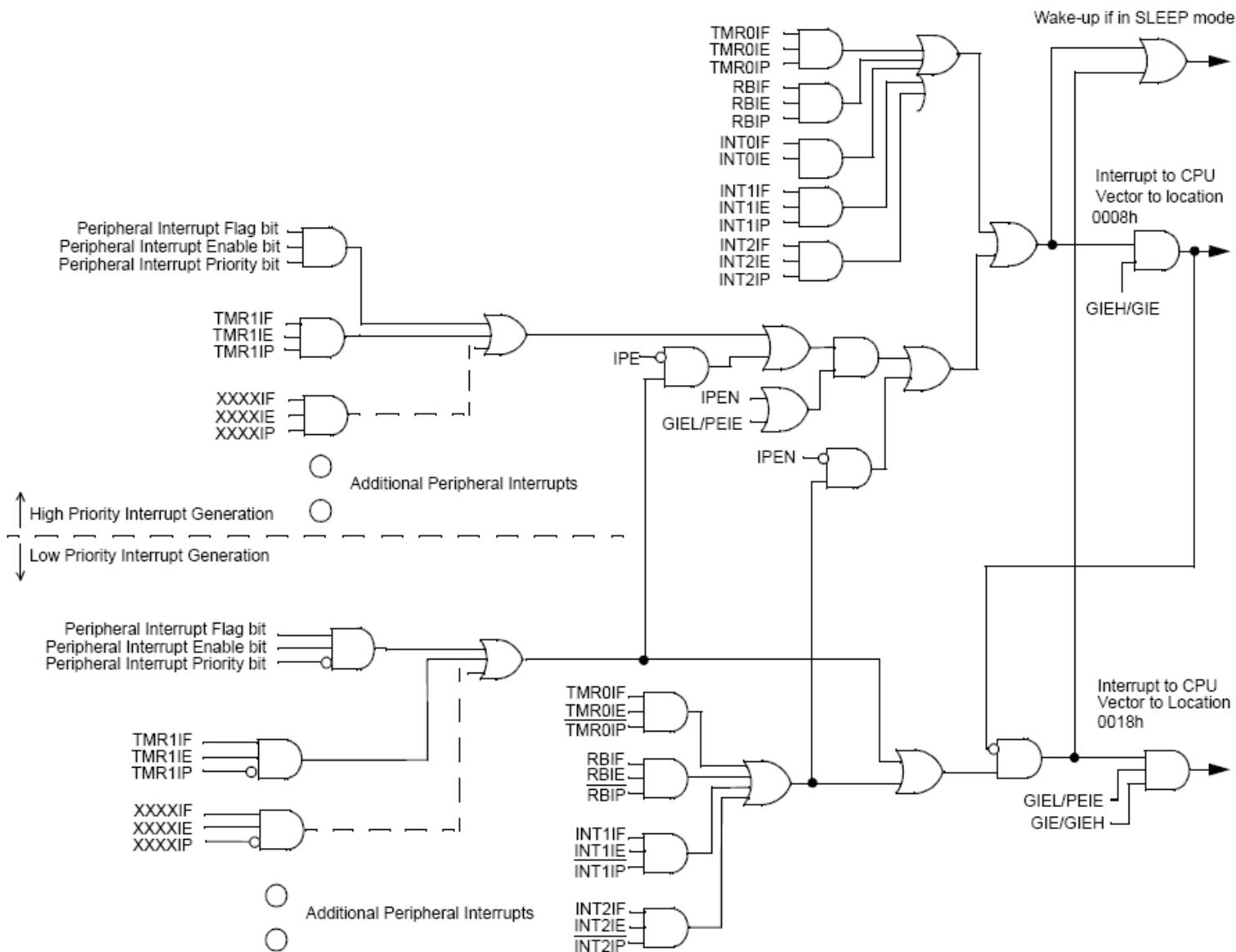
PIC16 típusú mikrokontroller megszakítási rendszere



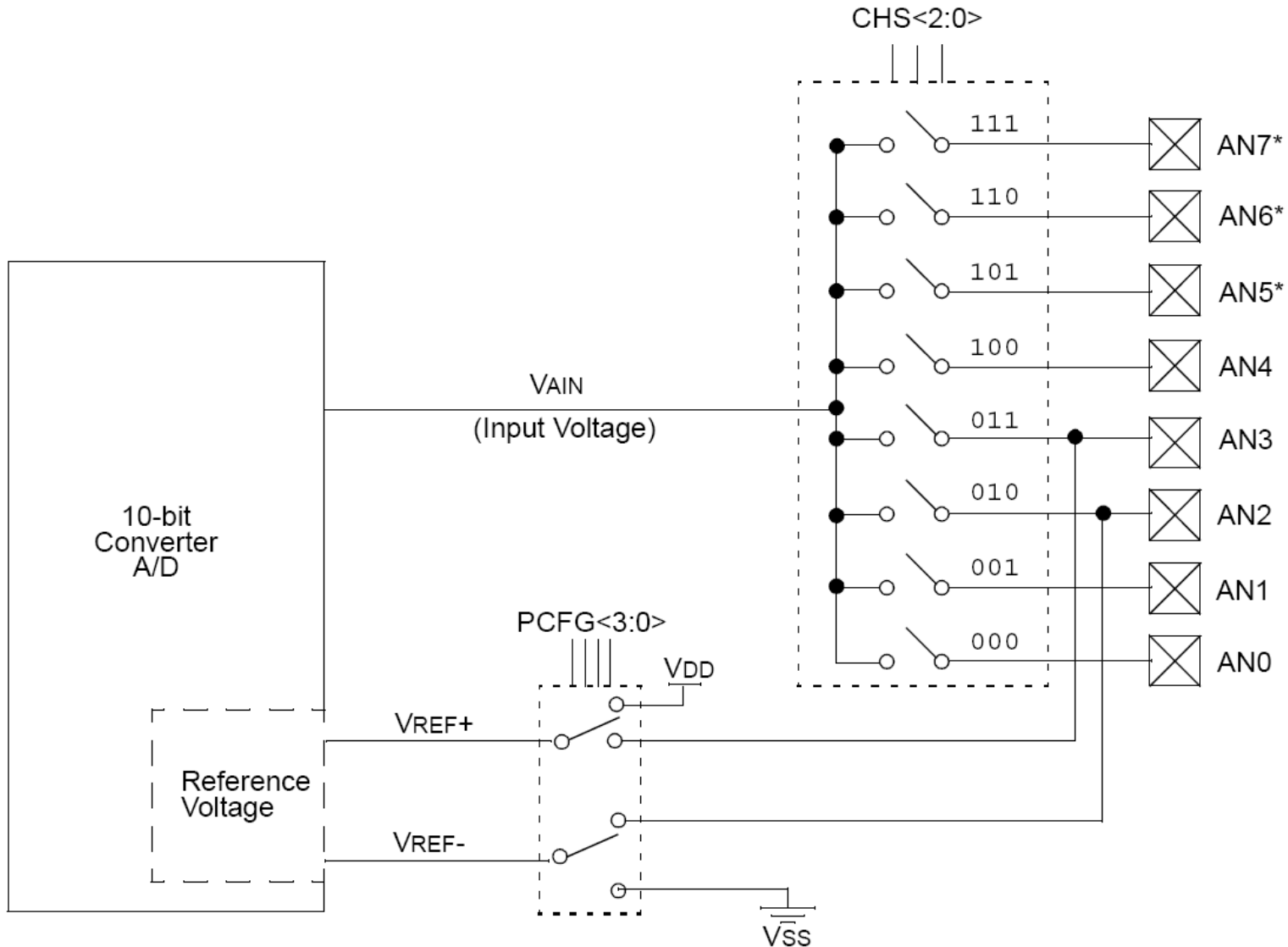
IF az interrupt flag, IE az engedélyező bemenet

A megszakítási flageket mindig törölni kell!!! (végtelen ciklus)

PIC18 típusú mikrokontroller megszakítási rendszere

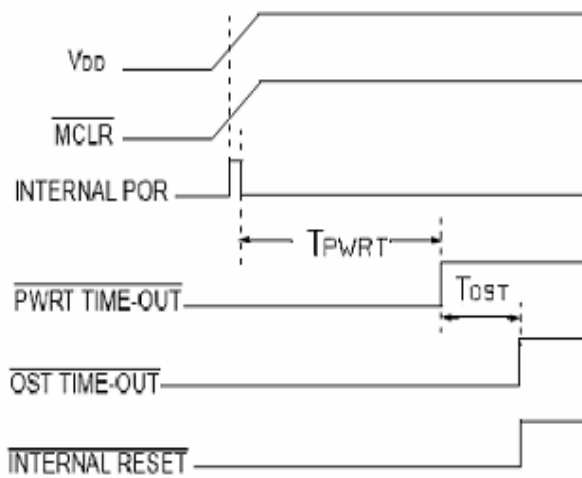


8 csatornás 10 bites AD konverter

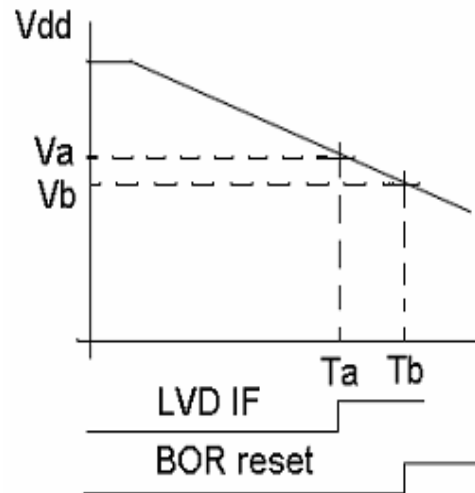


Mikrokontroller biztonságos működése

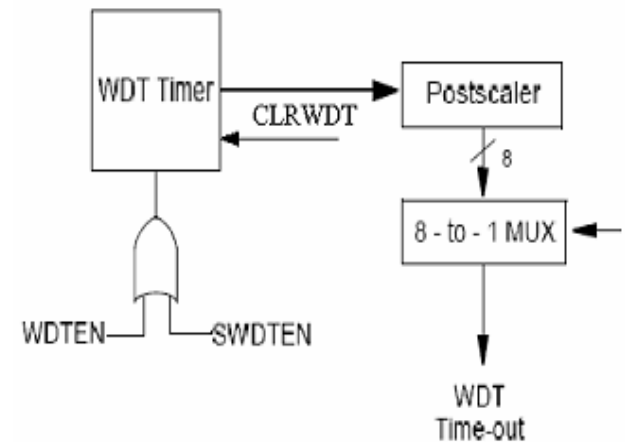
Program indulása



Tápfeszültség csökkenés



Program hiba



Tápfeszültség bekapcsoláskor:

TPWRT a feszültség stabilizálására

TOST az oszcillátor beállítására

Tápfeszültség csökkenése:

Va : az LVD áramkör megszakítást ad

Tb-Ta idő a BOR resetig

Program leállítás:

A WD timer által meghatározott idő után

WDT reset lép fel

PIC16F84

adat memória terület

00-0B vezérlő regiszterek

0C-4F általános célú regiszterek

File Address			File Address
00h	Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 ⁽¹⁾	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	68 General Purpose Registers (SRAM)	Mapped (accesses) in Bank 0	
4Fh			CFh
50h			D0h
7Fh			FFh
	Bank 0	Bank 1	

□ Unimplemented data memory location, read as '0'.

Note 1: Not a physical register.

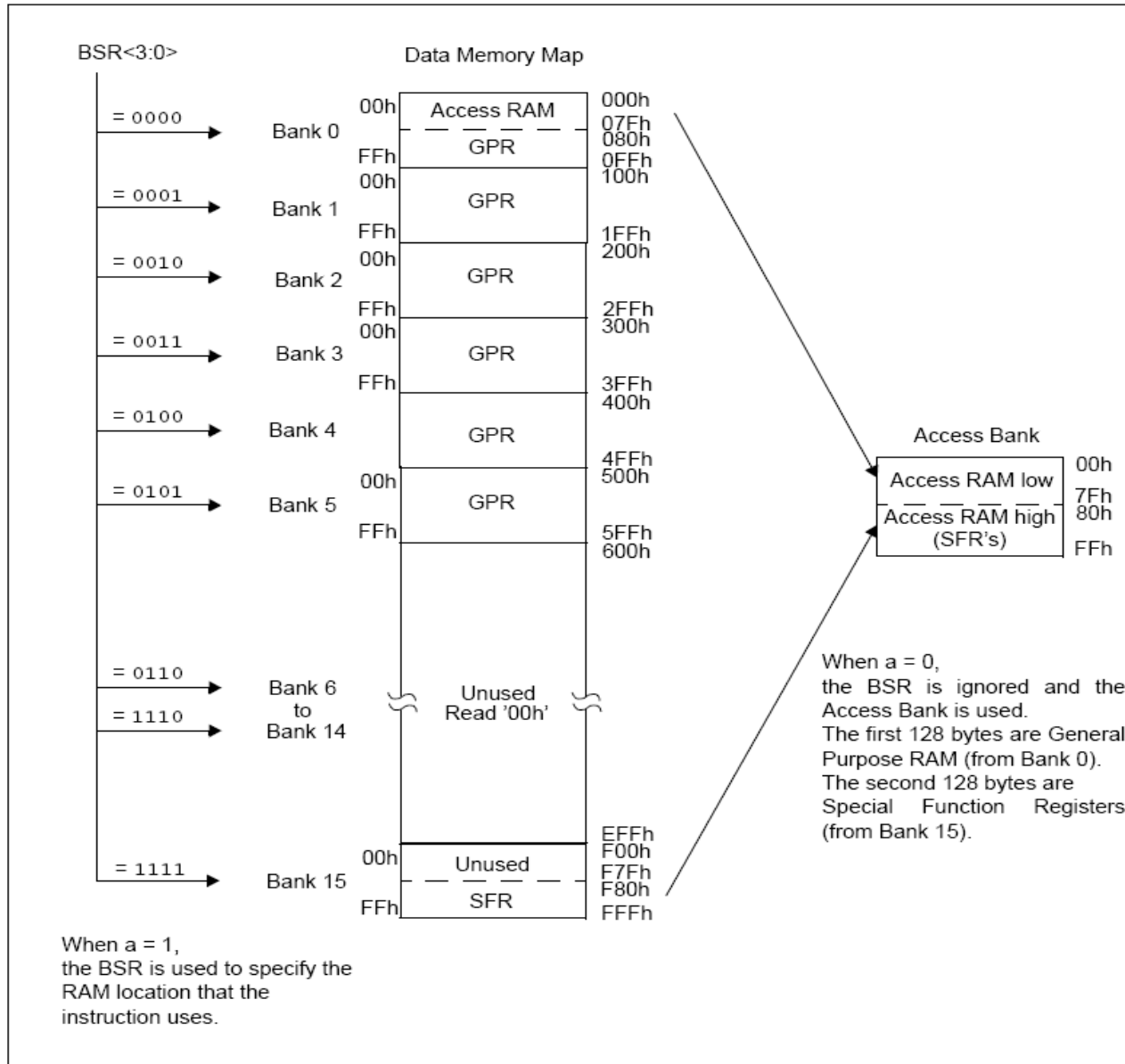
PIC17C44

adat memória terület

- 00-19 vezérlő regiszterek
- 1A-FF általános célú regiszterek

Addr	Unbanked			
00h	INDF0			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	T0STA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	Bank 1 ⁽¹⁾	Bank 2 ⁽¹⁾	Bank 3 ⁽¹⁾
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h	PRODL			
19h	<u>PRODH</u>			
1Ah				
1Fh				
20h	General Purpose RAM ⁽²⁾	General Purpose RAM ⁽²⁾		
FFh				

PIC18F452 mikrokontroller adatmemória és címzési módjai



PIC16 mikrokontroller utasításkészlete 1.

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	
			MSb		LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS								
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z
DECFSZ	f, d	Decrement f, Skip if 0	1 (2)	00	1011	dfff	ffff	
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z
INCFSZ	f, d	Increment f, Skip if 0	1 (2)	00	1111	dfff	ffff	
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z
MOVWF	f	Move W to f	1	00	0000	1fff	ffff	
NOP	-	No Operation	1	00	0000	0xx0	0000	
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z

PIC16 mikrokontroller utasításkészlete 2.

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	
			MSb			LSb		
BIT-ORIENTED FILE REGISTER OPERATIONS								
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff	
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff	
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff	
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff	
LITERAL AND CONTROL OPERATIONS								
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk	
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk	
RETFIE	-	Return from interrupt	2	00	0000	0000	1001	
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk	
RETURN	-	Return from Subroutine	2	00	0000	0000	1000	
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z

PIC18 típusú mikrokontrollerek utasításkészlete I.

Aritmetikai utasítások

Mnemonic, Operands	Description	Cycles	Status Affected
BYTE-ORIENTED FILE REGISTER OPERATIONS			
ADDWF	f, d, a Add WREG and f	1	C, DC, Z, OV, N
ADDWFC	f, d, a Add WREG and Carry bit to f	1	C, DC, Z, OV, N
ANDWF	f, d, a AND WREG with f	1	Z, N
CLRF	f, a Clear f	1	Z
COMF	f, d, a Complement f	1	Z, N
CPFSEQ	f, a Compare f with WREG skip =	1 (2 or 3)	None
CPFSGT	f, a Compare f with WREG skip >	1 (2 or 3)	None
CPFSLT	f, a Compare f with WREG skip <	1 (2 or 3)	None
DECf	f, d, a Decrement f	1	C, DC, Z, OV, N
DECFSZ	f, d, a Decrement f. Skip if 0	1 (2 or 3)	None
DCFSNZ	f, d, a Decrement f. Skip if Not 0	1 (2 or 3)	None
INCF	f, d, a Increment f	1	C, DC, Z, OV, N
INCFSZ	f, d, a Increment f. Skip if 0	1 (2 or 3)	None
INFSNZ	f, d, a Increment f. Skip if Not 0	1 (2 or 3)	None
IORWF	f, d, a Inclusive OR WREG with f	1	Z, N
MOVf	f, d, a Move f	1	Z, N
MOVFF	f _s , f _d Move f _s (source) to 1st word f _d (destination)2nd word	2	None
MOVWF	f, a Move WREG to f	1	None
MULWF	f, a Multiply WREG with f	1	None
NEGF	f, a Negate f	1	C, DC, Z, OV, N
RLCF	f, d, a Rotate Left f through Carry	1	C, Z, N
RLNCF	f, d, a Rotate Left f (No Carry)	1	Z, N
RRCF	f, d, a Rotate Right f through Carry	1	C, Z, N
RRNCF	f, d, a Rotate Right f (No Carry)	1	Z, N
SETf	f, a Set f	1	None
SUBFWB	f, d, a Subtract f from WREG with borrow	1	C, DC, Z, OV, N
SUBWF	f, d, a Subtract WREG from f	1	C, DC, Z, OV, N
SUBWFB	f, d, a Subtract WREG from f with borrow	1	C, DC, Z, OV, N
SWAPF	f, d, a Swap nibbles in f	1	None
TSTFSZ	f, a Test f, skip if 0	1 (2 or 3)	None
XORWF	f, d, a Exclusive OR WREG with f	1	Z, N
BIT-ORIENTED FILE REGISTER OPERATIONS			
BCF	f, b, a Bit Clear f	1	None
BSF	f, b, a Bit Set f	1	None
BTFSC	f, b, a Bit Test f. Skip if Clear	1 (2 or 3)	None
BTFSS	f, b, a Bit Test f. Skip if Set	1 (2 or 3)	None
BTG	f, d, a Bit Toggle f	1	None

PIC18 típusú mikrokontrollerek utasításkészlete II.

Program szervező utasítások

Mnemonic, Operands	Description	Cycles	Status Affected	
CONTROL OPERATIONS				
BC	n	Branch if Carry	1 (2)	None
BN	n	Branch if Negative	1 (2)	None
BNC	n	Branch if Not Carry	1 (2)	None
BNN	n	Branch if Not Negative	1 (2)	None
BNOV	n	Branch if Not Overflow	1 (2)	None
BNZ	n	Branch if Not Zero	2	None
BOV	n	Branch if Overflow	1 (2)	None
BRA	n	Branch Unconditionally	1 (2)	None
BZ	n	Branch if Zero	1 (2)	None
CALL	n, s	Call subroutine 1st word 2nd word	2	None
CLRWDT	—	Clear Watchdog Timer	1	\overline{TO} , \overline{PD}
DAW	—	Decimal Adjust WREG	1	C
GOTO	n	Go to address 1st word 2nd word	2	None
NOP	—	No Operation	1	None
NOP	—	No Operation (Note 4)	1	None
POP	—	Pop top of return stack (TOS)	1	None
PUSH	—	Push top of return stack (TOS)	1	None
RCALL	n	Relative Call	2	None
RESET		Software device RESET	1	All
RETFIE	s	Return from interrupt enable	2	GIE/GIEH, PEIE/GIEL
RETLW	k	Return with literal in WREG	2	None
RETURN	s	Return from Subroutine	2	None
SLEEP	—	Go into standby mode	1	\overline{TO} , \overline{PD}

PIC18 típusú mikrokontrollerek utasításkészlete III.

Konstansokkal
végezhető utasítások

Közvetlen utasítások
Program memória
hozzáféréshez

Mnemonic, Operands	Description	Cycles	Status Affected
LITERAL OPERATIONS			
ADDLW k	Add literal and WREG	1	C, DC, Z, OV, N
ANDLW k	AND literal with WREG	1	Z, N
IORLW k	Inclusive OR literal with WREG	1	Z, N
LFSR f, k	Move literal (12-bit) 2nd word to FSRx 1st word	2	None
MOVLB k	Move literal to BSR<3:0>	1	None
MOVLW k	Move literal to WREG	1	None
MULLW k	Multiply literal with WREG	1	None
RETLW k	Return with literal in WREG	2	None
SUBLW k	Subtract WREG from literal	1	C, DC, Z, OV, N
XORLW k	Exclusive OR literal with WREG	1	Z, N
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS			
TBLRD*	Table Read	2	None
TBLRD*+	Table Read with post-increment		None
TBLRD*-	Table Read with post-decrement		None
TBLRD+*	Table Read with pre-increment		None
TBLWT*	Table Write	2 (5)	None
TBLWT*+	Table Write with post-increment		None
TBLWT*-	Table Write with post-decrement		None
TBLWT+*	Table Write with pre-increment		None

Mikrokontroller assembler programozása

Irodalom:

www.microchip.com

MPLAB Integrated Development Environment

www.chipcad.hu

Mikrokontroller assembler programozása

- alacsony szintű programnyelv
- közvetlen gépi utasításokat használ
- processzorfüggő program
- nem hordozható
- korlátozott aritmetikai lehetőségek
- programfutas pontos időzítése lehetséges
- real-time rendszerekben ajánlott
- PIC sorozatú processzorok assemblere: MPASM.EXE
- MPLAB fejlesztői környezet és szoftver szimulátor
- ICE 2000 in circuit emulator

Assembler program felépítése

- A program sorokból áll
- Egy utasítás nem folytatható a következő sorban
- Egy sorban csak egy utasítás állhat
- Egy utasítás felépítése:

címke: utasítás argumentum,argumentum ;megjegyzés

- A négy elem közül bármennyi hiányozhat
- címke mindig a sor elején, általában max. 6 karakter hosszú
- az utasítás lehet processzor utasítás (az utasításkészletből)
- az utasítás lehet az assemblert vezérlő parancs
- a ';' után a sor végéig az assembler semmit nem vesz figyelembe

Assembler vezérlő parancsok

- nincs egységes szintaktika

- számok deklarációja:

használható számok: pozitív egész számok 16 vagy 32 bit
hosszban (MPASM 32 bites)

bináris, oktális, decimális vagy hexadecimális alakban
beállítható (radix), MPASM default hexadecimális

írásmódok:	hexa:	0A13 (MPASM) 0A13h vagy 0xA13
	decimális:	D'0123' (MPASM) 0123d
	bináris:	B'01001110' (MPASM) 01001110 b

stringek: '' string ''

Aritmetikai kifejezések

elemek: számok, konstansok, változók, címkék
az elemeket műveletek kötik össze
műveletek és kiértékelés assemblerfüggő
csak egész számokkal dolgozik
aritmetikai túlcsondulást nem jelez
MPASM C stílusú aritmetikát használ

aritmetikai műveletek: (nem teljes felsorolás)

szorzás: *, /, %, *=, /=, %=

összeadás: +, -, +=, -=, ++, --

logikai: &, &&, |, ||, ^

összehasonlítás: >, >=, <, <=, ==

eltolás: >>, >>=, <<, <<=

byte leválasztás: high, low

Deklarácók

konstansok deklarácója:

konstansnev equ (vagy =) aritmetika kifejezes

nincs ':' mert nem része a processzor programnak
értéke a program futása során nem változhat!

szerepe hasonló a pascal **const** utasításhoz

a program belsejében ne használjunk számokat !!

változók deklarácója:

valtozonev set kifejezes

nincs ':' mert nem része a processzor programnak
értéke a program futása során változhat

szerepe hasonló a pascal **var** utasításhoz

Speciális változó **\$** értéke az aktuális programcím

Adatok elhelyezése a programban

[címke:]	db	kif, kif,...	; byte-ok beírása
[címke:]	dw	kif, kif,...	;word-ök beírása
[címke:]	data	'string',..	;szöveg karakterei

A program memóriában lévő adatokhoz címke rendelhető, amely alapján az adott helyre hivatkozni lehet. A címke értéke konstans (akár szerepelhet aritmetikai kifejezésben), értéke a mögötte álló utasítás kezdő címe. Adott [címke:] mindig csak egyszer használható, értéket a program fordítása során a 2. menetben kapnak.

pl:

tabla:	db	0,1,2,3,4,5	;számok 0..5
hexkar:	data	'0123456789ABCDEF'	;karakterek

Program elhelyezése memóriában

A fordító a 0-s címnél kezdi, és sorban rakja le az utasításokat. Ennek megváltoztatására szolgáló utasítás az **org**, melynek hatására meghatározza a következő utasítás, vagy adatszámát.

org cím ;a programrész **cím**-nél indul

pl:

```
org 20 ;megszakítási vektorcím  
goto megszakitasrutin
```

Pl. megszakítási vektorcím, amit kötelező ide tenni. Gond akkor van, ha már túljutottam rajta, és vissza akarom tenni. Ezt nem lehet, foglalt helyet nem szabad felülírni, ezért mindig a program elején célszerű használni. **org** után lehet aritmetikai kifejezés is, ha kiértékeléskor értelmezhető.

Mi történik akkor, ha azt mondom, hogy `org $+2` ?

Program szervezés

Szöveghelyettesítő címke definiálása

```
#define   nev   szoveg   ; helyettesítés
```

A fordító a hivatkozásban a „nev”-et a „szoveg”-el helyettesíti, majd utána fordít.

Pl.:

```
#define   Z     ALUSTA,2 ; Z flag
```

Zérus flag, amely az ALUSTA regiszter 2. bitje, e helyett használom a Z karaktert.

```
include   <filenev>   ;szoveg beiktatás  
end       ;programvége
```

Feltételes fordítás, ciklusszervezés

Feltételes fordítás:

A forrásprogram bizonyos részei a fordítás során fennálló feltételtől függően lesznek benne a tárgyprogramban. (pl.: tesztelés)

```
if           kifejezes      ; ha kifejezes≠0
utasítások

.....
else                ;egyébként, nem kötelező
utasítások

.....
endif                ;lezárás
```

Ciklusszervezés:

Az utasításokat beszerkeszti a programba fordítás során.

```
while        kifejezes      ; amíg a kifejezes>0
utasítások

.....
endw                ;ciklus vége, külön sorban!
```

Makró

A makró olyan programrészlet, amelyet többször akarunk alkalmazni, de eltérő argumentumokkal. Előbb kell definiálni, aztán meghívni!
Tetszőleges mélységig egymásba ágyazhatók.

```
nev macro parameter1, parameter2,...  
local címke,címke.. ;lokális címkék  
utasítások  
exitm ;kilépés, ha kell  
.....  
endm ;macro vége
```

makró használat (hívás):

```
nev adatok1,adatok2,..
```

Célszerű makró készletet külön fájlban tárolni, és **include** utasítással beilleszteni a programba.

Makró vagy szubrutin (futási idő, hely, paraméter átadás)?

PIC 18 program példák

```
include    p18f452.inc
w         equ    0           ;aritmetikai eredmény w-be
f         equ    1           ;aritmetikai eredmény_file-ba
a         set    0           ;címezési mód

;2 byte-os összeadás: (high, low sorrend)
add2      macro    f1,f2     ;f1=f1+f2
          movf    f2+1,w,a   ;alsó byte w-be
          addwf   f1+1,f,a   ;f1[low]=f1[low]+f2[low]
          movf    f2+0,w,a   ;felső byte w-be
          addwfc  f1+0,f,a   ;összeadás Carry-vel
          endm

;használat:
adat1     equ    10         ;adatok RAM területen
adat2     equ    20
          add2    adat1,adat2 ;adat1=adat1+adat2
```


Összeadás megadott hosszban:

```
add    macro    f1,f2,nb          ;f1[nb]=f1[nb]+f2[nb]
        movf    f2+nb-1,w,a      ;alsó byte w-be
        addwf   f1+nb-1,f,a      ;f1[low]=f1[low]+f2[low]
nn     set     nb-1              ;nb-1 ciklus számlálóba
        while   nn > 0          ;feltétel vizsgálat
        movf    f2+nn-1,w,a      ;felső(bb) byte-ok w-be
        addwfc  f1+nn-1,f,a      ;összeadás Carry-vel
nn     set     nn-1              ;ciklus számláló csökkentése
        endw    ;ciklus vége
        endm    ;makró vége
```

használat:

```
add    cim1,cim2,4              ;cim1=cim1+cim2
                                           ;4 byte hosszan
```

Inkrementálás adott hosszban:

A legalacsonyabb helyértéket 1-el növeljük, és kezeljük az átvitelt.

```
inc    macro    f1,n        ;f1[n]=f1[n]+1
        incf    f1+n-1,f    ;legalsó helyérték 1-el növelése
        movlw   0           ;w reg törlése összeadáshoz
bn     set      n-1        ;ciklusváltozó beállítása
        while   bn > 0     ;feltételvizsgálat
        addwfc  f1+bn-1,f   ;összeadás Carry-n keresztül
bn     set      bn-1
        endw
        endm
```

Másik megoldás? Feltételvizsgálat van e átvitel (C) és leállítás.

Buta és hosszabb, de gyorsabb, mint a feltételvizsgálat és leállítás.

2-es komplementum adott hosszban


2-es komplementum = 1-es komplementum + 1

A korábban elkészített inc makró felhasználásával

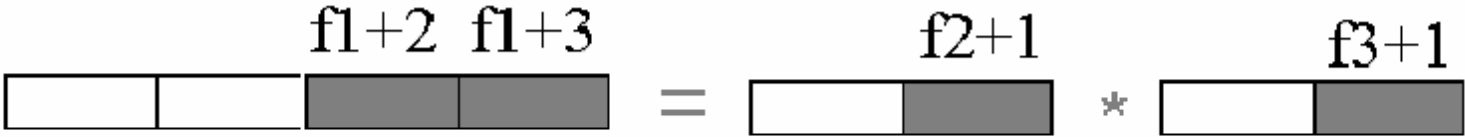
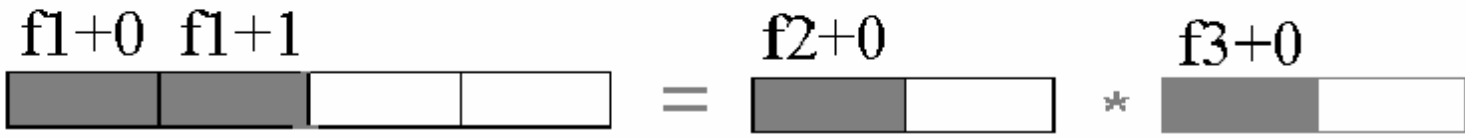
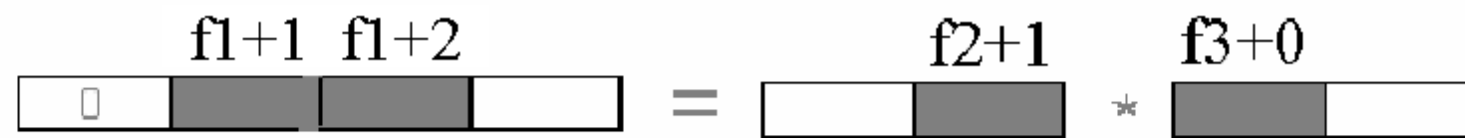
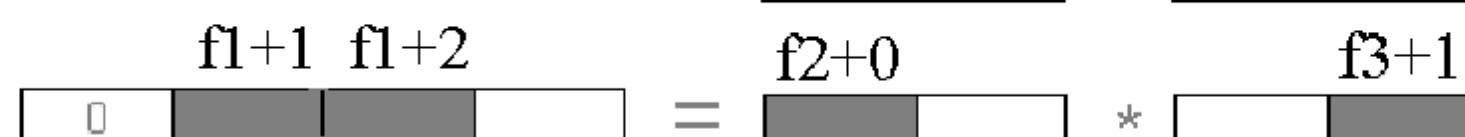
```
neg    macro    f1,n          ;f1[n]=-f1[n]
bn     set      n
      while    bn > 0
      comf     f1+bn-1,f      ;1-es komplementum képzés
bn     set      bn-1
      endw
      inc      f1,n          ;beágyazott makró hívás
      endm
```

A beágyazott makrónak természetesen híváskor már definiálnak kell lennie!

2 byte-os előjel nélküli szorzás

$$[f_{1+0} \ f_{1+1} \ f_{1+2} \ f_{1+3}] = [f_{20+} \ f_{2+1}] * [f_{3+0} \ f_{3+1}]$$


Felbontás 1 byte-os rész szorzásokra:

- 1, $f_{1+2} \ f_{1+3}$

- 2, $f_{1+0} \ f_{1+1}$

- 3, $f_{1+1} \ f_{1+2}$

- 4, $f_{1+1} \ f_{1+2}$


A szorzat a négy érték összege lesz. Természetesen az első két tagot nem adjuk össze, csak a byte-kat helyre tesszük.

Segéd makrók

;1 byte-os előjel nélküli szorzás

```
mul1uu    macro    f1,f2,f3                ;f1[2]=f2[1]*f3[1]
           movf    f2,w,a
           mulwf   f3                      ;prod= f2[1]*f3[1]
           movpf   prodh,f1+0             ;f1+0=prodh
           movpf   prodl,f1+1            ;f1+1=prodl
           endm
```

;részösszeg képzése

```
add3      macro    f1                      ; f1[3]=f1[2]+prod[2])
           movpf   prodl,w                ;wreg=prodl
           addwf   f1+2,f                 ;f1+2=f1+2+wreg
           movpf   prodh,w                ;wreg=prodh
           addwfc  f1+1,f                 ;f1+1=f1+1+wreg
           movlw   0                      ;túlcsordulás kezelése
           addwfc  f1+0,f
           endm
```

2 byte-os előjel nélküli szorzás makróval

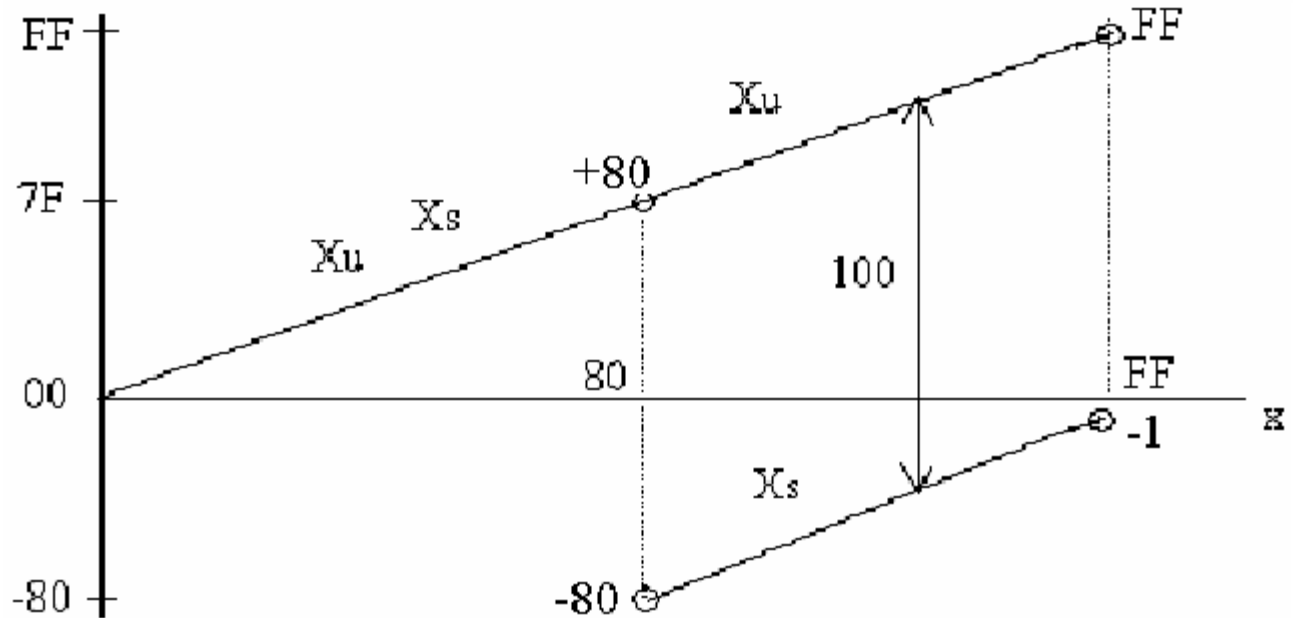
```
mul2uu    macro    f1,f2,f3                ;f1[4]=f2[2]*f3[2]
           mul1uu  f1+2,f2+1,f3+1         ;alsó 2 byte
           mul1uu  f1+0,f2+0,f3+0         ;felső 2 byte
           movf    f2+1,w,a                ;keresztorzozat
           mulwf   f3+0                    ;prodh,l=f2+1*f3+0
           add3    f1                      ;keresztorzozat hozzáadása
           movf    f2+0,w,a                ;másik keresztorzozat
           mulwf   f3+1                    ;prodh,l=f2+0*f3+1
           add3    f1                      ;keresztorzozat hozzáadása
           endm
```

használat:

```
mul2uu    30,20,22
mul2uu    34,24,26
```

Előjeles szorzás (8 biten)

Analóg jel	unipoláris	2-es kompl.
+FS	1111.1111	-
+FS-LSB	1111.1110	0111.1111
1 LSB	0000.0001	0000.0001
0	-	0000.0000
-1 LSB	-	1111.1111
-FS+LSB	-	1000.0001
-FS	-	1000.0000



Kettes komplementum szám értelmezése, ha $x=a$ byte értéke:

$$X_s = X_u \quad \text{ha } x < 80\text{h azaz pozitív, és}$$

$$X_s = X_u - 100\text{h} \quad \text{ha } x \geq 80\text{h azaz negatív}$$

1. Legyen $X_s < 0$ és $Y_s \geq 0$, ekkor

$$M = X_s * Y_s = (X_u - 100h) * Y_u = X_u * Y_u - 100h * Y_u$$

2. Legyen $X_s \geq 0$ és $Y_s < 0$, ekkor

$$M = X_s * Y_s = X_u * (Y_u - 100h) = X_u * Y_u - 100h * X_u$$

3. Legyen $X_s < 0$ és $Y_s < 0$, ekkor

$$\begin{aligned} M &= X_s * Y_s = (X_u - 100h) * (Y_u - 100h) = \\ &= X_u * Y_u - 100h * Y_u - 100h * X_u + 10000h \end{aligned}$$

Az előjeles szorzás ($X_s * Y_s$) algoritmus:

($M_u = X_u * Y_u$ az előjel nélküli szorzás)

ha $X < 0$ akkor $M_s = M_u - 100h * Y_s$

ha $Y < 0$ akkor $M_s = M_u - 100h * X_s$

ha $X < 0$ és $Y < 0$ akkor $M_s = M_u - 100h * Y_s - 100h * X_s$

A $10000h$ hozzáadása elhagyható, mert kilóg a számtartományból!!

Előjeles szorzó makró

```
mul2ss    macro    f1,f2,f3    ;f1[4]=f2[2]*f3[2]
          local    lab1,lab2
          mul2uu    f1,f2,f3    ;előjel nélküli szorzás
          btfss     f2,7        ;előjel vizsgálat
          goto      lab1        ;f2 pozitív, ugrás tovább
          sub       f1,f3,2     ;f2 negatív ezért kivonás
lab1:     btfss     f1,7        ;előjel vizsgálat
          goto      lab2        ;f1 pozitív, ugrás tovább
          sub       f1,f2,2     ;f1 negatív ezért kivonás
lab2:
          endm
```

100h-val való szorzás az 2 bájtal való jobbra tolást jelent
sub kivonó makró $f1=f1-f3$, 2 bájt hosszan

Négyzetösszegző makró

;input = új adat (16 bites)

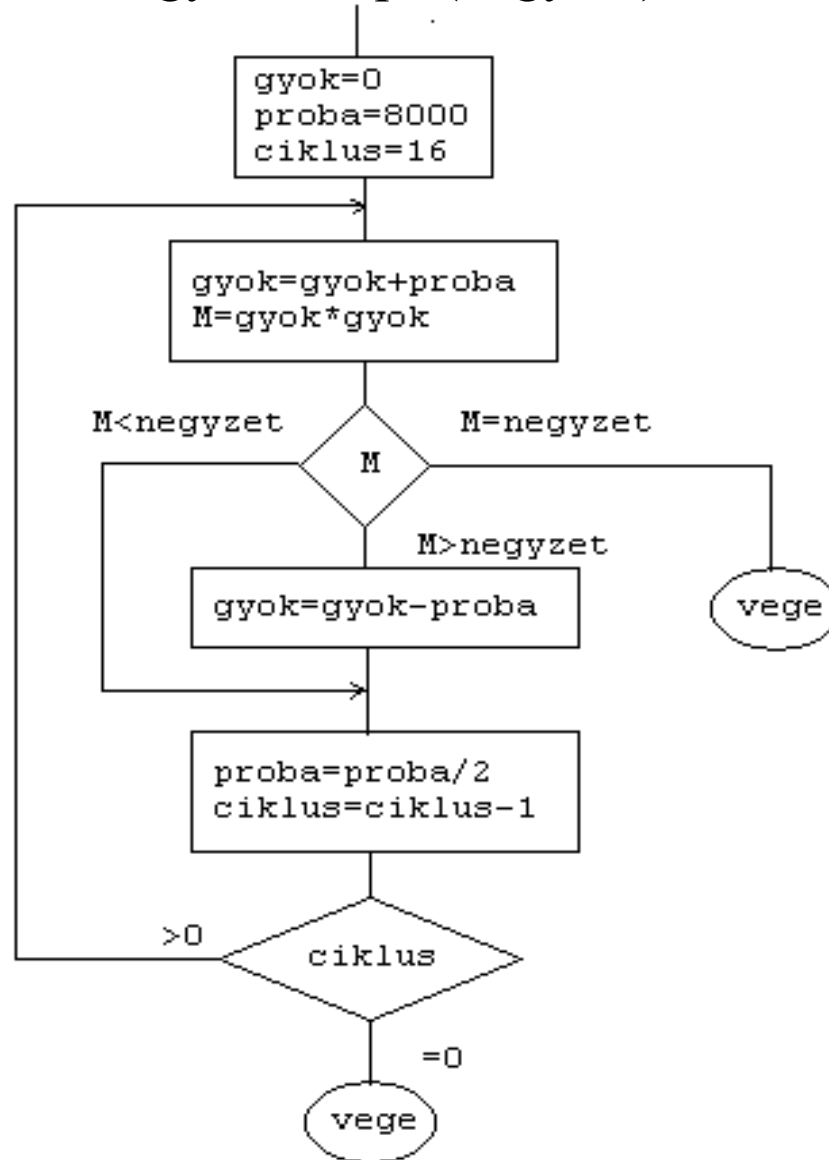
;mulbuf = szorzat tároláshoz (32 biten)

;sumbuf = összegző tár (48 biten)

```
addinp macro    sumbuf,mulbuf,input
               mul2ss mulbuf,input,input    ;szorzással négyzet
               add    sumbuf+2,mulbuf,4     ;teljes összeadó makró
               movlw  0
               addwfc sumbuf+1,f           ;átvitel kezelése
               addwfc sumbuf+0,f
               endm
```

32 bites szám négyzetgyöke

gyok = sqrt (negyzet)

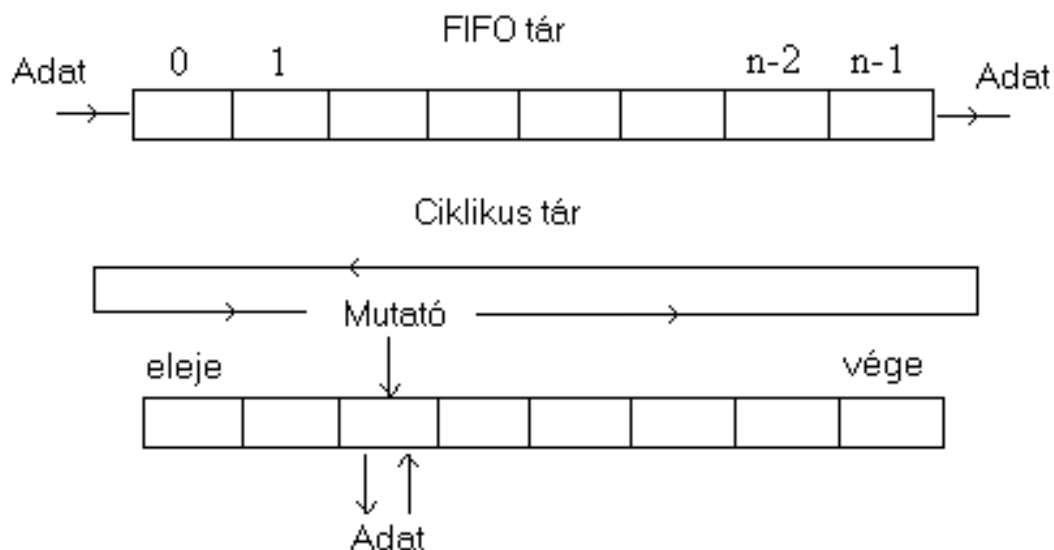


Gyökvonás makró

```
sqrt  macro      gyok,negyzet,M,proba,ciklus ;gyok=sqrt[negyzet]
      local      ujra,kovet,vege
      clr        gyok,2                      ;változók inicializálása
      movn       proba,8000,2                ;proba feltöltése
      movn       ciklus,10,1                 ;ciklusváltozó feltöltése
ujra:  add       gyok,proba,2                 ;gyok=gyok+proba
      mul2uu     M,gyok,gyok                 ;próbálkozás
      cmp       negyzet,M,4                  ;komparálás
      btfsc     Z
      goto      vege                        ;ha egyenlő, sikeres
      btfss     C
      goto      kovet                       ;ha kevés, következő
      sub       gyok,proba,2                 ;túl sok volt, kivonás
kovet: srl      proba,2                       ;következő bit jobbra tolással
      decfsz    ciklus,f                     ;ciklusszámláló csökkentése
      goto      ujra
vege:  endm
```

Mintavett adatok késleltetése

Egyes mérési algoritmusok több mintavétellel (pl. 32 mintával) korábbi adatokkal dolgoznak. A mért adatokat ennek megfelelő hosszban tárolni kell. A tárolás módszere lehet FIFO vagy ciklikus tár.



- FIFO tár - nagyobb mintaszám esetén sok időt vesz igénybe
- ciklikus tár - csak egy adat cseréje és a mutató léptetése szükséges. Figyelni kell, hogy a mutató még a tárra mutat-e! Indirekt címzéssel kezelhető. A DSP processzorok beépített gyors ciklikus tár kezelő utasításokkal rendelkeznek

Ciklikus tár programozása I.

;ciklikus tár 16 byte hosszan=16 db 8 bites adat
;bemenő adat: ujadat
;mutató = FSR0 indirekt regiszter (L,H)

```
csere: movff      INDF0,WREG      ;ciklikus tárból ki  
      movff      ujadat,POSTINC0 ;mozgatás és pointer inkrementálása  
      movff      WREG,ujadat     ;kivett adat eltárolás
```

Eddig az adatmozgatás, utána tár szervezés 3 féle módon:

a, A ciklikus tár tetszőleges helyen szerepel:

```
      movlw      vege             ;vege betöltése teszthez  
      subwf      FSR0L           ;vege és FSR0L pointer komparálása  
      movlw      ciktar          ;ciklus elejének beírása (Z nem vált.)  
      btfsc      Z,0            ;ha nincs vége ugord át  
      movwf      FSR0L,0        ;pointer frissítése  
      return                    ;visszatérés a szubrutinból
```

Ciklikus tár programozása II.

b, A ciklikus tár 10h-val (1.0000b) osztható kezdőcímen van. Ilyenkor csak az alsó négy bit pörög, ha ez túlpörgött, akkor az e felett lévő biteket helyre állítom és kezdődik előről.

```
movf      FSR0L,W,0    ;FSR0L betöltése W-regbe
andlw     00F          ;logikai AND 0F-el
iorlw     ciktar       ;logikai Kizáró VAGY ciktar-ral
movwf     FSR0L,0     ;W-reg betöltése FSR0L-be
return    ;visszatérés a szubrutinból
```

c, A ciklikus tár 20h-val (10.0000b) osztható kezdő címen van. Ilyenkor is csak az alsó négy bit pörög, és az 5. bit 0 kivéve ha túlléptem a ciklus végét, mert akkor 1-re áll be. Ha ezt a bitet minden ciklusban törölöm, akkor amikor túlpörgött helyre állítja a kezdő címet és ezzel kezdődik előről, egyébként nem csinál semmi.

```
bcf      FSR0L,4,0    ;FSR0L,4 bit törlése
return   ;visszatérés a szubrutinból
```

Makró kifejtése MPASM assembler által

```
00025          NEG    10,4
00000004      M NN   SET    4
              M      WHILE  NN > 0
000038 1E13   M      COMF  10+NN-1,F,A
00000003      M NN   SET    NN-1
00003A 1E12   M      COMF  10+NN-1,F,A
00000002      M NN   SET    NN-1
00003C 1E11   M      COMF  10+NN-1,F,A
00000001      M NN   SET    NN-1
00003E 1E10   M      COMF  10+NN-1,F,A
00000000      M NN   SET    NN-1
              M      ENDW
              M      INC    10,4
000040 2A13   M      INCF  10+4-1,F,A
000042 0E00   M      MOVLW 0
00000003      M NN   SET    4-1
              M      WHILE  NN > 0
000044 2212   M      ADDWFC 10+NN-1,F,A
00000002      M NN   SET    NN-1
000046 2211   M      ADDWFC 10+NN-1,F,A
```


Program fejlesztés assemblerrel vagy C fordítóval?

	assembler	C fordító
hordozhatóság	gépfüggő	hordozható (?)
futási idő	gyors	lassúbb
real time rendszerhez	alkalmas	kevésbé alkalmas
program memória igény	kevés	több
program optimalizálás	lehetséges	nem lehetséges
fejlesztői munka igény	sok	kevés
fordító	ingyenes	nem ingyenes
processzor típus	mikrokontroller, DSP	DSP

128 byte memória terület törlése különböző módszerekkel

a, MPASM assembler (futási idő ciklus, 1 tár törlése 4 utasítás)

```
tar          equ          00
hossz       equ          80
torles1:    LFSR          0,tar          ;mutató, FSR0->tar
           movlw         hossz          ;WREG=ciklus változó
torcik:     clrf          POSTINC0,0    ;FSR++=0
           decf          WREG,w,0
           bnz          torcik          ;ugrás torcik, ha nem 0
           return
```

b, MPASM assembler makró használattal (fordítási idő ciklus, sokkal gyorsabb mert 1 tár törlése 1 utasítás, de jóval hosszabb)

```
torles2:
set         n            hossz
           while n
           clrf          tar+n-1
n          set         n-1
           endw
           return
```

c, C18 compiler segítségével

```
char *tar[0x80];  
void torol(char *tar, int hossz)  
{int n;  
for (n=0;n<hossz;n++,tar++) *tar=0;}
```

Az egyes módszerek összehasonlítása

Összehasonlítás program hossza (word) és a program futási ideje (gépi ciklus) szerint:

		program hossza	program futási idő
A	asm	6	518
B	macro	128	132
C	C	104	4913

Példa szubrutin működésre

A szubrutin lefutása után wreg=?

```
                clrf  wreg
                call  rut1
rut1:           call  rut2
rut2:           call  rut3
rut3:           incf  wreg
                return
```

Példa megoldása 1. lépés

A szubrutin lefutása után wreg=?

```
                clrf  wreg          ;wreg=0
                call  rut1
rut1:           call  rut2
rut2:           call  rut3
rut3:           incf  wreg          ;wreg=wreg+1
                return
```

Példa megoldása 2. lépés

A szubrutin lefutása után wreg=?

```
        clrf    wreg        ;wreg=0
        call   rut1
rut1:   call   rut2
rut2:   call   rut3        ;2x rut3-> wreg=wreg+2
rut3:   incf   wreg        ;wreg=wreg+1
        return
```

rut2 meghívja rut3-t, végrehajtja, majd visszatérés után folytatja a következő utasításnál, ami éppen rut3.

Példa megoldása 3-4. lépés

A szubrutin lefutása után wreg=?

```
                clrf   wreg           ;wreg=0
                call  rut1           ;2x rut1-> wreg=wreg+8
rut1:           call  rut2           ;2x rut2-> wreg=wreg+4
rut2:           call  rut3           ;2x rut3-> wreg=wreg+2
rut3:           incf   wreg           ;wreg=wreg+1
                return
```

wreg=8

Digitális szignál processzorok (DSP)

Digitális szignál processzor (DSP) alkalmazása

Speciális mikroprocesszor nagymennyiségű számítást igénylő feladatok elvégzésére pl. digitális szűrő, FFT algoritmus.

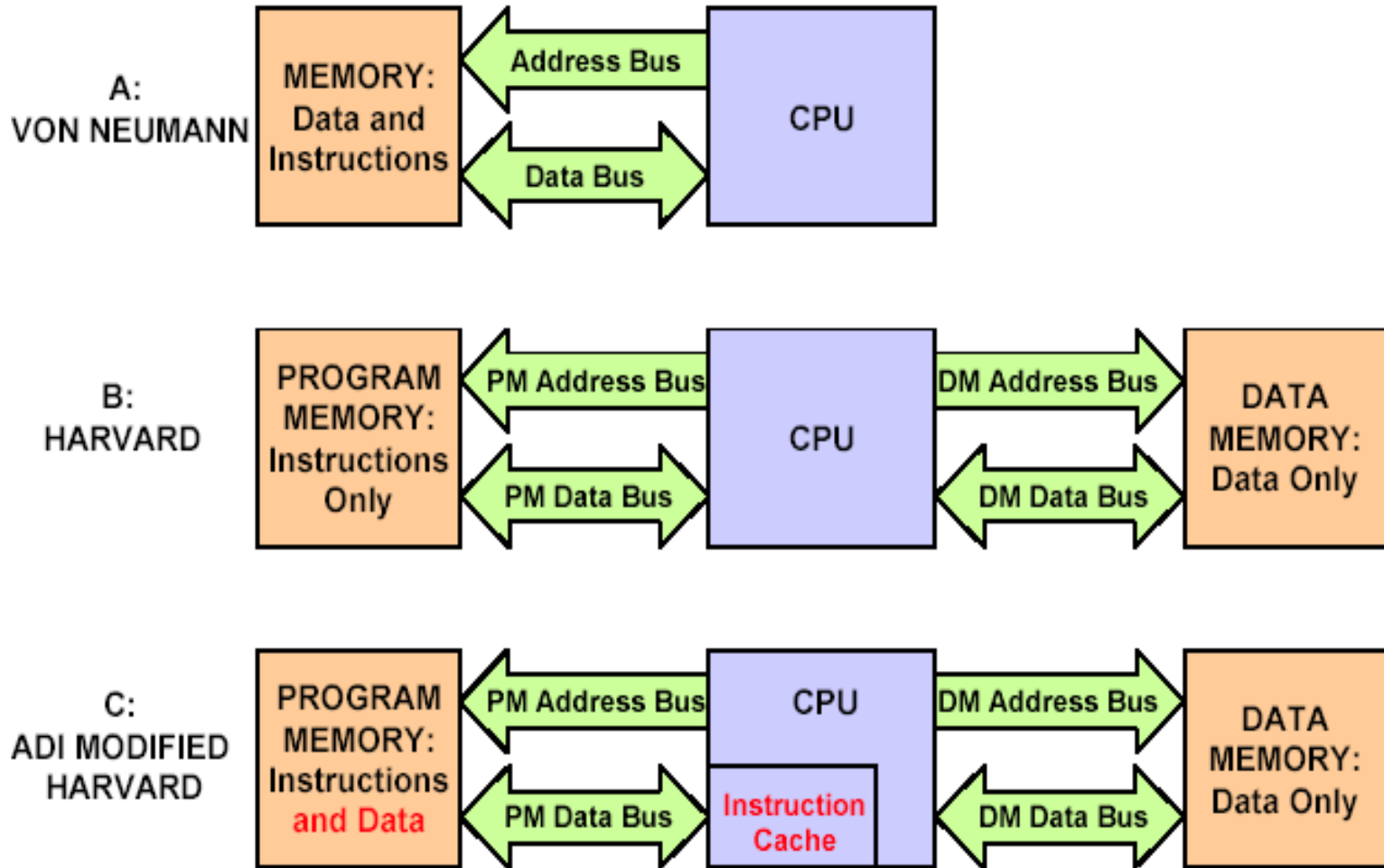
A DSP elsősorban a következő aritmetikai művelet gyors elvégzésére lett kialakítva:

$$y = \sum_{i=1}^n h(i) * x(i)$$

Jellemzői:

- külön adat és program busz (Harvard architektúra)
- külön adat és program memória
- kevés beépített periféria áramkör (timer, soros IF)
- külső kapcsolat a buszrendszeren és soros porton keresztül
- egyszerű (RISC) utasításkészlet
- gyors szorzó és összegző utasítások
- fejlett indirekt címzési módok

Mikroprocesszorok felépítése

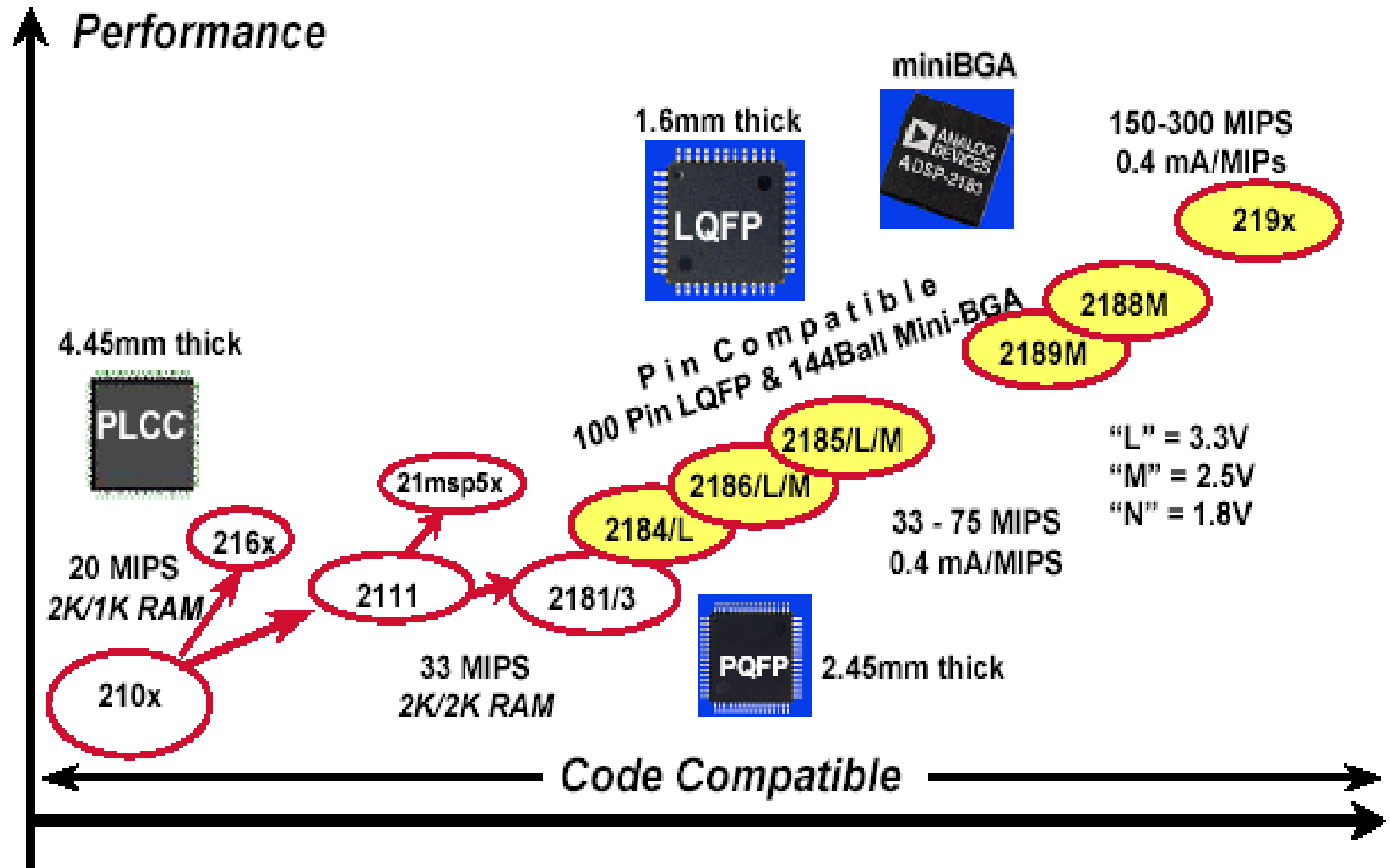


Analóg Devices ADSP 21xx típusú processzorok

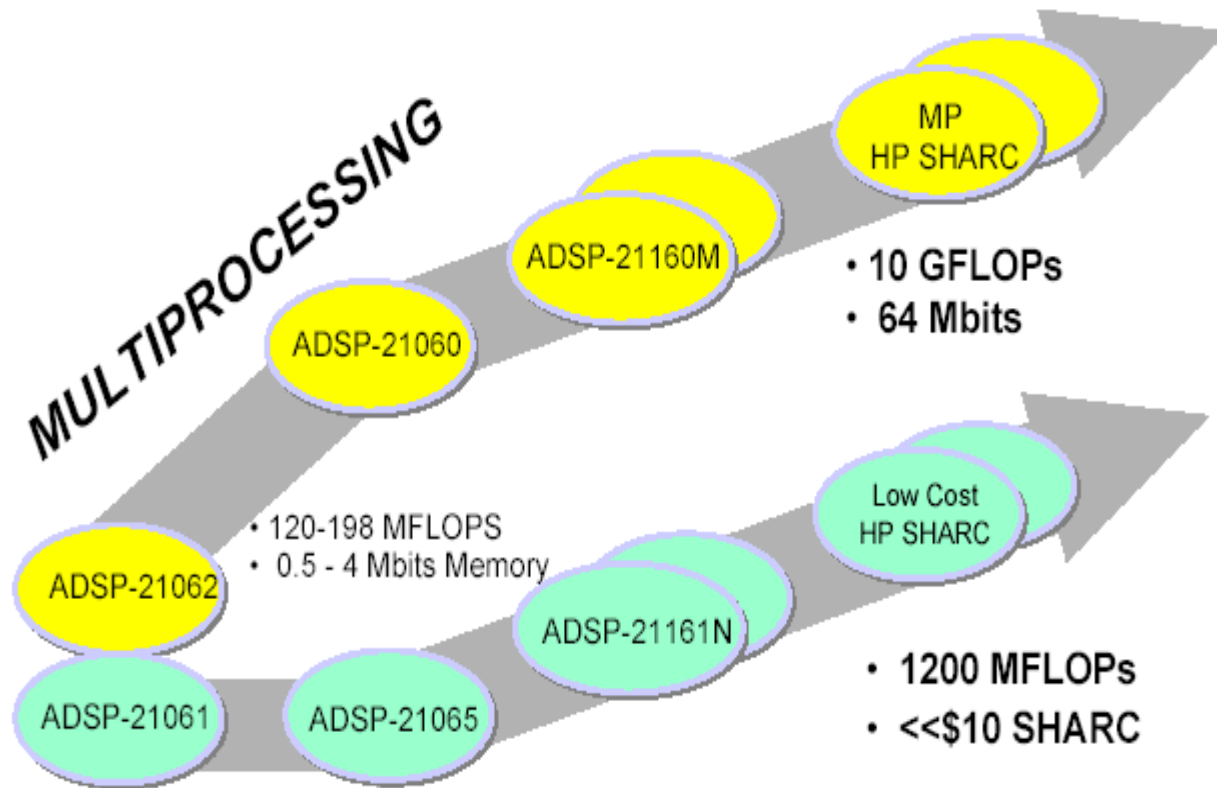
16 bites adatmemória 4..16 kWord

24 bites programmemória 4..16 kWord

fixpontos aritmetika

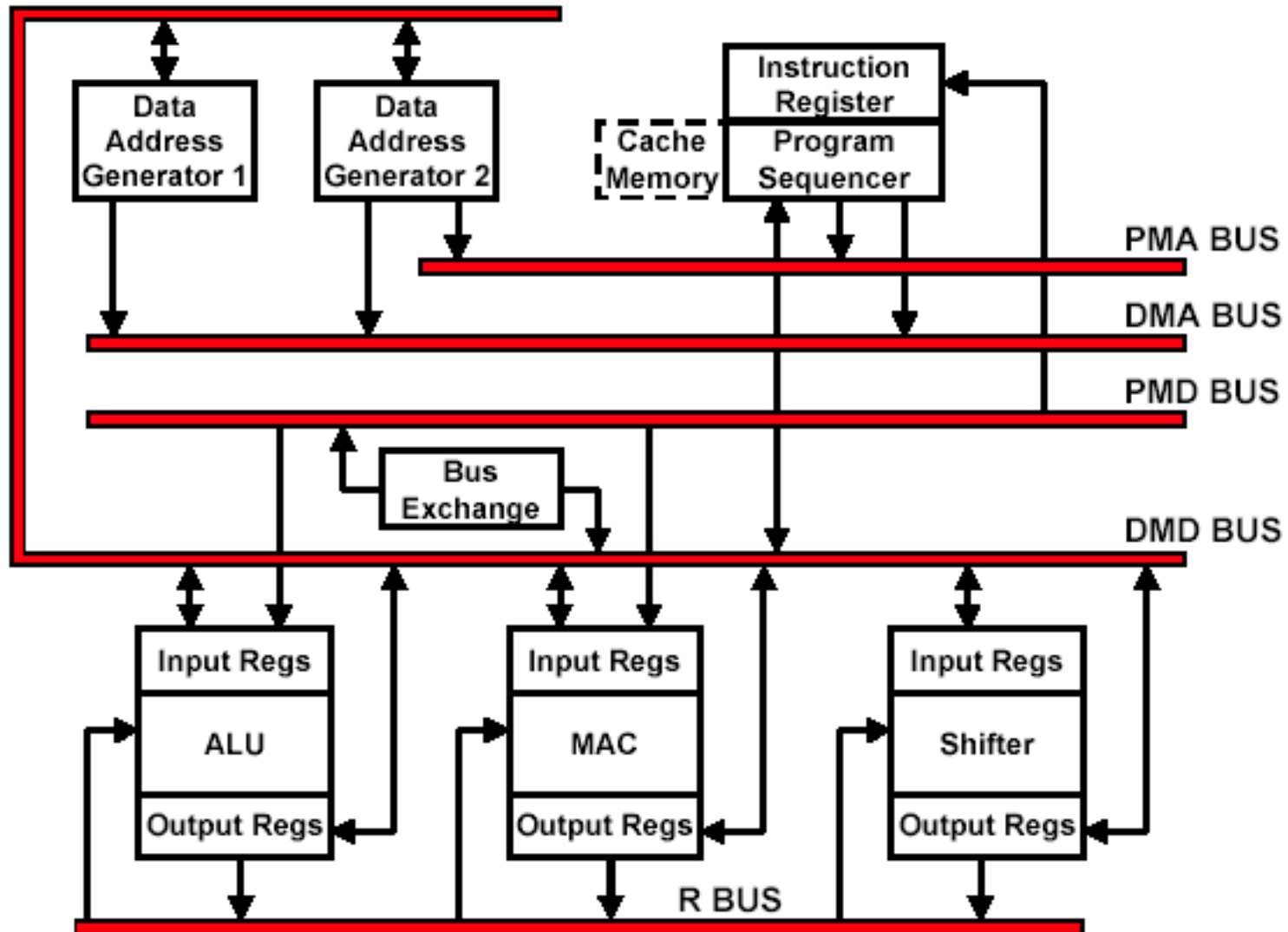


ADSP 21xxx 32 bites lebegőpontos processzorok



<http://www.analog.com/processors/sharc/overview/archOverview.html>

ADSP218x processzor belső felépítése



ADSP218x processzor felépítése

1. Aritmetikai-logikai egység

összeadás

kivonás

logikai műveletek

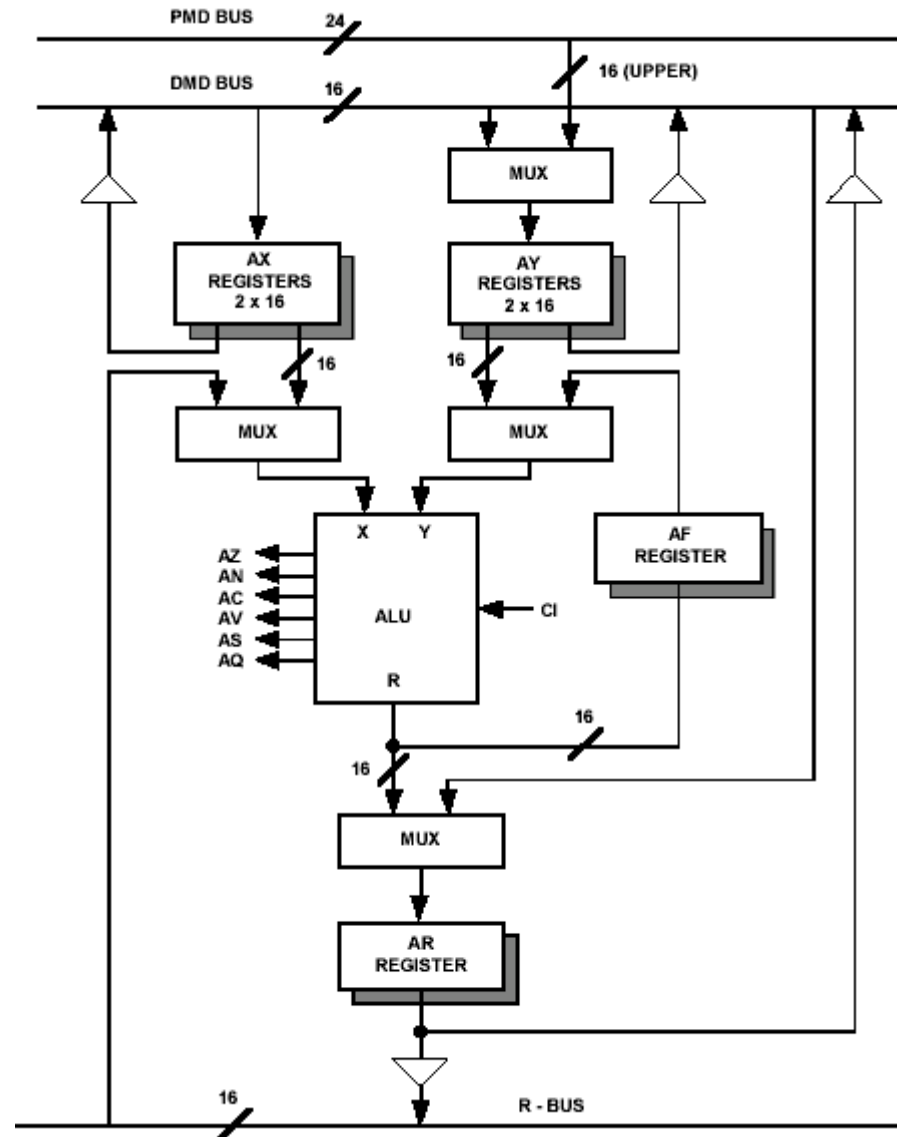
increment, decrement

osztás

pl.:

$$AR = AX0 + AY0$$

$$AR = AX1 \text{ and } AY1$$



2. Szorzó egység

16 bites előjeles szorzás

összegzés

pl.:

$$MR = MX_0 * MY_0 \text{ (SS)}$$

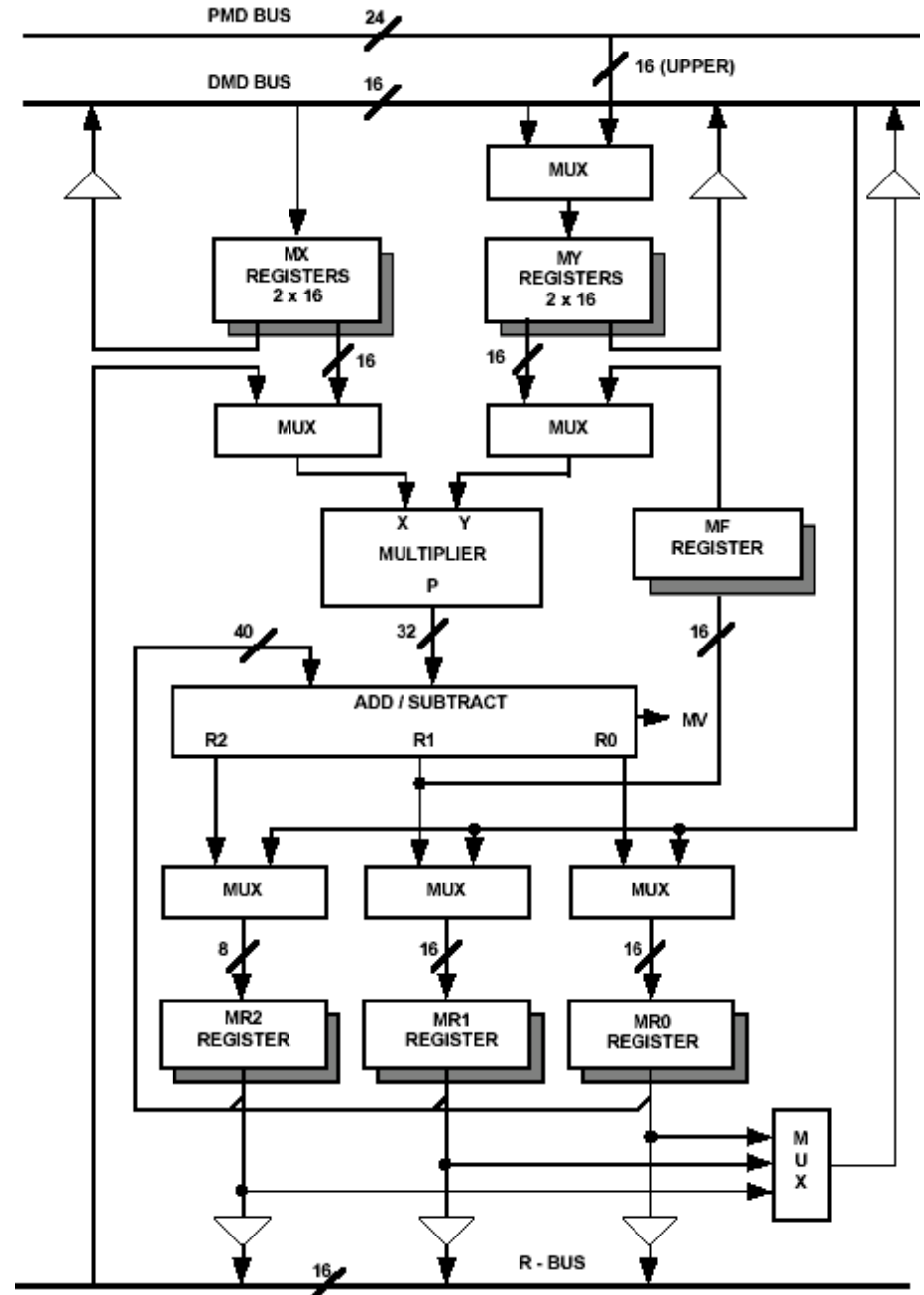
előjeles szorzás

$$MR = MR + MX_1 * MY_1 \text{ (UU)}$$

előjel nélküli szorzás és

a szorzat hozzáadása az

MR 40 bites regiszterhez



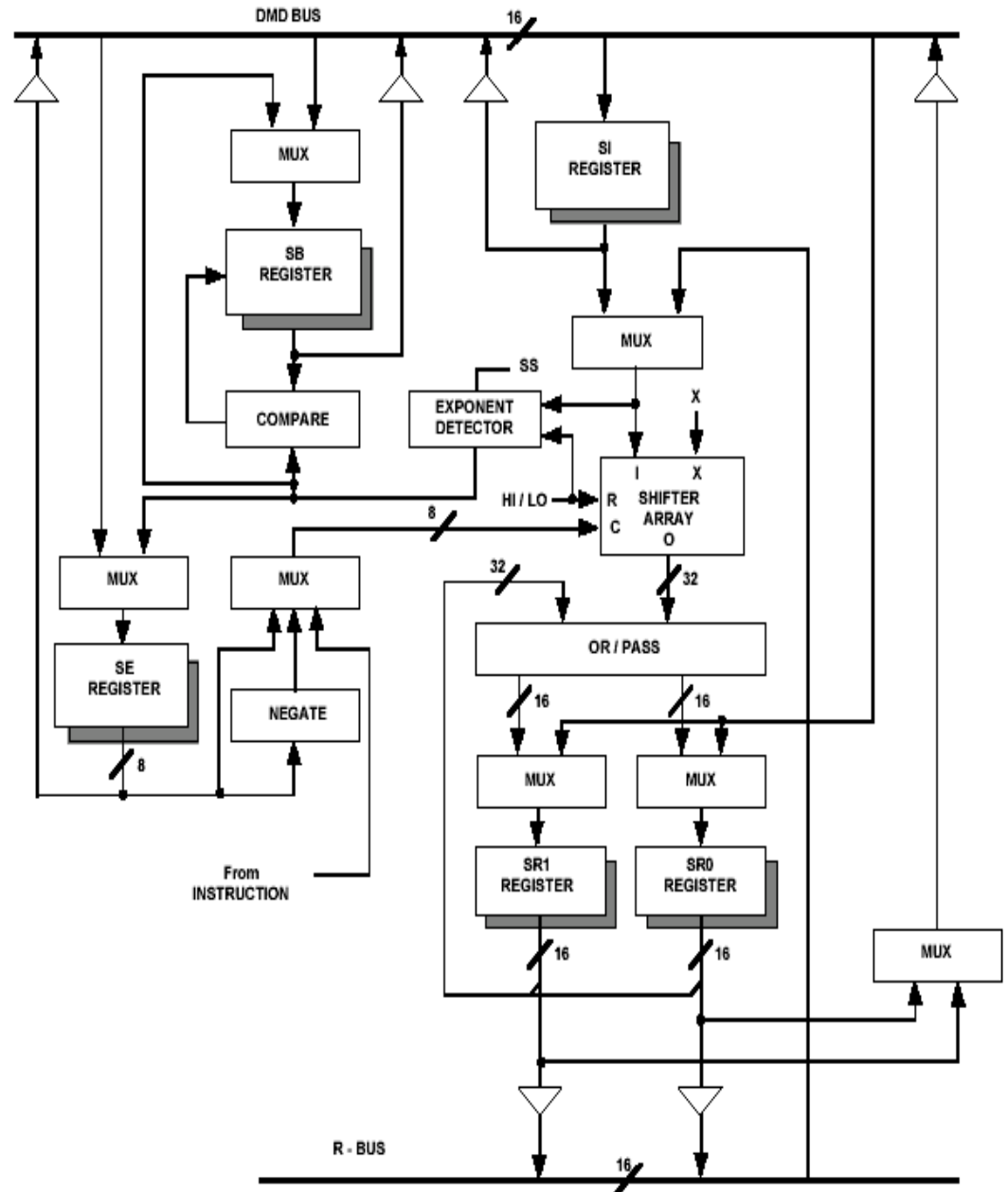
3. Shifter

- aritmetikai shift
- logikai shift
- normalizálás

pl.:

SR=ASHIFT AR BY 4

Az AR regiszter 4 bittel elcsúsztatva kerül a 32 bites SR regiszterbe.

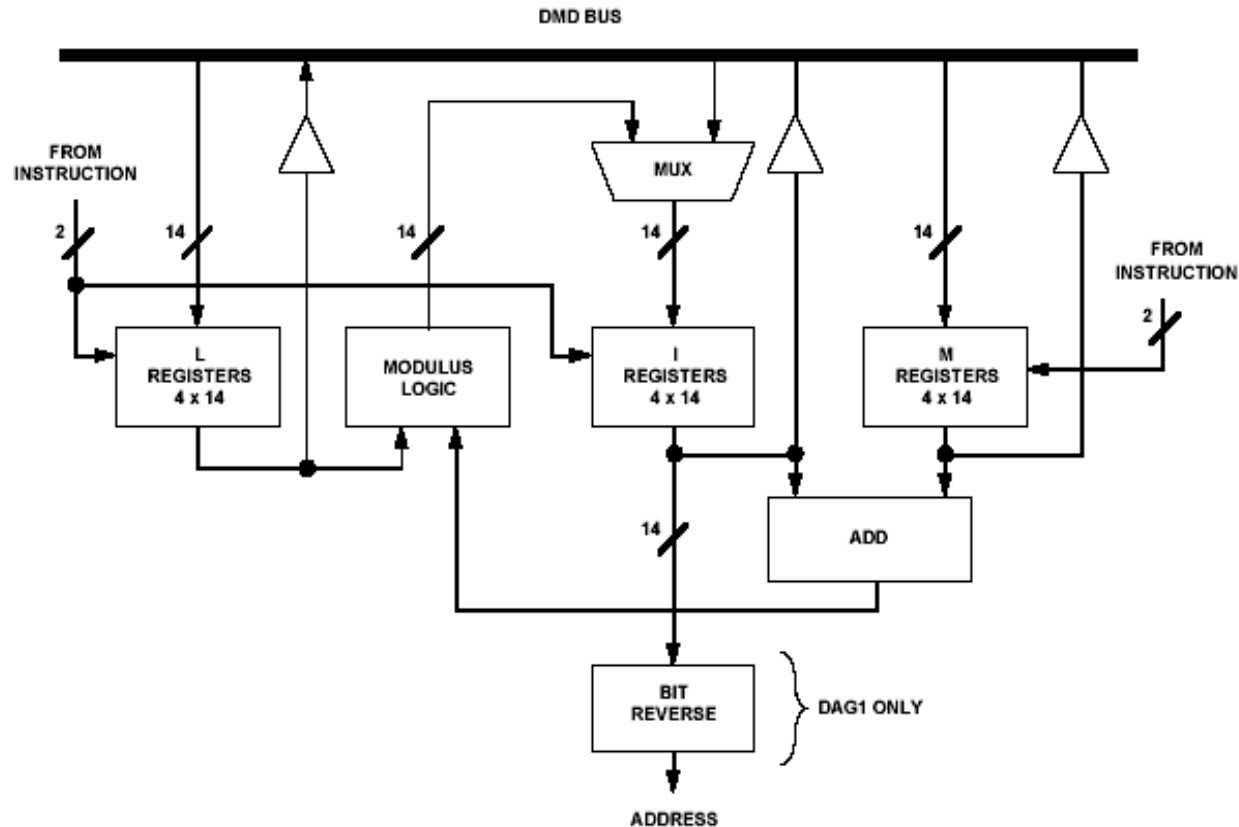


Indirekt címzés és címgenerátor

8 címgenerátor van, mindegyikben I, M és L regiszterrel

pl.: $AX0 = DM(I1, M1)$

Az I1 címen lévő adatot betölti az AX0 regiszterbe,
majd az I1 regisztert módosítja $I1 = I1 + M1$



1. Multifunkciós utasítások

- a. Aritmetikai művelet és adatmozgatás egy utasításban

CNTR = N ;CNTR-be tölti N ciklusszámot
do sum until CE ;ciklus szerv. feltételvizsgálattal

sum: AR=AR+AY0, AY0=DM(I0,M0)

Összeadás, majd az AY0 regiszter beolvasása, majd az I0 regiszter módosítása és a művelet ismétlése.

- b. Szorzás, összeadás és adatmozgatás egy utasításban

MR=MR+MX0*MY0(SS), MX0=DM(I0,M0), MY0=PM(I4,M4)

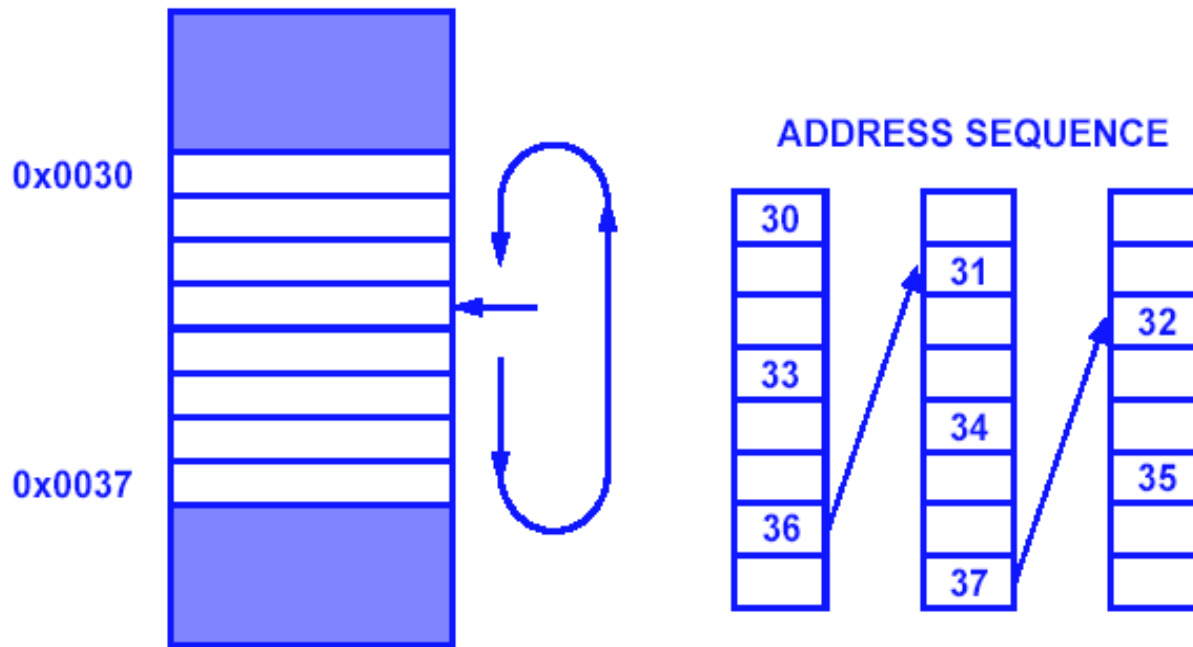
Előjeles szorzás, a szorzat hozzáadása az MR regiszterhez, majd az MX0 és a MY0 regiszterek beolvasása, majd az I0 (adat memória) és I4 (program memória) címek módosítása

2. ciklikus memória használata

A memória hossza az L regiszterben van (itt L=8)

A kezdőcím az I regiszterben (itt I=30)

A lépésköz az M regiszterben (itt M=3)



$AX1 = DM(I1, M1)$; $AX1 = [I]$ utána $I = I + M$ és ciklus vizsgálat

3. Bit reverse címzés

cím [D]	0	1	2	3	4	5	6	7
cím[B]	000	001	010	011	100	101	110	111
bitrev. cím [B]	000	100	010	110	001	101	011	111
bitrev. cím [D]	0	4	2	6	1	5	3	7

pl. A 1 címen lévő adat a 4 címre kerül

Az I4 indirekt címen lévő adatokat bit fordított címekre helyezi el az I0 indirekt címre:

```
CNTR = N;  
ENA BIT_REV;      {Enable bit-reversed outputs on DAG1}  
DO brev UNTIL CE;  
    AY1=DM(I4,M4);    {Read sequentially ordered data}  
brev:            DM(I0,M0)=AY1;  
                {Write data in bit-reversed location}  
DIS BIT REV;     {Disable bit-reverse}
```

4. Szinusz érték számítás közelítéssel (0-90°-ig)

A szinusz függvény értékét az alábbi 5-öd rendű sorral számítjuk:

$$\sin(x) = 3.140625x + 0.02026367x^2 - 5.325196x^3 + 0.5446778x^4 + 1.800293x^5$$

Az AR regiszter az x értéket tartalmazza,
MF segéd regiszterben az x hatványait tároljuk,
I3 pointer az együttható táblázatra mutat.

```
MY1=AR;
MF=AR*MY1 (RND), MX1=DM(I3,M3); {MF = x2}
MR=MX1*MY1 (SS), MX1=DM(I3,M3); {MR = C1x}
CNTR=3;
DO approx UNTIL CE;
    MR=MR+MX1*MF (SS);
approx:    MF=AR*MF (RND), MX1=DM(I3,M3);
    MR=MR+MX1*MF (SS);
```

Microchip dsPIC típusú processzor

Egyesíti a mikrokontroller és a DSP processzor tulajdonságait

16 bites adat hossz

16 bites aritmetika és DSP típusú utasításkészlet

megmaradt a sokoldalú periféria készlet

max 80 MHz órajel

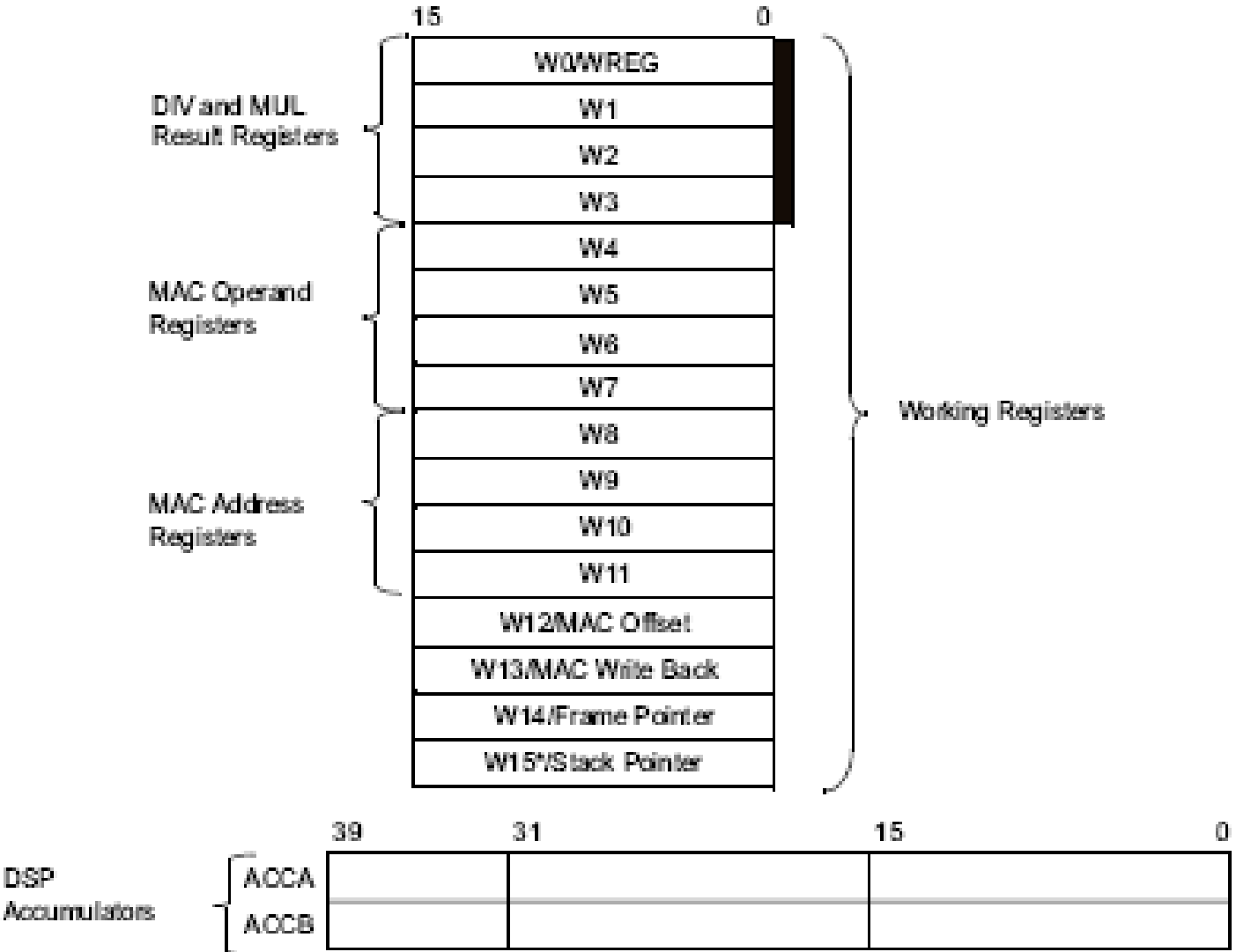
új fordítók: ASM30, C30

Device	Pins	Program Memory		SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Capt.	Output Compare/ Std PWM	Codec Interface	A/D 12-bit 100 Ksps	UART	SPI	I ² C	CAN
		Bytes	Instructions											
dsPIC30F3014	40/44	24K	8K	2048	1024	3	2	2	—	13 ch	2	1	1	—
dsPIC30F4013	40/44	48K	16K	2048	1024	5	4	4	AC97, I2S	13 ch	2	1	1	1
dsPIC30F5011	64	66K	22K	4096	1024	5	8	8	AC97, I2S	16 ch	2	2	1	2
dsPIC30F6011	64	132K	44K	6144	2048	5	8	8	—	16 ch	2	2	1	2
dsPIC30F6012	64	144K	48K	8192	4096	5	8	8	AC97, I2S	16 ch	2	2	1	2
dsPIC30F5013	80	66K	22K	4096	1024	5	8	8	AC97, I2S	16 ch	2	2	1	2
dsPIC30F6013	80	132K	44K	6144	2048	5	8	8	—	16 ch	2	2	1	2
dsPIC30F6014	80	144K	48K	8192	4096	5	8	8	AC97, I2S	16 ch	2	2	1	2

DSP típusú utasítások:

Assembly Syntax	Description	Words	Cycles	Page #
ADD Acc	Add accumulators	1	1	5-11
ADD Ws,#Sli14,Acc	16-bit signed add to Acc	1	1	5-12
CLR Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Acc	1	1	5-77
ED Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean distance (no accumulate)	1	1	5-111
EDAC Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean distance	1	1	5-113
LAC Ws,#Sli14,Acc	Load Acc	1	1	5-133
MAC Wm*Wn,Acc,Wx,Wxd,Wy,Wyd,AWB	Multiply and accumulate	1	1	5-141
MAC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square and accumulate	1	1	5-143
MOVSAC Acc,Wx,Wxd,Wy,Wyd,AWB	Move Wx to Wxd and Wy to Wyd	1	1	5-159
MPY Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Multiply Wn by Wm to Acc	1	1	5-161
MPY Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square to Acc	1	1	5-163
MPY.N Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wn by Wm) to Acc	1	1	5-165
MSC Wm*Wn,Acc,Wx,Wxd,Wy,Wyd,AWB	Multiply and subtract from Acc	1	1	5-167
NEG Acc	Negate Acc	1	1	5-183
SAC Acc,#Sli14,Wd	Store Acc	1	1	5-216
SAC.R Acc,#Sli14,Wd	Store rounded Acc	1	1	5-218
SFTAC Acc,#Sli16	Arithmetic shift Acc by Sli16	1	1	5-223
SFTAC Acc,Wn	Arithmetic shift Acc by (Wn)	1	1	5-224
SUB Acc	Subtract accumulators	1	1	5-235

dsPIC processzor munkaregiszterei



dsPIC processzor ADD utasítás címzési módjai

	arg1	arg2	arg3	művelet	operanduszok
ADD	f			$f=f+W0$	
ADD	f	W0		$W0=f+W0$	
ADD	#lit10	Wn		$Wn=Wn+\#lit10$	$Wn \in [W0..W15]$ $lit10 \in [0..1023]$
ADD	Wb	#lit5	Wd [Wd] [Wd++] [Wd--] [++Wd] [--Wd]	$Wd=Wb+\#lit5$	$Wb \in [W0..W15]$ $Wd \in [W0..W15]$ $lit5 \in [0..31]$
ADD	Wb	Ws [Ws] [Ws++] [Ws--] [++Ws] [--Ws]	Wd [Wd] [Wd++] [Wd--] [++Wd] [--Wd]	$Wd=Wb+Ws$	$Wb \in [W0..W15]$ $Ws \in [W0..W15]$ $Wd \in [W0..W15]$

- 16 bites művelet
- Sokfajta indirekt címzési lehetőség
- Összeadás 3 argumentummal
- 16 különböző munka regiszter választható

Új lehetőségek a PIC kontrollerekhez képest

1. Gyors program ciklusok szervezése:

repeat #n ;következő utasítást n-szer hajtja végre

do #n,vege ; következő utasításokat n-szer hajtja végre
utasítások

vege: utolsó utasítás

2. Új címzési módok:

modulo címzés: ciklikus tár automatikus körbejárása

bit reverse címzés

3. Előjeles szorzás 16x16 biten

4. Szorzás, összeadás és operandusok beolvasása egy utasításban

mac W4*W5, A, [W8]+=2, W4, [W10]+=2, W5

Memória terület törlése különböző módszerekkel

ASM:

```
torol:  mov    #0,W0
        mov    #mintak,W1
        repeat #0x7f
        mov    W0,[W1++]
        return
```

C:

```
int      *tar[0x80];
void torol(int *tar, int hossz)
{int n;
  for (n=0;n<hossz;n++,tar++)
    *tar=0;}
```

		program hossza	program futási idő usec
PIC	asm	6	51.8
PIC	macro	128	13.2
PIC	C	104	491.3
dsPIC	asm	5	7.15
dsPIC	C	40	11.5

Algoritmusok időigénye dsPIC processzor esetén

Function	Cycle Count Equation	Conditions*	Number of Cycles	Execution Time @30 MIPS
Complex FFT**	-	N=64	3675	121.3 μ s
Complex FFT**	-	N=128	8294	273.7 μ s
Complex FFT**	-	N=256	18545	612.0 μ s
Block FIR	$49+N(4+M)$	N=32, M=32	1201	39.6 μ s
Block FIR Lattice	$49+N(4+7M)$	N=32, M=32	7345	242.4 μ s
Block IIR Canonic	$28+N(8+7S)$	N=32, S=4	1180	38.9 μ s
Block IIR Lattice	$41+N(27+9(M-1))$	N=32, M=8	2921	96.4 μ s
Matrix Add	$13+3(C*R)$	C=8, R=8	205	6.8 μ s
Matrix Transpose	$10+C(6+3(R-1))$	C=8, R=8	226	7.5 μ s
Vector Dot Product	$12+3N$	N=32	108	3.6 μ s
Vector Max	$8+7(N+1)$	N=32	239	7.9 μ s
Vector Multiply	$12+4N$	N=32	140	4.6 μ s
Vector Power	$12+2N$	N=32	76	2.5 μ s

Algoritmusok időigénye ADSP TS201 TigerSHARC processzor esetén

TigerSHARC Processor Benchmarks

Peak Rates at 600 MHz		
1-Bit Performance	32 Billion MACs/second	
16-Bit Performance	4.8 Billion MACs/second	
32-Bit Fixed-Point Performance	1,200 Million MACs/second	
32-Bit Floating-Point Performance	3,600 MFLOPS	
16-Bit Algorithms	Execution Time at 600 MHz	Clock Cycles
256-Point Complex FFT (Radix 2)	1.5 μ s	901
50-Tap FIR on 1024 Input	12 μ s	7,200
Single FIR MAC	.23 ns	0.14
Single Complex FIR MAC	.95 ns	0.57
32-Bit Algorithms	Execution Time at 600 MHz	Clock Cycles
1024-Point Complex FFT (Radix 2)	16.8 μ s	10,061
50-Tap FIR on 1024 Input	45.8 μ s	27,500
Single FIR MAC	0.9 ns	0.54
Single Complex FIR MAC	3.6 ns	2.16

FFT spektrumanalízis időigénye különböző DSP processzorok esetén

processzor	bitszám	MHz	FFT	futási idő usec
218xx	16	80	256	93
218xx	16	80	1024	465
219xx	16	160	1024	302
21160	32	100	1024	100
21535	16	300	1024	47
TS201	32	600	1024	16,8
TMS320C6711	32	150	1024	130
dsPIC	16	80	256	635
PIC18	8	40	256	58 300

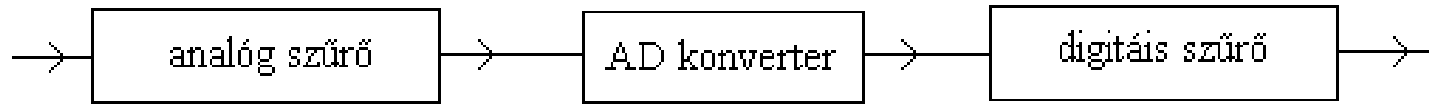
TS201 dupla aritmetikai egység 8 utasítás/ciklus
1.200.000 MAC/sec, (200 USD)

Digitális szűrők

Analog Devices www.analog.com

Tietze-Schenk: Analóg és digitális áramkörök

Analóg vagy digitális szűrő?



analóg

digitális

kevésbé pontos (alkatrész tol.)

nagyon pontos

időben változhat

stabil

nagyobb sávszélesség lehetséges

sávszélesség korlátozott

antialiasing szűrő csak analóg

real time mód korlátozott

nincs számításigény

számításigényes

korlátozott típusválaszték

nagyobb típusválaszték

Fontosabb digitális szűrőtípusok

1. Mozgó átlagolású szűrők
2. FIR (Finite Impulse Response -véges impulzus válaszijű) szűrők
3. IIR (Infinite Impulse Response -végtelen impulzus válaszijű) szűrők

Mozgó átlagolású szűrők

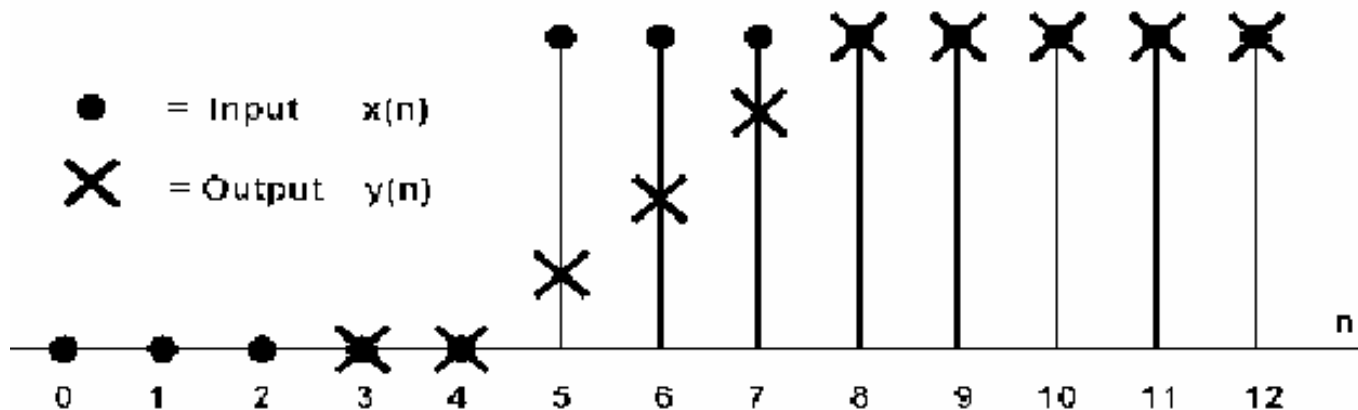
A kimenet az előző n darab bemeneti minta átlaga

A gyors változást csillapítja \rightarrow aluláteresztő jellegű

Nincs analóg megfelelője

Egyszerű algoritmus (bemenetek ciklikus tárban)

4-TAP MOVING AVERAGE FILTER STEP RESPONSE

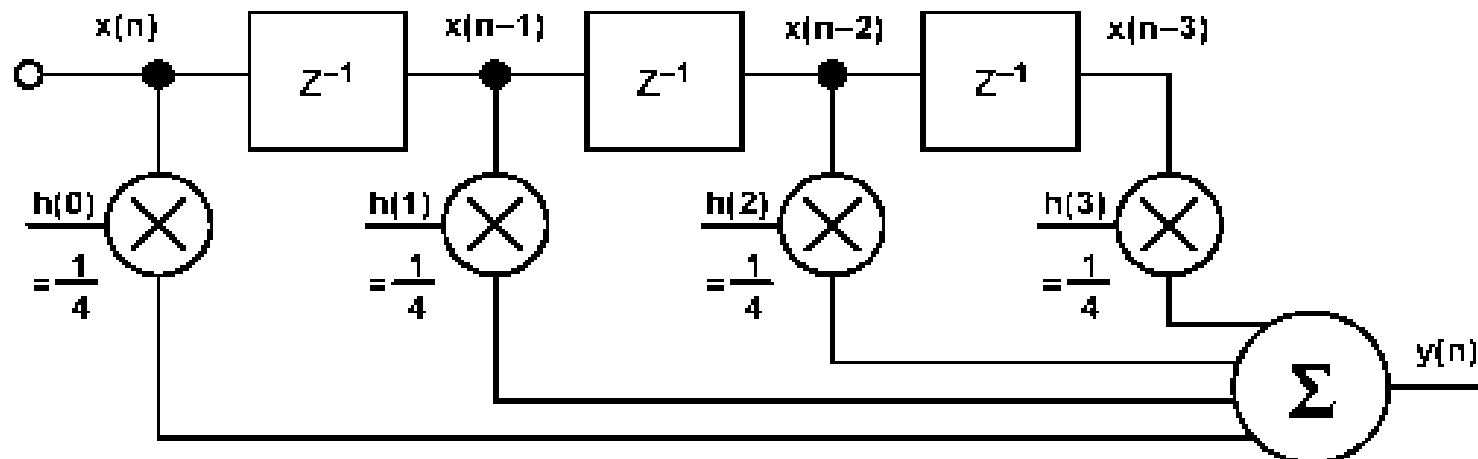


General:
$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(n-k)$$

For $N=4$:
$$y(n) = \frac{1}{4} \sum_{k=0}^3 x(n-k)$$

4 pontos mozgó átlagolású szűrő algoritmus

4-POINT MOVING AVERAGE FILTER

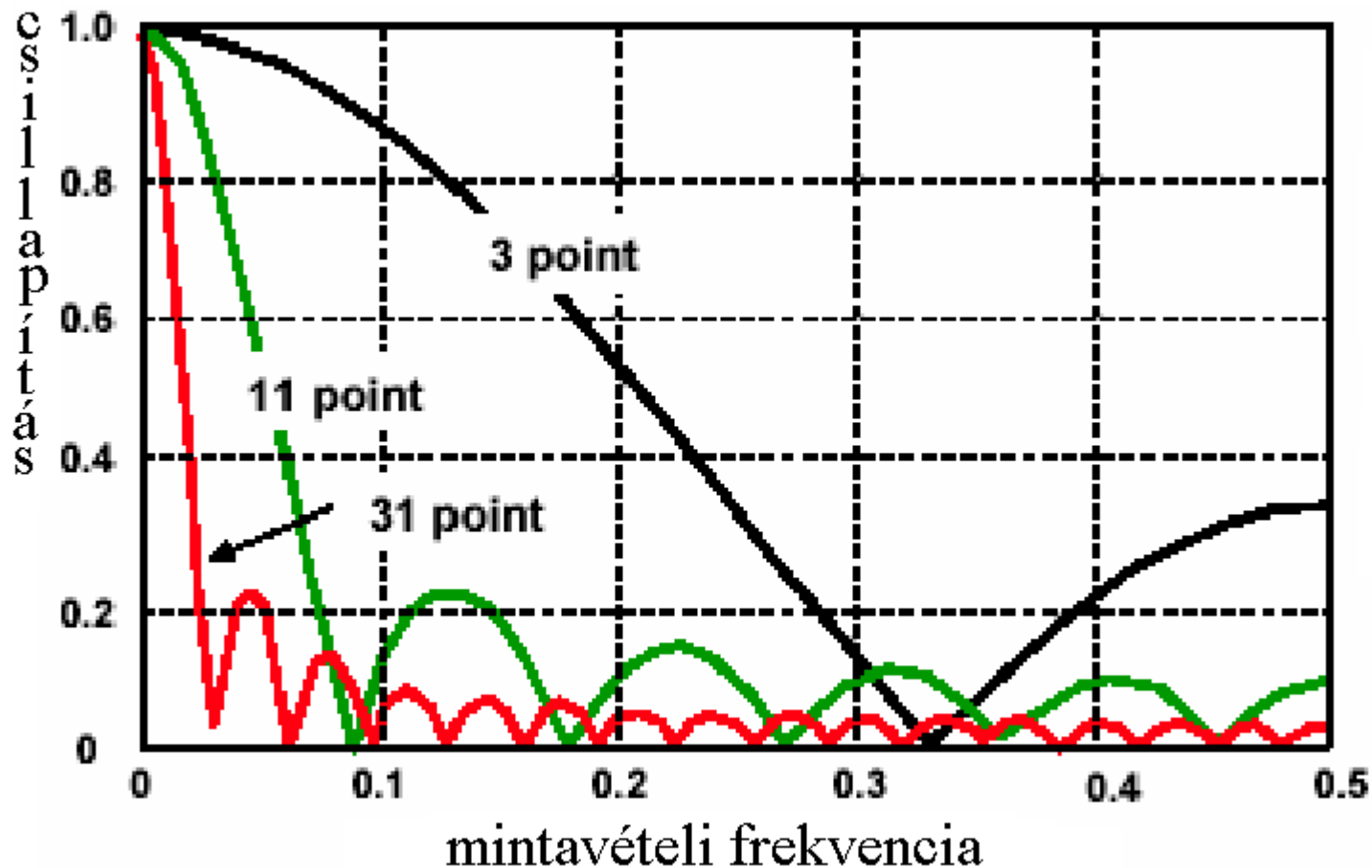


$$\begin{aligned} y(n) &= h(0) x(n) + h(1) x(n-1) + h(2) x(n-2) + h(3) x(n-3) \\ &= \frac{1}{4} x(n) + \frac{1}{4} x(n-1) + \frac{1}{4} x(n-2) + \frac{1}{4} x(n-3) \\ &= \frac{1}{4} [x(n) + x(n-1) + x(n-2) + x(n-3)] \end{aligned}$$

For N-Point

Moving Average Filter:
$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(n-k)$$

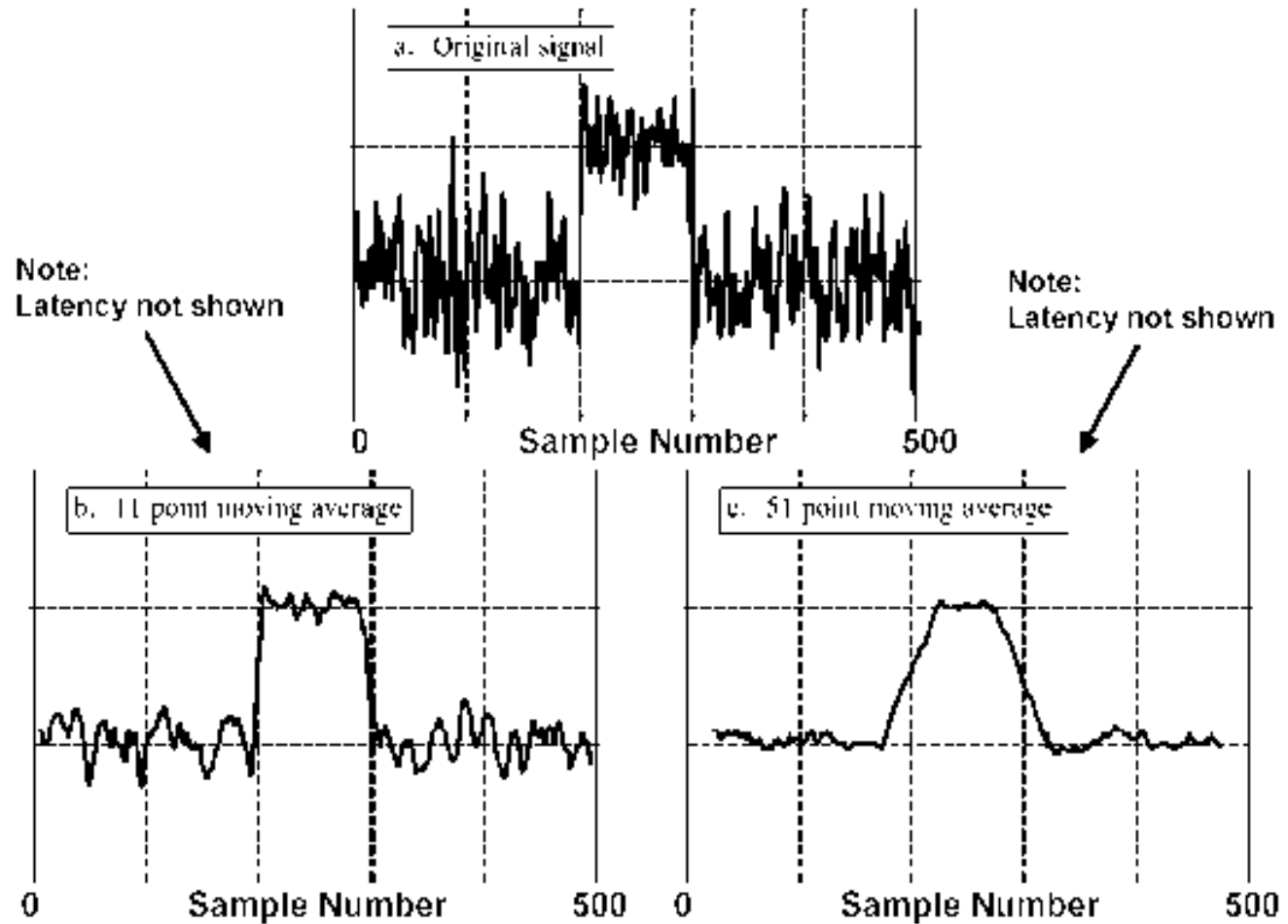
Mozgó átlagolású szűrő frekvenciamenete



31 pontos szűrő csillapítása csak kb. 0.1 - > 20 db

Mozgó átlagolású szűrő alkalmazása zajelnyomás céljára

MOVING AVERAGE FILTER RESPONSE TO NOISE SUPERIMPOSED ON STEP INPUT



FIR (Finite Impulse Response) szűrők

A mozgó átlagolású szűrőkből származtatható, ha az átlagolások súlyozása különböző.

Az impulzus válasz véges idejű

Lineáris fáziskarakterisztika, konstans csoportfutási idő

Nincs analóg megfelelője (nincs jel azonos időbeli eltolása)

Mindig stabil (nincs visszacsatolás)

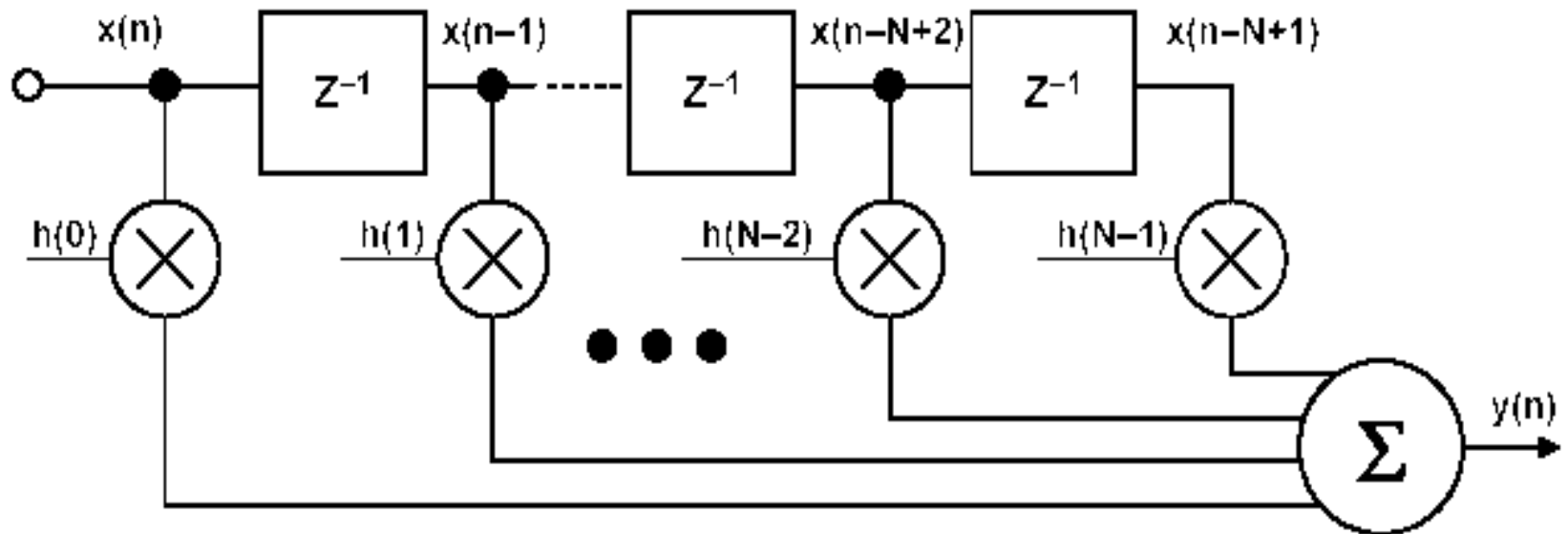
Igény szerinti frekvencia karakterisztika megvalósítható

Számításigényes, akár több száz pontos is lehet

Mikrokontrollerekben nehezen alkalmazható (DSP-nek jó)

Méretezéshez, fejlesztéshez programok vannak

FIR szűrő algoritmus

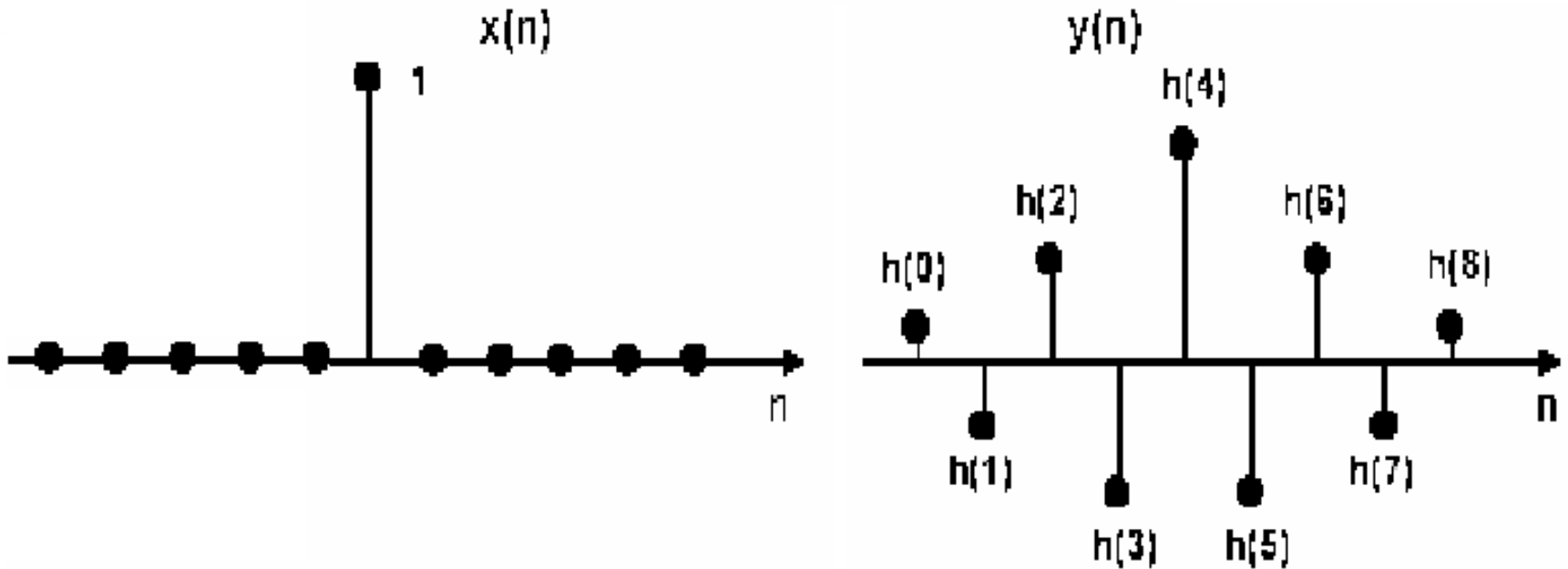


■ $y(n) = h(n) * x(n) = \sum_{k=0}^{N-1} h(k) x(n-k)$

■ * = Symbol for Convolution

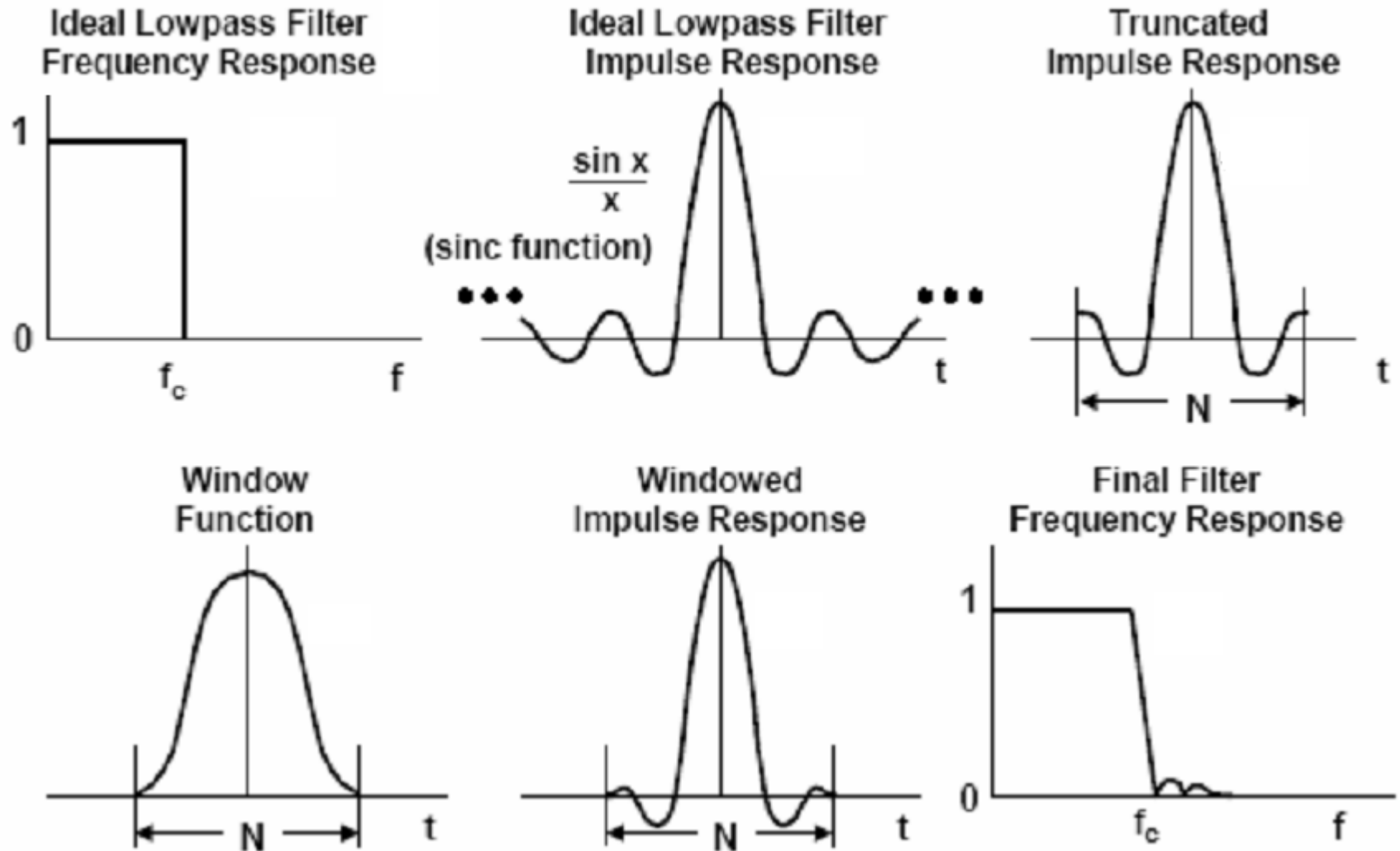
■ Requires N multiply-accumulates for each output

FIR szűrő válasza impulzus bemenetre



Impulzus bemenet esetén a kimenet maguk az együtthatók

FIR szűrő méretezése



IIR (Infinite Impulse Response) szűrők

Analóg szűrőkből származtatható frekvencia transzformáció segítségével, azok digitális megfelelői.

Az impulzus válasz végtelen idejű

Nemlineáris fáziskarakterisztika

Van analóg megfelelője

Instabil lehet

Csak az analóg megfelelőjéhez hasonló frekvencia karakterisztika valósítható meg

Kevésbé számításigényes

Mikrokontrollerekben alkalmazható

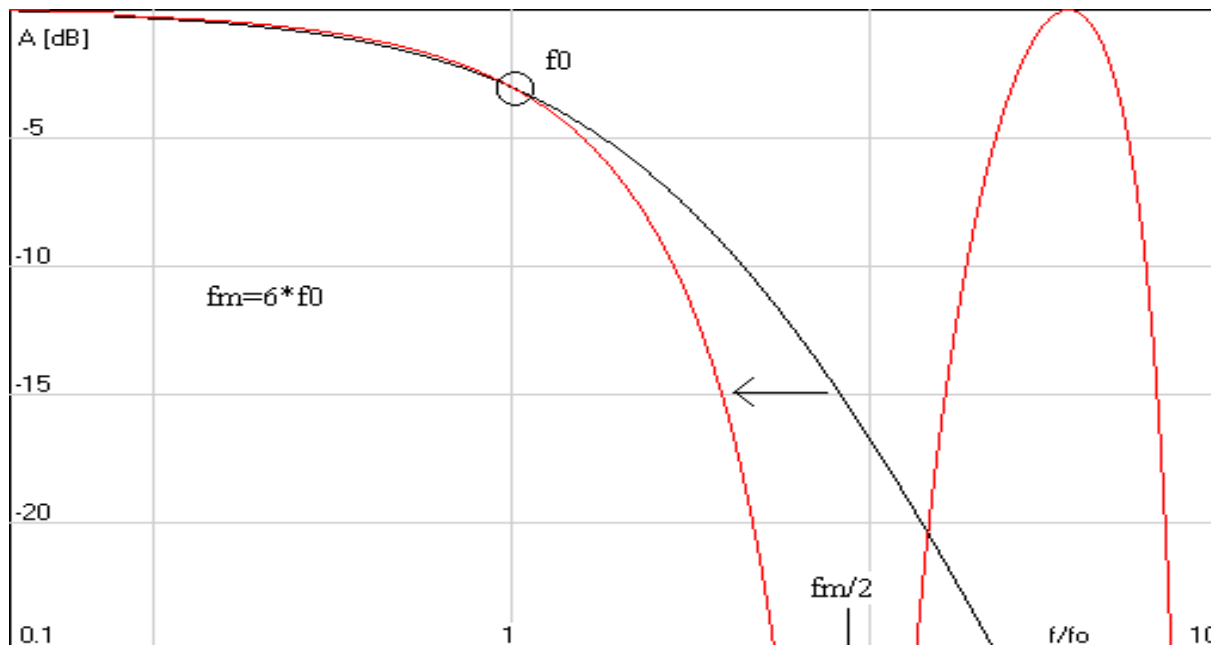
Méretezéshez, fejlesztéshez programok vannak

A bilineáris transzformáció

Az analóg szűrők frekvenciatartománya $0 \leq f < \infty$

A digitális szűrők frekvenciatartománya $0 \leq f < f_m/2$

Keresünk olyan frekvencia transzformációt, amely az analóg frekvenciatartományt átviszi a digitális tartományba oly módon, hogy az f_0 könyökfrekvencia értéke ne változzon.



A normált mintavételi frekvencia (definíció):

$$\Omega_m = \frac{f_m}{f_0} \quad \left(\frac{\text{mintavételi frekvencia}}{\text{könyökponthi frekvencia}} \right)$$

A transzformáció feltételei:

analóg	digitális
$\Omega_a=1$	$\Omega_d=1$
$\Omega_a=\infty$	$\Omega_d= \Omega_m/2$

A transzformációs egyenlet ami a fenti feltételt kielégíti:

$$\Omega_a = l * \operatorname{tg} \frac{\pi \Omega_d}{\Omega_m}$$

ahol

$$l = \operatorname{ctg} \frac{\pi}{\Omega_m}$$

A P komplex frekvencia felhasználásával és a $\operatorname{tg}(x)$ függvény komplex alakjával kapjuk:

$$\operatorname{tg}(x) = \frac{1}{j} * \frac{e^{2jx} - 1}{e^{2jx} + 1}$$

$$P_a = j * \Omega_a = j * l * \operatorname{tg} \frac{\pi P_d / j}{\Omega_m} = l * \frac{e^{2\pi P_d / \Omega_m} - 1}{e^{2\pi P_d / \Omega_m} + 1}$$

felhasználva hogy

$$\Omega_m = \frac{f_m}{f_0} = \frac{1}{f_0 T_m} \quad \text{és} \quad P_d = \frac{P_d}{2\pi f_0}$$

kapjuk

$$P_a = l * \frac{e^{P_d T_m} - 1}{e^{P_d T_m} + 1} = l * \frac{z - 1}{z + 1} \quad (e^{P_d T_m} = z)$$

Ha $F(p)$ az $f(t)$ időfüggvény operátoros alakja, akkor az

$$e^{-pT_m} F(p) = z^{-1} F(p)$$

az $f(t-T_m)$ függvény operátoros alakja.

Az $f(t-T_m)$ függvény az $f(t)$ időfüggvénynek egy mintavétellel előző értéke.

Elsőfokú digitális szűrő paramétereinek meghatározása:

Az analóg szűrő általános egyenlete:

$$A(P) = \frac{d_0 + d_1 P}{c_0 + c_1 P}$$

A digitális szűrő általános egyenlete:

$$A(z) = \frac{D_0 + D_1 z}{C_0 + C_1 z}$$

Az analóg szűrőbe behelyettesítjük az előbb levezett

$$P = l * \frac{z - 1}{z + 1} \quad \text{összefüggést}$$

Az együtthatók összehasonlításával adódik:

$$D_0 = \frac{d_0 - d_1 l}{c_0 + c_1 l} \qquad C_0 = \frac{c_0 - c_1 l}{c_0 + c_1 l}$$
$$D_1 = \frac{d_0 + d_1 l}{c_0 + c_1 l} \qquad C_1 = \frac{c_0 + c_1 l}{c_0 + c_1 l} = 1$$

ahol

$$l = \operatorname{ctg} \frac{\pi}{\Omega_m} \qquad \text{és} \qquad \Omega_m = \frac{f_m}{f_0}$$

Másodfokú digitális szűrő paramétereinek meghatározása:

Az analóg szűrő általános egyenlete:
$$A(P) = \frac{d_0 + d_1P + d_2P^2}{c_0 + c_1P + c_2P^2}$$

Másodfokú analóg szűrők paramétereai:						
	számláló			nevező		
Típus	d0	d1	d2	c0	c1	c2
aluláteresztő	1	0	0	1	a1	b1
felüláteresztő	0	1	1	b1	a1	1
sávszűrő	0	1/Q	0	1	1/Q	1
sávzáró	1	0	1	1	1/Q	1

A digitális szűrő általános egyenlete:
$$A(z) = \frac{D_0 + D_1z + D_2z^2}{C_0 + C_1z + C_2z^2}$$

Behelyettesítés után kapjuk:

$$D_0 = \frac{d_0 - d_1 l + d_2 l^2}{c_0 + c_1 l + c_2 l^2}$$

$$C_0 = \frac{c_0 - c_1 l + c_2 l^2}{c_0 + c_1 l + c_2 l^2}$$

$$D_1 = \frac{2(d_0 - d_2 l^2)}{c_0 + c_1 l + c_2 l^2}$$

$$C_1 = \frac{2(c_0 - c_2 l^2)}{c_0 + c_1 l + c_2 l^2}$$

$$D_2 = \frac{d_0 + d_1 l + d_2 l^2}{c_0 + c_1 l + c_2 l^2}$$

$$C_2 = \frac{c_0 + c_1 l + c_2 l^2}{c_0 + c_1 l + c_2 l^2} = 1$$

ahol

$$l = ctg \frac{\pi}{\Omega_m} \quad \text{és} \quad \Omega_m = \frac{f_m}{f_0}$$

Az elsőrendű szűrő digitális algoritmusa:

A komplex átviteli függvény:

$$A(z) = \frac{Y(z)}{X(z)} = \frac{D_0 + D_1 z}{C_0 + z} = \frac{D_0 z^{-1} + D_1}{C_0 z^{-1} + 1} \quad (C_1 = 1)$$

ebből az $Y(z)$ -t kifejezve ($/z$)

$$Y(z) = D_1 X(z) + z^{-1} [D_0 X(z) - C_0 Y(z)]$$

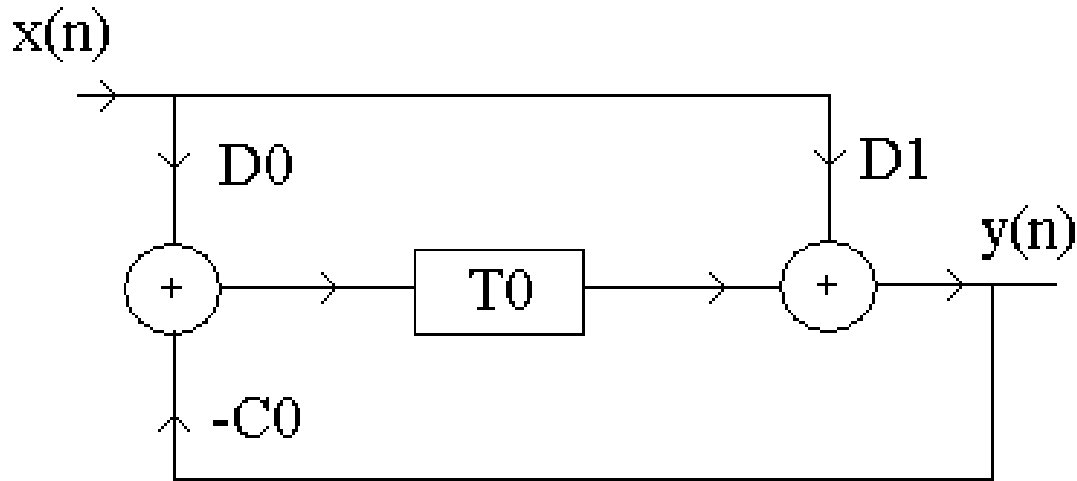
az időfüggvény a következő:

$$y(n) = D_1 x(n) + [D_0 x(n-1) - C_0 y(n-1)]$$

- Jobb oldalon is van egy „y” tehát visszacsatolás van benne!
- Elsőrendű szűrő, ezért 1 mintával korábbi értékek kellenek
- Rendkívül egyszerű algoritmus

Az elsőrendű digitális szűrő megvalósítása

A szűrő megvalósításához egy T_0 átmeneti tárat használunk:



A program lépései:

1. $y(n) = T_0(n-1) + D_1 * x(n)$
2. $T_0(n) = D_0 * x(n) - C_0 * y(n)$

Nem tároljuk külön x és y -t, mert mindig ugyan azt kell velük csinálni, a műveletet elvégezzük és az eredményt tároljuk. 22

A másodrendű szűrő digitális algoritmus:

A komplex átviteli függvény:

$$A(z) = \frac{Y(z)}{X(z)} = \frac{D_0 + D_1z + D_2z^2}{C_0 + C_1z + z^2} = \frac{D_0z^{-2} + D_1z^{-1} + D_2}{C_0z^{-2} + C_1z^{-1} + 1} \quad (C_2 = 1)$$

ebből az $Y(z)$ -t kifejezve

$$Y(z) = D_2X(z) + z^{-1}[D_1X(z) - C_1Y(z)] + z^{-2}[D_0X(z) - C_0Y(z)]$$

az időfüggvény a következő:

$$y(n) = D_2x(n) + [D_1x(n-1) - C_1y(n-1)] + [D_0x(n-2) - C_0y(n-2)]$$

ahol $x(n)$ az aktuális, $x(n-1)$ az eggyel, $x(n-2)$ a kettővel korábbi minta.

A szűrőt kétféle formában szokás megvalósítani.

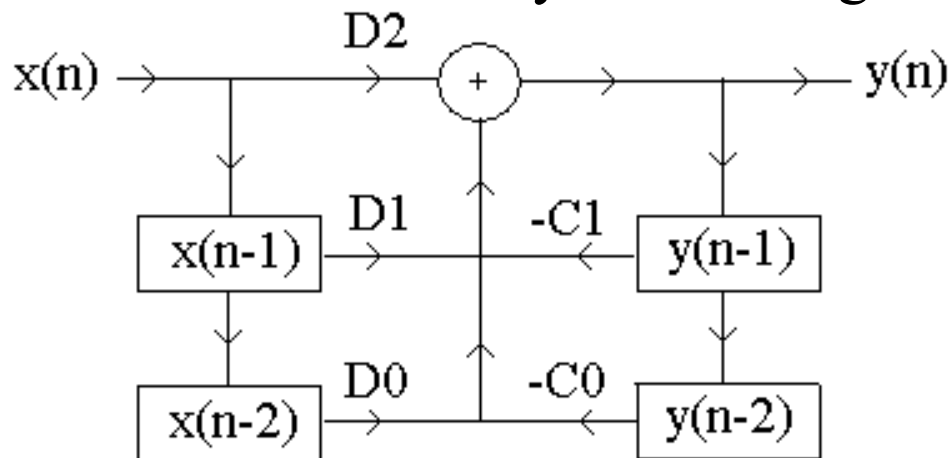
A másodrendű digitális szűrő megvalósítása I.

1. Forma:

Az előző egyenletet a következő formában írjuk fel:

$$y(n) = D_2x(n) + D_1x(n-1) + D_0x(n-2) - C_1y(n-1) - C_0y(n-2)$$

Az $y(n)$ kimenet az előző két x és y értékek segítségével számoljuk:



A számítás szorzatösszegek képzéséből áll, ezért ezt a módszert elsősorban DSP processzorok használják.

A másodrendű digitális szűrő megvalósítása II.

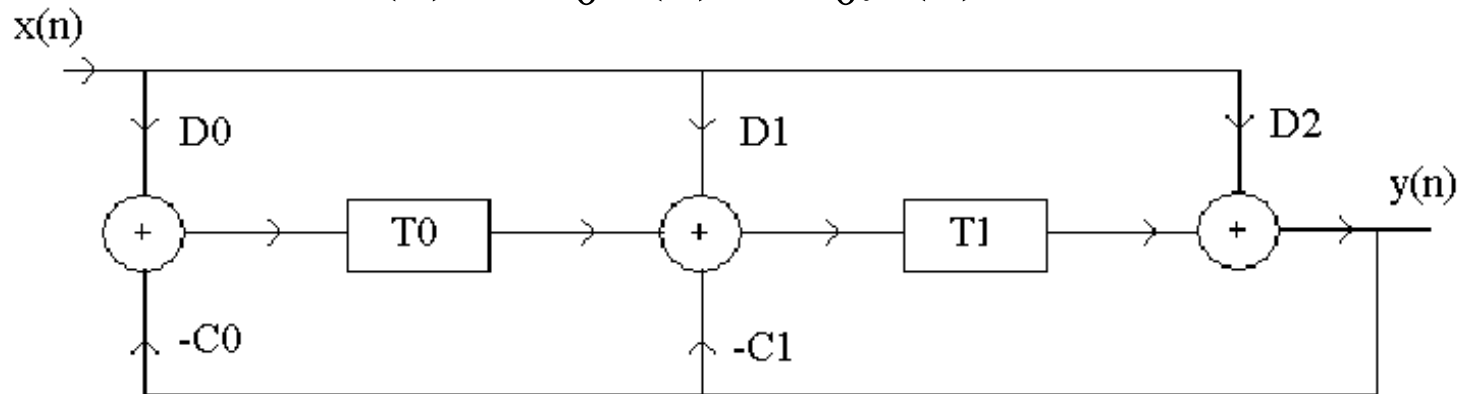
2. Forma:

Az előző egyenletet a következő formában írjuk fel:

$$y(n) = D_2 x(n) + T1(n-1)$$

$$T1(n) = T0(n-1) + D_1 x(n) - C_1 y(n)$$

$$T0(n) = D_0 x(n) - C_0 y(n)$$



Ez a forma csak két tárhelyet igényel, ezt a módszert mikrokontrollerek használják

Példa I.

2. rendű Bessel aluláteresztő szűrő méretezése, ahol $f_0=50$ Hz és $f_m=1600$ Hz

az analóg szűrő:
$$A(P) = \frac{1}{1 + 1.361P + 0.618P^2}$$

paraméterek:
$$\begin{aligned} d_0 &= 1 & d_1 &= 0 & d_2 &= 0 \\ c_0 &= 1 & c_1 &= 1.361 & c_2 &= 0.618 \end{aligned}$$

digitális szűrő méretezése:

$$\Omega_m = f_m/f_0 = 32 \quad l = \operatorname{ctg} \frac{\pi}{\Omega_m} = 10.15$$

a digitális szűrő paramétereit:

$$C_0 = 0.6479 \quad C_1 = -1.5970 \quad (C_2=1)$$

$$D_0 = 0.0127 \quad D_1 = 0.0255 \quad D_2 = 0.0127$$

(Általában amit mi használunk ott a paraméterek -2 és +2 között vannak)

Példa II.

Sávszűrő méretezése, ahol $f_0=50$ Hz, $f_m=3200$ Hz és $Q=10$

az analóg szűrő:

$$A(P) = \frac{\frac{1}{Q}P}{1 + \frac{1}{Q}P + P^2} = \frac{\frac{1}{10}P}{1 + \frac{1}{10}P + P^2}$$

paraméterek: $d_0 = 0$ $d_1 = 0.1$ $d_2 = 0$
 $c_0 = 1$ $c_1 = 0.1$ $c_2 = 1$

digitális szűrő méretezése:

$$\Omega_m = f_m/f_0 = 64 \quad l = \operatorname{ctg} \frac{\pi}{\Omega_m} = 20.35$$

a digitális szűrő paraméterei:

$$C_0 = 0.9902 \quad C_1 = -1.9807 \quad (C_2=1)$$

$$D_0 = -0.0049 \quad D_1 = 0 \quad D_2 = 0.0049$$

Programozási tudnivalók I.

1. A processzor csak egész számokkal dolgozik, ezért a paraméterek egész számú többszörösét használjuk

$$\text{pl.: } C0=0.6479=42460/2^{16}$$

2. A nagyon kis értékű paraméterek kis bemenőjelek esetén alulcsordulást okozhatnak, ezért az átmeneti tárokat hosszabbra, 16 helyett 32 bitesre célszerű választani.

$$\text{pl.: } D2=0.0049$$

3. Az egynél nagyobb abszolút értékű paraméterek aritmetikai túlcsordulást (előjel váltást) okozhatnak,

$$\text{pl.: } C1=-1.98 \text{ esetén}$$

ami a szűrő tartós belengéséhez vezethet.

Programozási tudnivalók II.

4. Időkritikus alkalmazások esetén kihasználjuk a digitális szűrő paramétereinek tulajdonságait:

- Aluláteresztő szűrő esetén

$$D0=D2 \text{ és } D1=2*D0$$

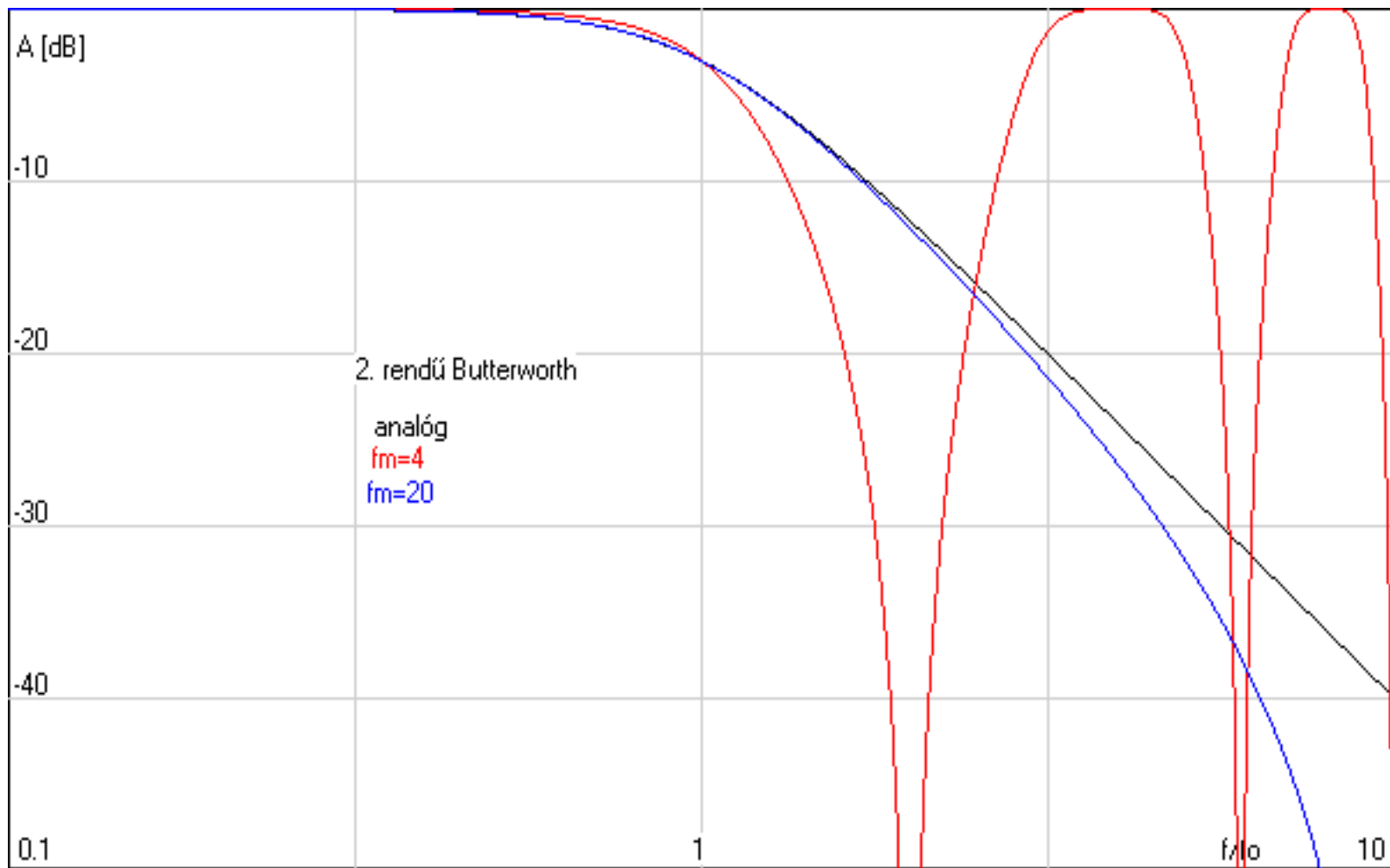
- Sávszűrő esetén

$$D0=-D2 \text{ és } D1=0$$

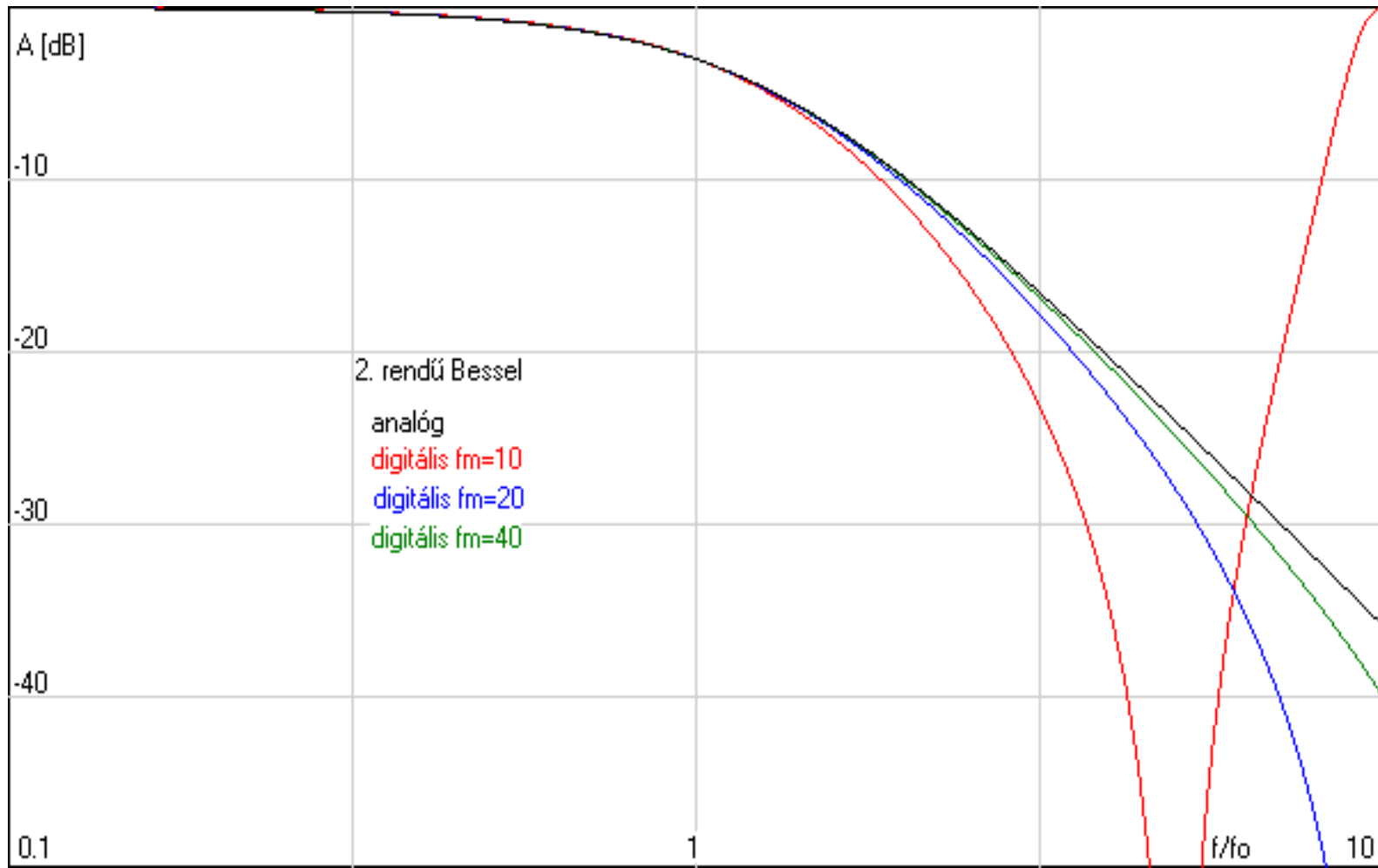
- A könyökponti frekvencia kismértékű változtatásával egyes paraméterek kerek értéket vehetnek fel, ezáltal a szorzásokat gyorsabban lehet elvégezni.

Ezzel a szűrő program futási ideje közel a felére csökkenthető.

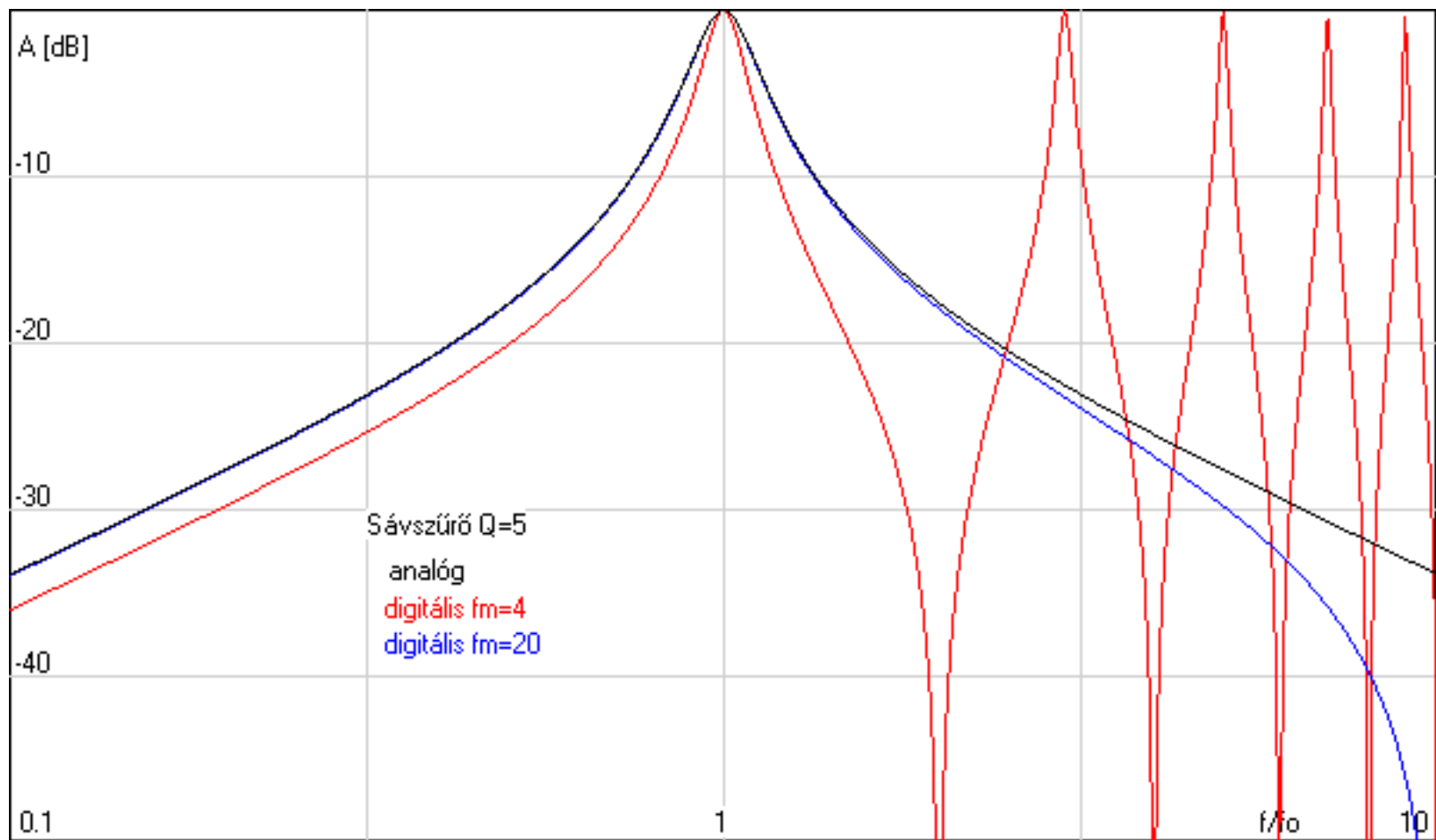
2. rendű Butterworth aluláteresztő szűrő



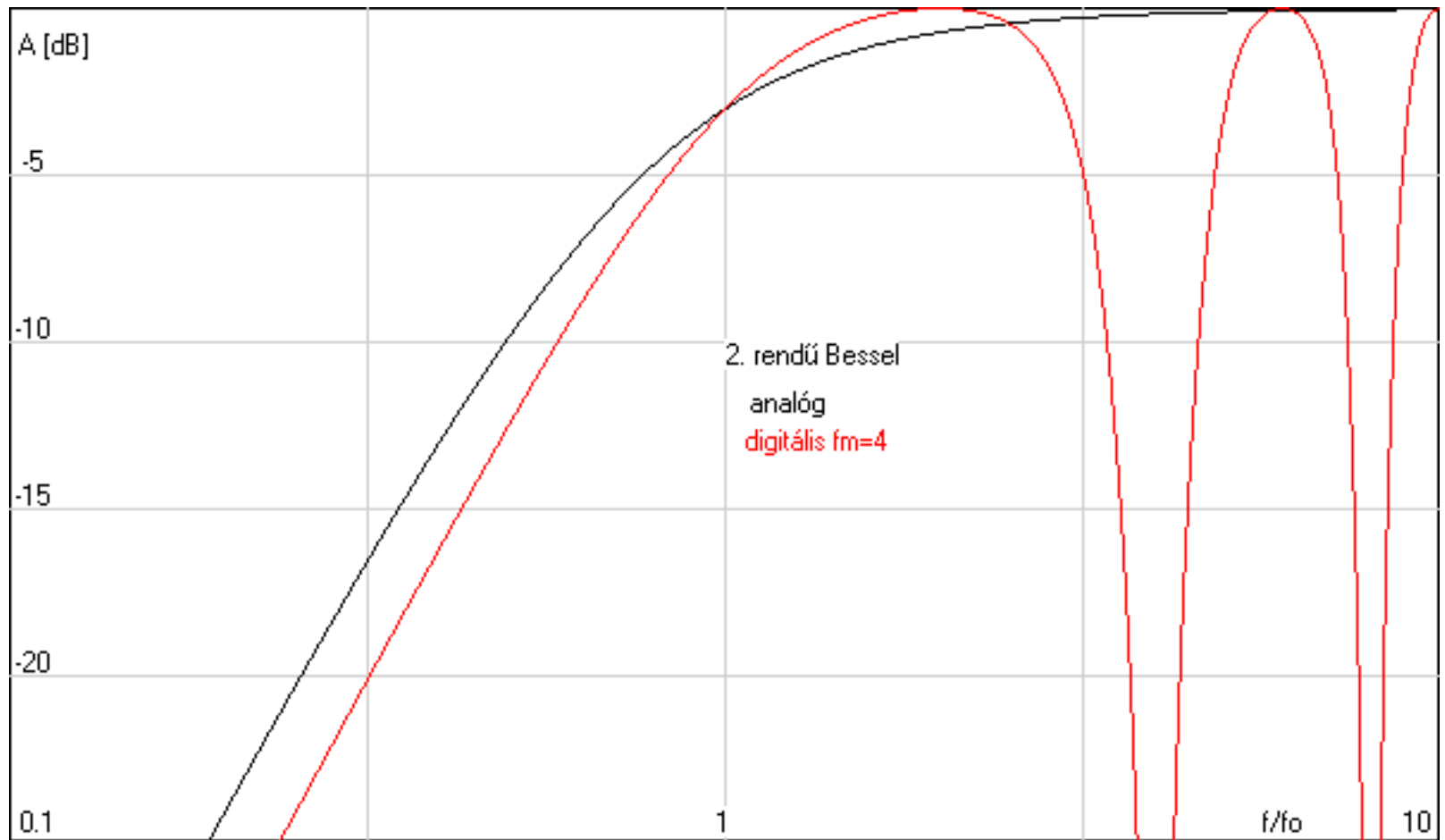
2. rendű Bessel szűrők különböző mintavételi frekvenciával



Sávszűrő különböző mintavételi frekvenciával



2. rendű felül áteresztő szűrő karakterisztika



IIR és FIR szűrők tulajdonságai

IIR	FIR
végtelen hosszú impulzus válasz	véges hosszú impulzus válasz
nagyon hatékony	kevésbé hatékony
van megfelelő analóg szűrő	nincs megfelelő analóg szűrő
visszacsatolást tartalmaz	visszacsatolást nem tartalmaz
instabil lehet	mindig stabil
nem lineáris fáziskarakterisztika	lineáris fáziskarakterisztika
kevésbé számításigényes	nagyon számításigényes
analóg szűrőnek megfelelő karakterisztika lehetséges	szinte tetszőleges karakterisztika kialakítható

IIR szűrő dsPIC processzorra 1. forma

```
x:      .space 6           ;x-2,x-1,x
y:      .space 6           ;y-2,y-1,y
coeffs: .word  C0,D0,C1,D1,D2
```

```
      ; input W8->x   W9->y   W10->coeffs
IIR2: mov    [W9++],W5
      mov    [W10++],W6
      mpy   W5*W6,A,[W8]+=2,W4,[W10]+=2,W6
      mac   W4*W6,A,[W9]+=2,W5,[W10]+=2,W6
      mov   W5,[W9-4]
      mac   W5*W6,A,[W8]+=2,W4,[W10]+=2,W6
      mov   W4,[W8-4]
      mac   W4*W6,A,[W8]+=2,W4,[W10]+=2,W6
      mac   W4*W6,A,[W10]+=2,W6
      sftac A,#-2
      mov   W4,[W8-4]
      mov   ACCAH,W5
      mov   W5,[W9-2]
      return
```

IIR filter C fordítóval 2. forma

```
static int C0=0, C1=1, D0=2, D1=3, D2=4, T0=0, T1=1;  
typedef long filter[2];  
typedef long coeffs[5];
```

```
int IIRfilter(int x, filter filt, coeffs coef)  
{ long yy;  
  yy=(filt[T1]+coef[D2]*x)/0x10000;  
  filt[T1]=filt[T0]+(coef[D1]*x+coef[C1]*yy);  
  filt[T0]=(coef[D0]*x+coef[C0]*yy);  
  return yy;}
```

A lebegőpontos szűrőt a típusok módosításával hozhatjuk létre

Egy másodrendű IIR szűrő futási ideje különböző processzorok és fordítók esetén:

processzor		PIC18F452			dsPIC30F	
órajel	40 MHz	40 MHz	40 MHz	80 MHz	80 MHz	80 MHz
fordító	asm	C	C	asm2	C	C
típus	int	int	float	int	int	float
T _{cyc}	229	1 460	2 280	23	138	1 600
usec	22.9	146	228	1.15	6.9	80
C/asm		6.4	10.0		6.0	69.6

Effektív érték és teljesítmények mérése

(RMS+PQ mérés)

Effektív érték mérése abszolút érték összegzés módszerével

Az egyenirányított bemenő feszültséget összegezzük egy periódusra, abszolút érték egy periódusra vett átlagértéke.

$$U_e = \frac{1}{T} \int_0^T \text{abs}(U(t)) dt$$

szinuszos jel esetén:

$$U_e = \frac{1}{T} \int_0^T \text{abs}(U_p \sin(\omega t)) dt = \frac{2}{\pi} U_p = \frac{2\sqrt{2}}{\pi} U_{\text{eff}} = 0.9 * U_{\text{eff}}$$

$U_{\text{eff}}/U_e = 1/0,9 = 1,11$ formatényező (ha nem tartalmaz harmonikusokat)

Digitális mintavételezés esetén, ha egy periódusból N mintát veszünk:

$$U_e = \frac{1}{N} \sum_{n=0}^{N-1} \text{abs}(U(n))$$

Digitális megvalósítása egyszerű, de nem a definíció szerinti effektív érték!

A mérés maximális hibája a periódusonkénti mintaszám függvényében:

mintaszám	4	8	16	32	64	128
hiba [%]	8,58	1,98	0,49	0,12	0,03	0,01

A módszer hátránya az, hogy az eredmény a φ kezdeti fázistól függ.₂

Effektív érték mérése négyzet összegzés módszerével

A bemenő feszültség négyzetét összegezzük egy periódus hosszan.

$$U_{\text{eff}} = \sqrt{\frac{1}{T} \int_0^T U(t)^2 dt}$$

Ez az effektív
érték definíciója!

szinuszos jel esetén:

$$U_{\text{eff}} = \sqrt{\frac{1}{T} \int_0^T (U_p \sin(\omega t))^2 dt} = \frac{1}{\sqrt{2}} U_p$$

Digitális mintavételezés esetén, ha egy periódusból N mintát veszünk:

$$U_{\text{eff}} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} U(n)^2}$$

Az eredmény nem függ a kezdeti fázistól.

A mérés pontosságát befolyásoló tényezők:

- A mintavételi frekvencia értéke és pontossága
- Az AD konverter felbontása, bitszáma
- A periódusonkénti minták száma

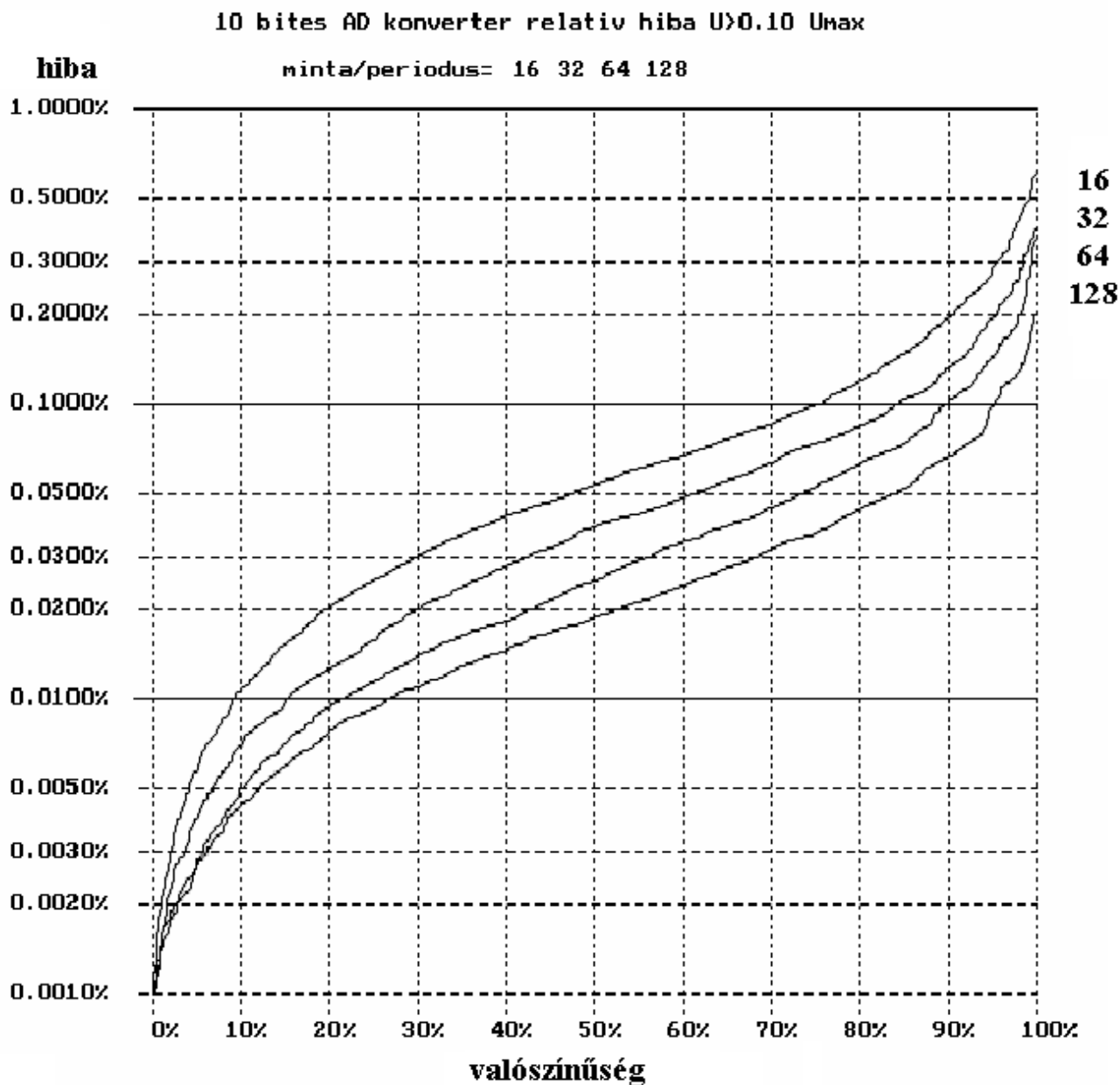
A mérési hibát kétféleképpen lehet meghatározni

- A mérési hiba a maximálisan mérhető érték százalékában (ez az általános, relatív hibának szokás nevezni)
- A mérési hiba a mért érték százalékában

A mérés eredménye a mintavételi frekvencia pontossága és a mintaszám függvényében.

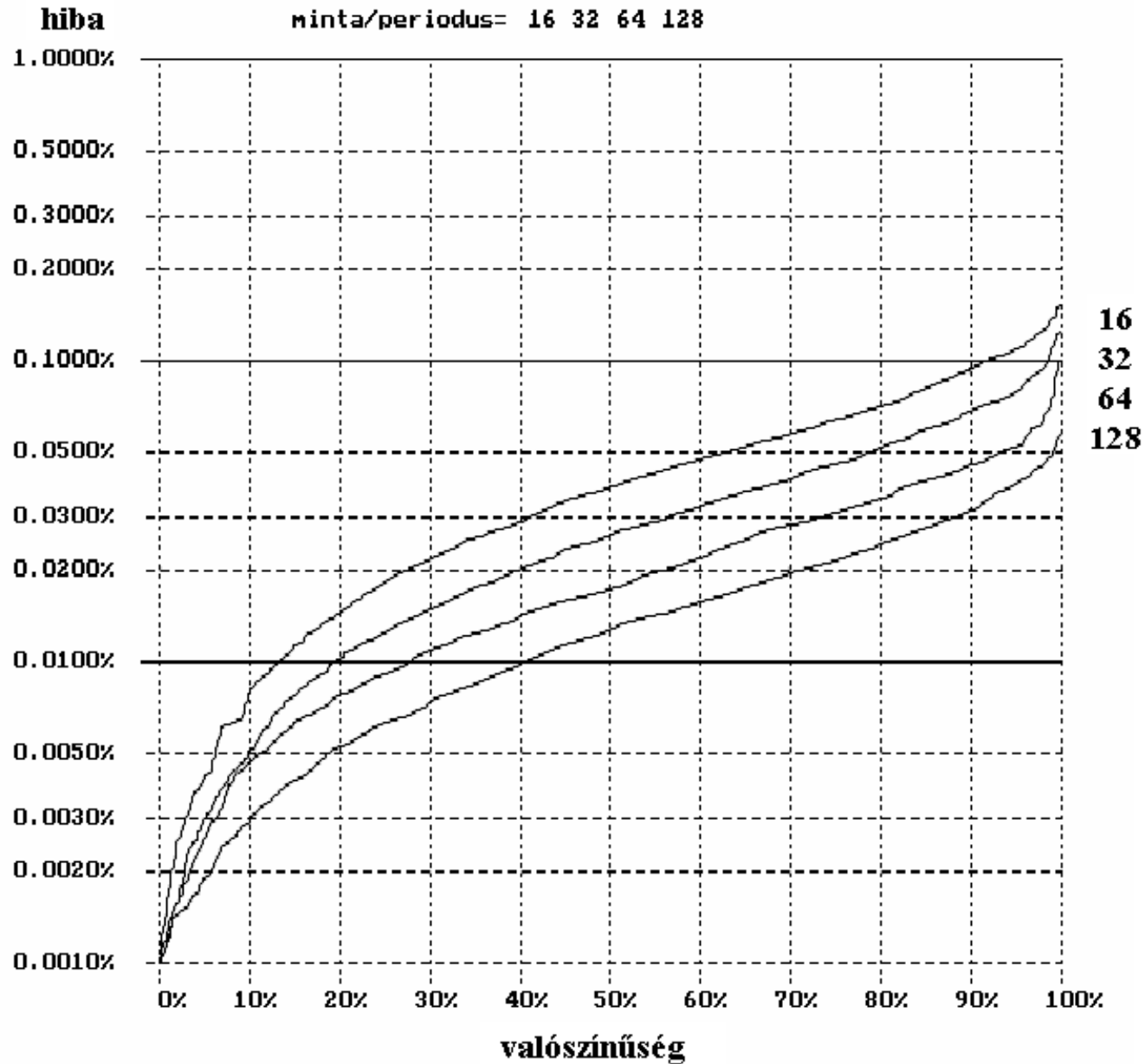
mintaszám:	8	32
49.5 Hz	0.046%	0.0040%
49.6 Hz	0.036%	0.0030%
49.7 Hz	0.026%	0.0020%
49.8 Hz	0.017%	0.0010%
49.9 Hz	0.008%	0.0010%
50.0 Hz	0.000%	0.0000%
50.1 Hz	0.008%	0.0000%
50.2 Hz	0.015%	0.0010%
50.3 Hz	0.022%	0.0020%
50.4 Hz	0.031%	0.0030%
50.5 Hz	0.040%	0.0040%

A mérési hiba mértékének valószínűsége 10 bites AD konverterrel

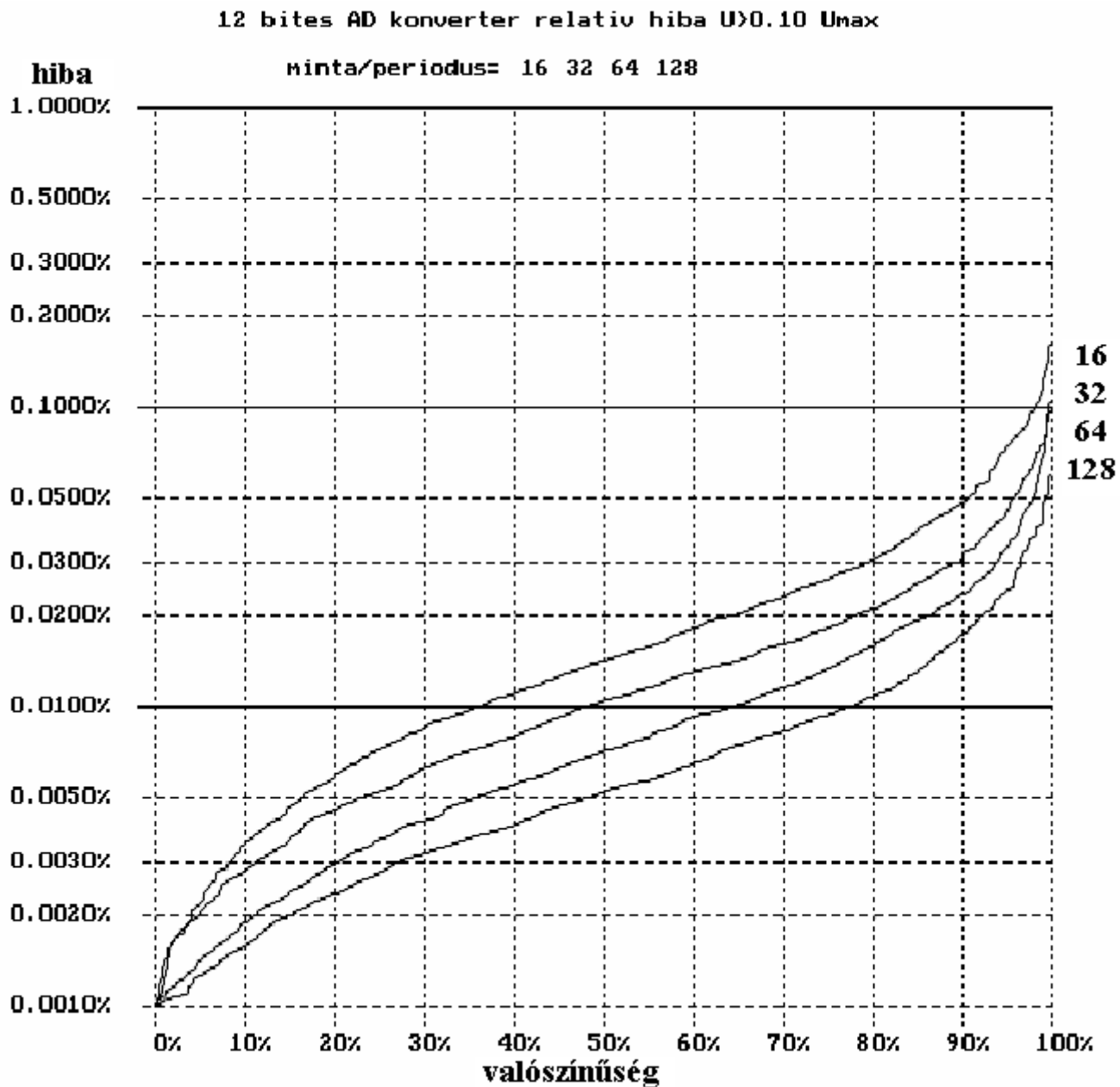


A mérési hiba mértékének valószínűsége 10 bites AD konverterrel

10 bites AD konverter relatív hiba $U > 0.50 U_{max}$



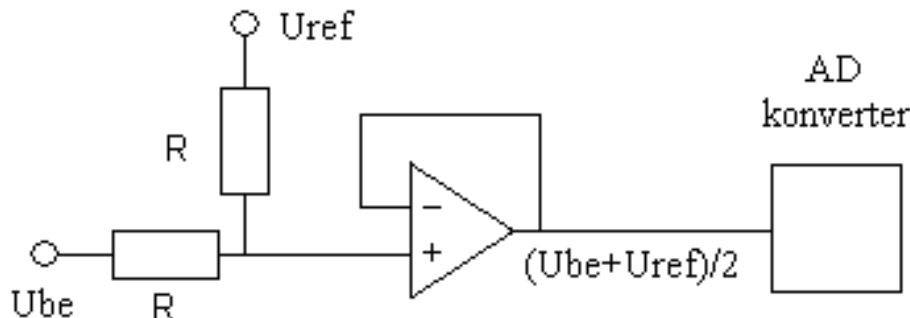
A mérési hiba mértékének valószínűsége 12 bites AD konverterrel



Váltakozó feszültség mérése monopoláris AD konverterrel

A bemenő feszültség egyen összetevőt nem tartalmaz és a mérés ideje az alapharmonikus periódusidő egész számú többszöröse.

A bemenő feszültséghez egyenfeszültséget adunk az alábbi módon:



Az eredő bemenő feszültség $U_{ad} = (U_{be} + U_{ref})/2$

A./ A bemenő feszültség egyen összetevőjét felül áteresztő szűrővel távolítjuk el még az effektívérték algoritmus használata előtt.

B./1. Mérjük az AD konverter bemenő feszültségének effektív értékét:

$$U_{ad\ eff} = \frac{1}{2} \sqrt{\frac{1}{N} \sum_{n=1}^N (U_{be} + U_{ref})^2} = \frac{1}{2} \sqrt{\frac{1}{N} \left(\sum_{n=1}^N U_{be}^2 + \sum_{n=1}^N 2 * (U_{be} * U_{ref}) + \sum_{n=1}^N U_{ref}^2 \right)}$$

Mivel az U_{be} feszültség nem tartalmaz egyen összetevőt:

$$\sum_{n=1}^N 2 * (U_{be} * U_{ref}) = 0$$

De mit tegyünk a harmadik taggal?

AD konverterrel külön mérjük a referencia feszültség értékét és ebből minden mintavételi pontban meghatározzuk a referencia feszültség négyzetét, majd kivonjuk a fenti kifejezésből, és így már a bemenő feszültség effektívértékét kapjuk.

Előnye, hogy referencia feszültség pontossága nem befolyásolja az eredmény pontosságát. Csak rövididejű stabilitás igényünk.

Hátránya, hogy külön AD bemenetet igényel, további időre van szükség a méréséhez, és számítási igény is fellép.

Van ennél jobb megoldás.

Mivel az U_{be} feszültség nem tartalmaz egyen összetevőt:

$$\sum_{n=1}^N 2 * (U_{be} * U_{ref}) = 0 \quad \text{ezért} \quad U_{ad\ eff} = \frac{1}{2} \sqrt{\frac{1}{N} \left(\sum_{n=1}^N U_{be}^2 + \sum_{n=1}^N U_{ref}^2 \right)}$$

A mérési eredményből **számoljuk** a bemenő feszültség egyen összetevőjét:

$$U_{ad\ dc} = \frac{1}{2N} \sum_{n=1}^N (U_{be} + U_{ref}) = \frac{1}{2N} \sum_{n=1}^N U_{ref} \left(= \frac{1}{2} U_{ref} \right) \text{ innen}$$

$$\frac{1}{N} \sum_{n=1}^N U_{ref}^2 = 4U_{ad\ dc}^2$$

Ezeket behelyettesítve a bemenő feszültség effektív értéke számítható:

$$U_{ad\ eff} = \frac{1}{2} \sqrt{U_{be\ eff}^2 + 4U_{ad\ dc}^2} \quad \text{innen}$$

$$U_{be\ eff} = 2\sqrt{U_{ad\ eff}^2 - U_{ad\ dc}^2}$$

A négyzetösszegző módszer mérés algoritmus:

1. Összegző tár és a mintaszámláló törlése
2. Mintavétel
3. A minta négyzetének hozzáadása az összegző tárhoz
4. Mintaszámláló növelése
5. Ismétlés a 2. lépéstől, amíg elérjük a szükséges mintaszámot
6. Az összegző tár osztása a mintaszámmal
7. Négyzetgyök vonás az összegző tárból

A mintaszám célszerűen 2 egész számú hatványa

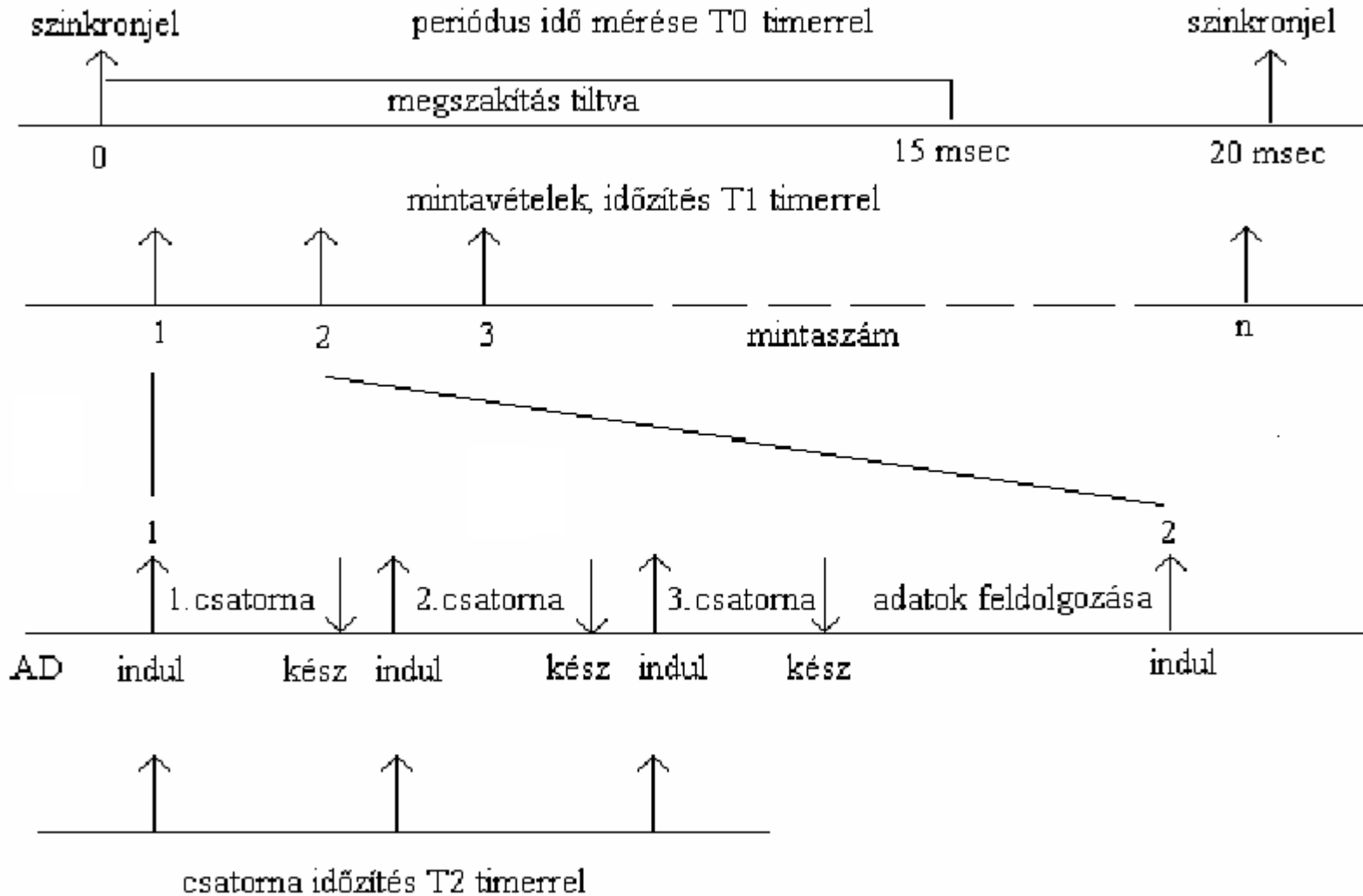
A szükséges mintatár hossz:

$$\log_2(\text{mintaszám}) + 2 * (\text{ADbitszám}-1)$$

pl. 64 minta, 12 bites AD konverter esetén:

$$\text{bitszám} = 6 + 2 * 11 = 28 \text{ bit; szükséges hossz } 4 \text{ byte}$$

Mintavételek időzítése 3 csatornás mintavétel esetén egy periódus alatt n minta



Teljesítmény fogalma

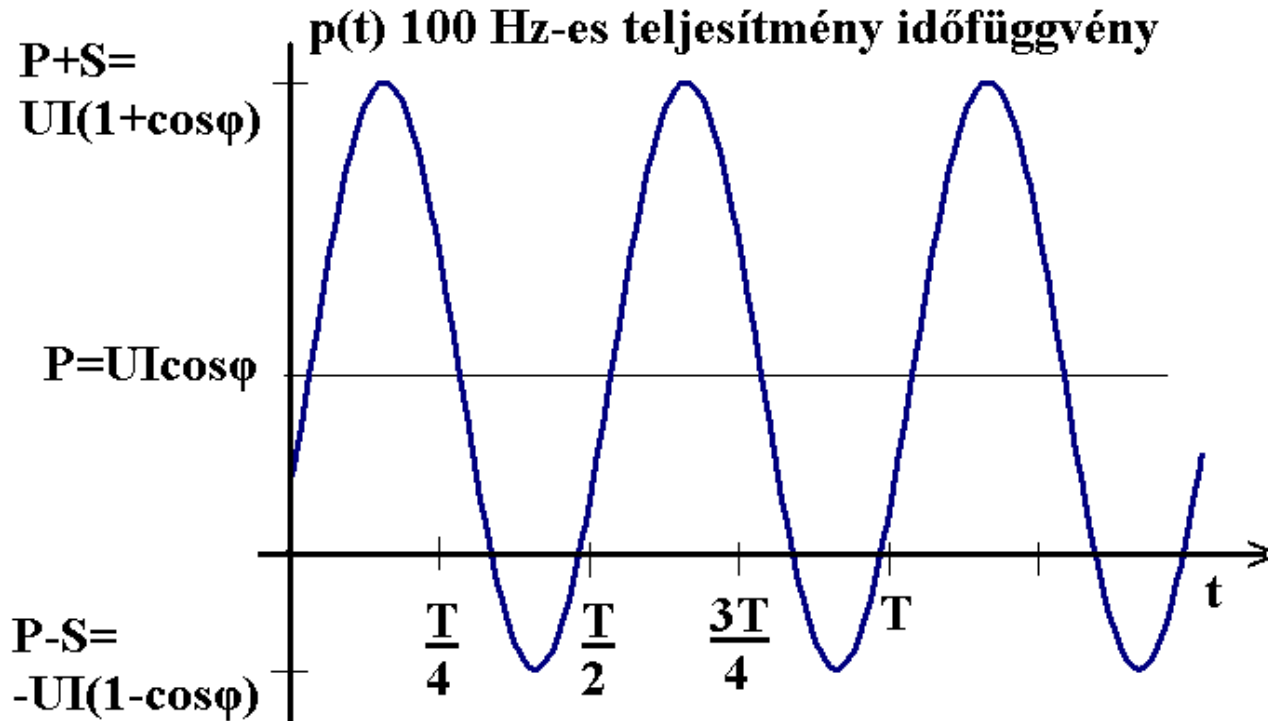
$$u(t) = \sqrt{2}U \cos \omega t, \quad i(t) = \sqrt{2}I \cos(\omega t + \varphi)$$

$$p(t) = u(t)i(t) = 2UI \cos \omega t \cos(\omega t + \varphi)$$

$$p(t) = UI \cos \varphi + UI \cos(2\omega t + \varphi)$$

$$p(t) = UI \cos \varphi + UI(\cos 2\omega t \cos \varphi - \sin 2\omega t \sin \varphi)$$

$$p(t) = P(1 + \cos 2\omega t) + Q \sin 2\omega t, \quad \text{ahol } P = UI \cos \varphi, \quad Q = -UI \sin \varphi$$



A pillanatérték időfüggvényének átlaga a wattos, a változás értéke a meddő teljesítmény.

Wattos teljesítmény mérése

A wattos (hatásos) teljesítmény értéke egy periódusra (T):

$$P = \frac{1}{T} \int_0^T U(t) * I(t) dt$$

illetve szinuszos időbeli változás esetén:

$$P = U * I * \cos \varphi$$

Mintavételezett jelek esetén (N minta/periódus):

$$P = \frac{1}{N} \sum_{i=1}^N U(i) * I(i)$$

Mintavételezésből eredő hiba mértéke

Ha a feszültség és az áram mintavételezése nem azonos időpontban történik, akkor ez fázistolást és ezáltal a teljesítménymérésben hibát idéz elő. Legyen a két mintavétel között 55 μ sec eltérés, ez 1 fok fázishibának felel meg, ekkor a mérés hibáját az alábbi táblázat adja meg ($S=1$):

$\cos\varphi$	0,98	0,94	0,87	0,64	0,50	0,34	0,17
dP	0,32%	0,61%	0,89%	1,35%	1,52%	1,65%	1,72%
dP/P	0,32%	0,65%	1,02%	2,10%	3,04%	4,81%	9,91%

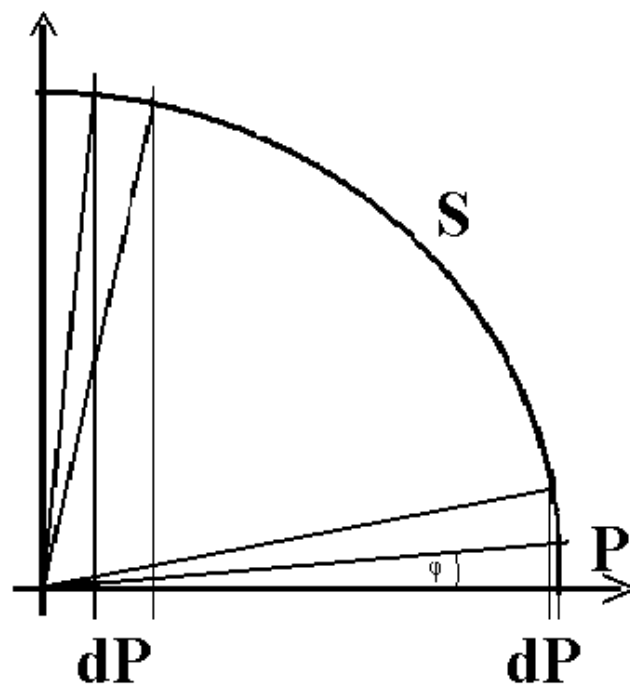
Miért van az, hogy kis $\cos\varphi$ értékeknél a hiba mértéke sokkal nagyobb?

A $\cos\varphi$ függés mértékének magyarázata

A koszinusz függvény meredeksége miatt.

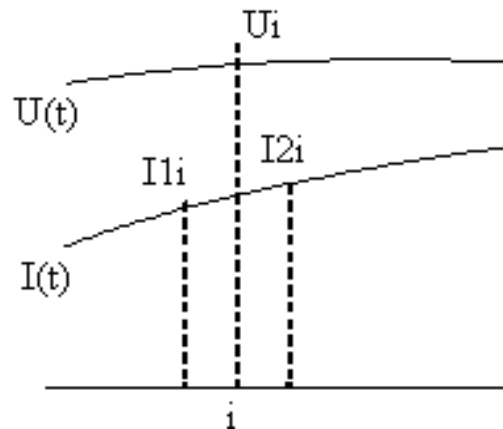
φ	1	10	20	30	50	60	70	80
$\cos\varphi$	1,00	0,98	0,94	0,87	0,64	0,50	0,34	0,17
dP	0,05%	0,32%	0,61%	0,89%	1,35%	1,52%	1,65%	1,72%
dP/P	0,05%	0,32%	0,65%	1,02%	2,10%	3,04%	4,81%	9,91%

Ugyan ezt nézzük meg az $S=1$ sugarú körön



A mintavételezésből eredő hiba kiküszöbölése

- Sample-Hold áramkörök alkalmazásával
- Analóg fázistoló áramkörrel az AD konverter előtt
- Digitális fázisforgatással a mintavételezés után
- Az egyik csatorna kétszeres mintavételével:



Az i . minta az alábbiak szerint számítható:

$$P_i = U_i * \frac{I_{1i} + I_{2i}}{2}$$

Meddő teljesítmény mérése

A meddő teljesítmény értéke egy periódusra (T) illetve szinuszos időbeli változás esetén:

$$Q = \frac{1}{T} \int_0^T U(t) * I(t - T/4) dt \qquad Q = U * I * \sin \varphi$$

Mintavételezett jelek esetén
(N minta/periódus):

$$Q = \frac{1}{N} \sum_{i=1}^N U(i) * I(i - N/4)$$

Mintavételezésből eredő hiba mértéke:

φ	1	10	20	30	50	60	70	80
$\cos\varphi$	1,00	0,98	0,94	0,87	0,64	0,50	0,34	0,17
$\sin\varphi$	0,02	0,17	0,34	0,50	0,77	0,87	0,94	0,98
dQ	1,74%	1,72%	1,63%	1,50%	1,11%	0,86%	0,58%	0,29%
dQ/Q	99,97%	9,88%	4,78%	3,01%	1,45%	0,99%	0,62%	0,29%

A meddőteljesítmény mérésére szolgáló módszerek

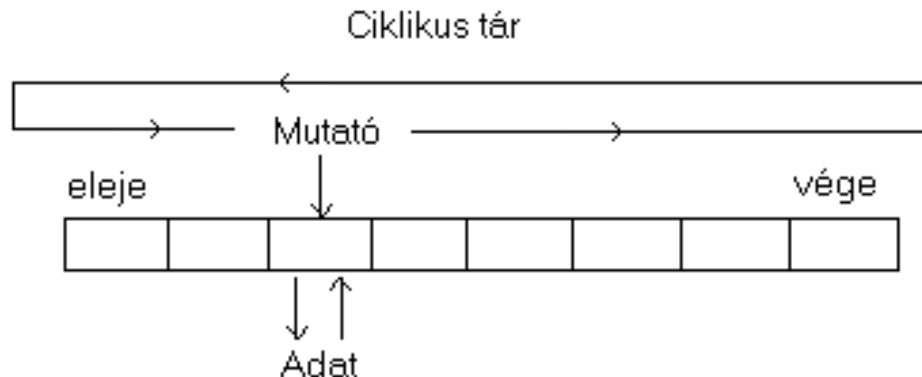
1. Negyed periódussal ($N/4$ mintával) korábbi minták használata az egyik bemenő jel (általában az áram) esetén.

A módszer használatakor $N/4$ mintát tárolni kell. Kétféle tár használata szokásos:

a. FIFO tár - nagyobb mintaszám esetén sok időt vesz igénybe ($N/4$ adat mozgatása szükséges)



b. ciklikus tár - csak egy adat cseréje és a mutató léptetése szükséges. A DSP processzorok beépített gyors ciklikus tár kezelő utasításokkal rendelkeznek.



2. Fázisforgatás digitális szűrő segítségével.

Az alábbi elsőrendű digitális szűrő
90 fokos fázistolást ad egység
erősítés mellett az $\Omega=1$ frekvencián:

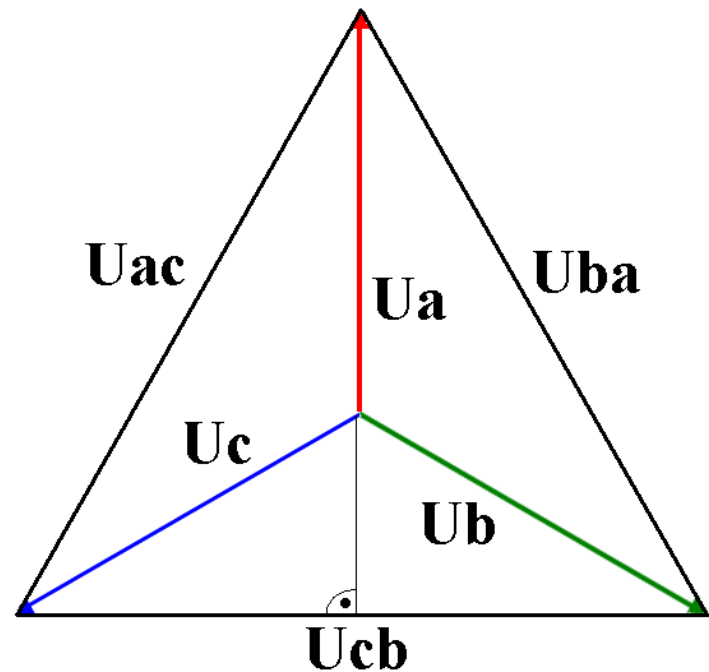
$$A(P) = \frac{1 - P}{1 + P}$$

3. Vonali feszültség használata

Háromfázisú rendszer esetén az
 U_{cb} vonali feszültség vektora
merőleges az U_a feszültségre.

A wattos teljesítmény mérő
algoritmus az U_{cb} és I_a jelek
használatával az a fázis meddő
teljesítményét szolgáltatja.

A módszer csak szimmetrikus
feszültségrendszer esetén
($U_0=U_2=0$) alkalmazható!



4. P és S mérésének segítségével

A feszültség és az áram effektív értékét és a wattos teljesítményt mérjük az ismert módszerekkel, ezekből a meddő teljesítmény számítható:

$$Q = \sqrt{S^2 - P^2} = \sqrt{(U_{eff} * I_{eff})^2 - P^2}$$

A módszer a meddő teljesítmény előjelét nem adja meg!

Ha a feszültség vagy áram jelentős harmonikustartalommal rendelkezik, a módszer nem alkalmas a meddőteljesítmény mérésére.

Néhány esetre a harmonikusok által okozott hiba (U=1, I=1, THD= áram harmonikus tartalma):

$\cos\varphi$	Q	THD	Smért	Qmért
1	0,00	10%	1,0050	0,10
0,7	0,71	10%	1,0050	0,72
1	0,00	30%	1,0440	0,30
0,7	0,71	30%	1,0440	0,77

Alapharmonikus effektív érték és teljesítmények mérése

Az eddig megismert teljesítmény mérési módszerek felharmonikusok jelenlétében nem pontosan az alapharmonikus teljesítményeket szolgáltatják. Az alapharmonikus teljesítmények méréséhez meg kell határozni az áram és a feszültség alapharmonikus összetevőjét mint komplex vektort.

Az $U(t)$ általános időfüggvény amelyből az U_1 alapharmonikus komplex érték egy periódusra történő integrálással határozható meg:

$$U_1 = \frac{1}{T} \int_0^T U(t) e^{-j\omega_1 t} dt$$

Mintavételezett jelek esetén (N minta/periódus esetén) az alábbi számítási módszer alkalmazzuk:

$$U_1 = \frac{1}{N} \sum_{n=0}^{N-1} U(n) \cos\left(n \frac{2\pi}{N}\right) - j \frac{1}{N} \sum_{n=0}^{N-1} U(n) \sin\left(n \frac{2\pi}{N}\right)$$

Ezzel megkapjuk az alapharmonikus feszültségvektor összetevőit és abszolút értékét:

$$U_1 = U_{1x} + j^* U_{1y} \quad |U_1| = \sqrt{U_{1x}^2 + U_{1y}^2}$$

A alapharmonikus áramvektor hasonló módszerrel számítható:

$$I_1 = I_{1x} + j^* I_{1y} \quad |I_1| = \sqrt{I_{1x}^2 + I_{1y}^2}$$

($|U_1|$ és $|I_1|$ csúcsértékek, szinuszos időfüggvény paraméterei)

Az áram és feszültségvektorokból a teljesítmények számíthatók:

$$S = U_1 * I_1^* = (U_{1x} * I_{1x} + U_{1y} * I_{1y}) + j * (U_{1y} * I_{1x} - U_{1x} * I_{1y})$$

Az alapharmonikus teljesítmények:

$$P = U_{1x} * I_{1x} + U_{1y} * I_{1y}$$

$$Q = U_{1y} * I_{1x} - U_{1x} * I_{1y}$$

A számításokhoz szükséges sin és cos függvényeket függvény táblázatokból vesszük. Ezeket célszerűen ciklikus tárukban tároljuk.

Mintavételi frekvencia meghatározása

A mintavételi frekvencia N minta/periódus esetén $f_m = T/N$, ahol T az alapharmonikus periódusidő. Szokásosan N értékét 2 egész számú hatványára választjuk, ez a számítási időt jelentősen megrövidítheti:

- az osztást egyszerű shift művelettel helyettesíthetjük
- a ciklikus tárok szervezése egyszerűbb

A mintavételi frekvencia meghatározása történhet:

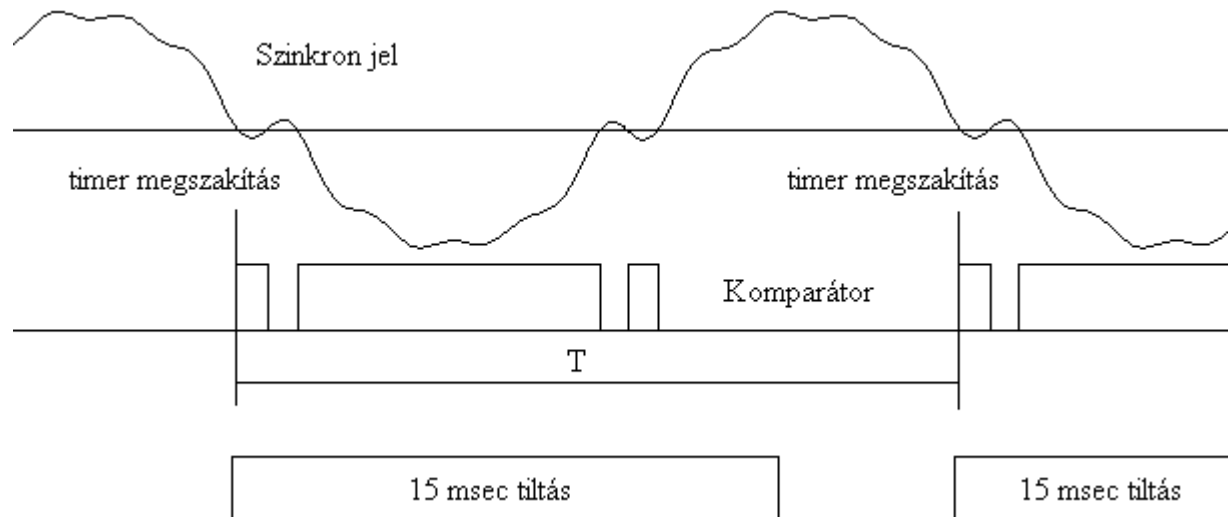
1. Az alapharmonikus frekvenciát állandó 50 Hz-nek tekintjük. A hálózati frekvencia változásakor kisebb mértékű mérési hiba lép fel és a mért periódus idők kezdetének fázishelyzete folyamatosan változik.

2. Mérjük a hálózati frekvenciát. A mérések kezdő fázisát nem szinkronizáljuk, ezek fázishelyzete változik.

3. Mérjük a hálózati frekvenciát és az egy periódus mérésének kezdetét szinkronizáljuk az adott jelhez, ez általában az U_a fázisfeszültség. A periódus idő mérése a szinkron jel negatív nulla átmenetei közötti idő mérésével történik.

A szinkron jel harmonikusokat tartalmazhat, ekkor több nulla átmenet keletkezik egymáshoz közel. A hibás mérés kiküszöbölhető:

- a periódusidő hihetőségének ellenőrzése
- megszakítás tiltása kb. 15 msec ideig



Effektív érték mérő algoritmus PIC18F típusra asm fordítóval

```
getrms:  clr          sumu,4           ;4 byte-s összegző tár törlése
         movlw       40                ;64 minta/periódus
         movff       WREG,minta        ;ciklusváltozó számláló
rlp:     movff       POSTINC0,u+0      ;2 byte-s mutató az adatokra
         movff       POSTINC0,u+1
         sqri2       u2,u              ;négyzetre emelés
         add         sumu,u2,4         ;összegzés
         decf        minta,f,0        ;ciklusvált. csökkentése 0-ig
         bnz         rlp                ;ugrás vissza, ha nincs vége
rept    set         6                    ;osztás 64-gyel
         while      rept                ;6 szor ciklikus eltolással
         srl         sumu,4
rept    set         rept-1
         endw
         sqrt4       Urms,sumu         ;gyökvonás
         return
```

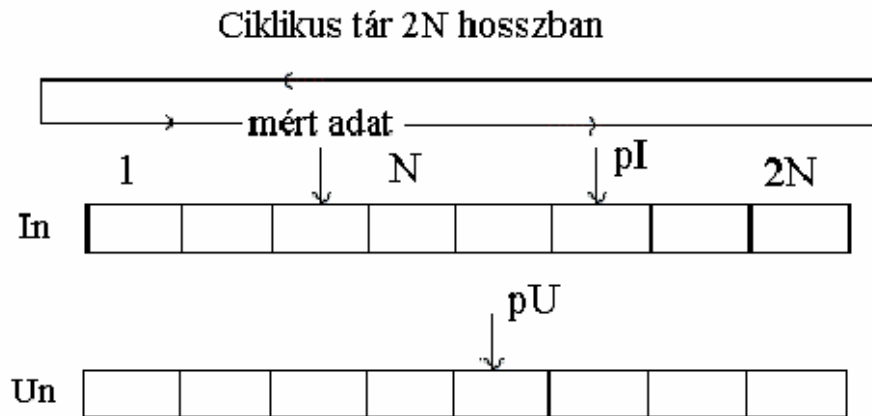
Effektív érték mérő algoritmus C fordítóval

A mért adatok egy 64 minta hosszú tárbán találhatóak

```
typedef int mintak[64];           // 64 minta/periódus
mintak  umintak;                 // mért adatok

unsigned int getrms(int *pu, int mintaszam)
{unsigned long sumu2=0;
  int n;
  for(n=0; n<mintaszam; n++,pu++)           //ciklus szervezés
    sumu2=sumu2+(long)(*pu)*(*pu);
  return sqrt(sumu2/mintaszam);}
}
```


Meddő mérő algoritmus C fordítóval



A méréshez $2N$ hosszú ciklikus tár szükséges, hogy az új adatok tárolása és a feldolgozás egy időben történhessen

```
typedef int mintak[128];    // 64 minta/periódus
mintak umintak, imintak;   // mért U és I minták
```

```
int getPQ(int *pu, int *pi, int *pimax, int mintaszam)
{long sump=0; int n;
for(n=0; n<mintaszam; n++, pu++,pi++){
    if (pi==pimax) pi=pi-2*mintaszam;    //ciklikus tár
    sump=sump+(long)(*pu)*(*pi);}
return sump/mintaszam/0x1000;}
```

Effektív érték és meddő teljesítmény mérő algoritmus dsPIC típusra asm fordítóval

```
getRms:   clr      A                ;összegző tár
          mov     [W8++],W4        ;W8->U mintatár
          repeat #63
          mac     W4*W4,A,[W8] +=2,W4
          sftac   A,#6             ;osztás 64-gyel
          call    sqrtA           ;gyökvonás
          return
```

```
getPQ:    clr      A
          mov     [W8++],W4        ;W8->U mintatár
          mov     [W9++],W5        ;W9->I mintatár
          do      #63,pqloop
          mac     W4*W5,A,[W8] +=2,W4
pqloop:   mov     [W9++],W5
          sftac   A,#6
          mov     ACCAH,W4
          return
```

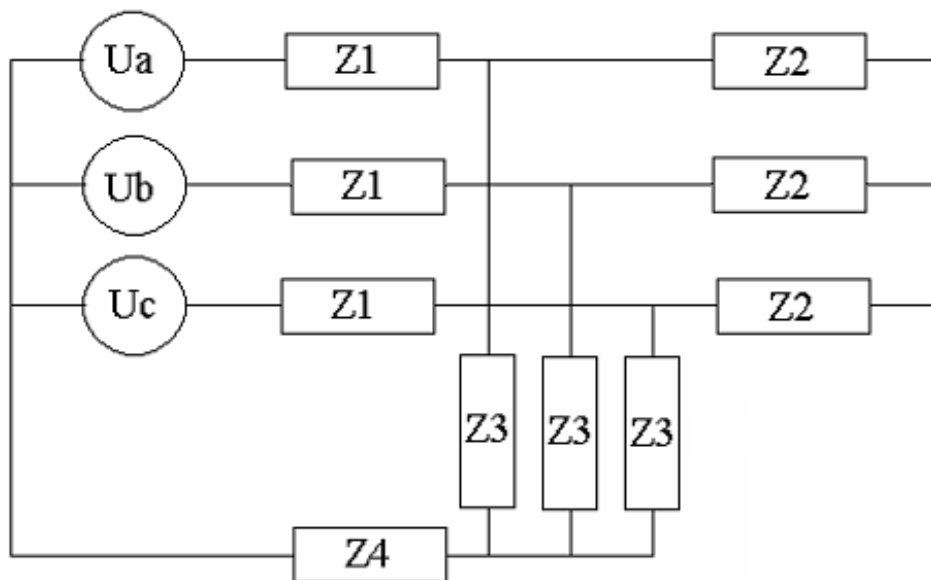
Az algoritmusok futási ideje különböző processzorok és fordítók esetén

processzor		PIC18F452		dsPIC30F	
	órajel	40 MHz	40 MHz	80 MHz	80 MHz
64 minta	fordító	asm	C	asm	C
rms	Tcyc	2 920	19 500	158	2 470
p,q	Tcyc	3 450	14 500	140	2 370
u,i,p,q	Tcyc	12 740	68 000	596	9 680
3 fázis	Tcyc	38 220	204 000	1 788	29 040
3 fázis	msec	3.822	20.4	0.0894	1.452
3 fázis	C/asm		5.3		16.2

Sorrendi mennyiségek mérése

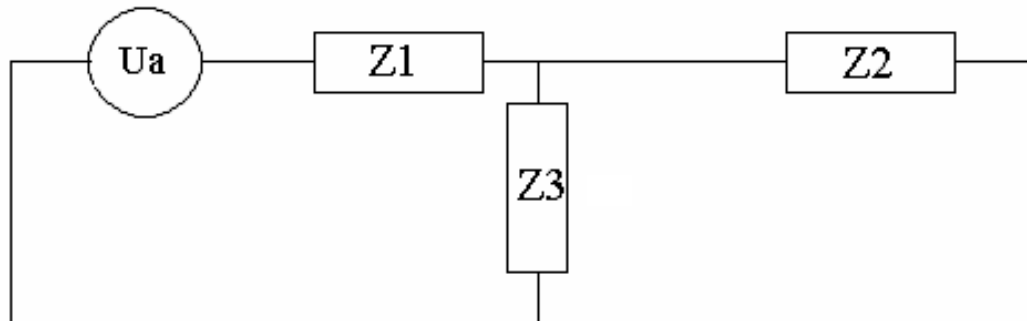
Sorrendi mennyiségek fogalma és alkalmazása

Az alábbi ábrán U_a , U_b és U_c tetszőleges amplitúdójú és fázisú alapharmonikus feszültségek vannak.

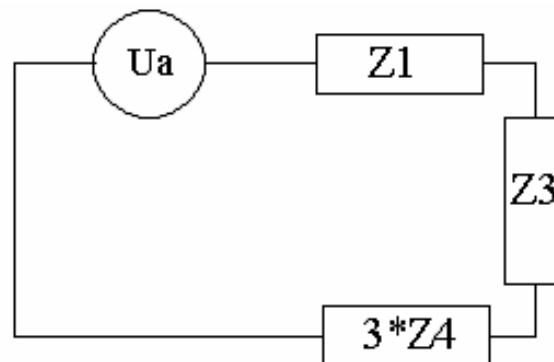


Az áramkör egyszerűsíthető, ha a feszültségek között speciális összefüggések állnak fenn. Ekkor az áramkör egyfázisúvá redukálható.

A./ A három feszültség időfüggvény összege minden pillanatban nulla:

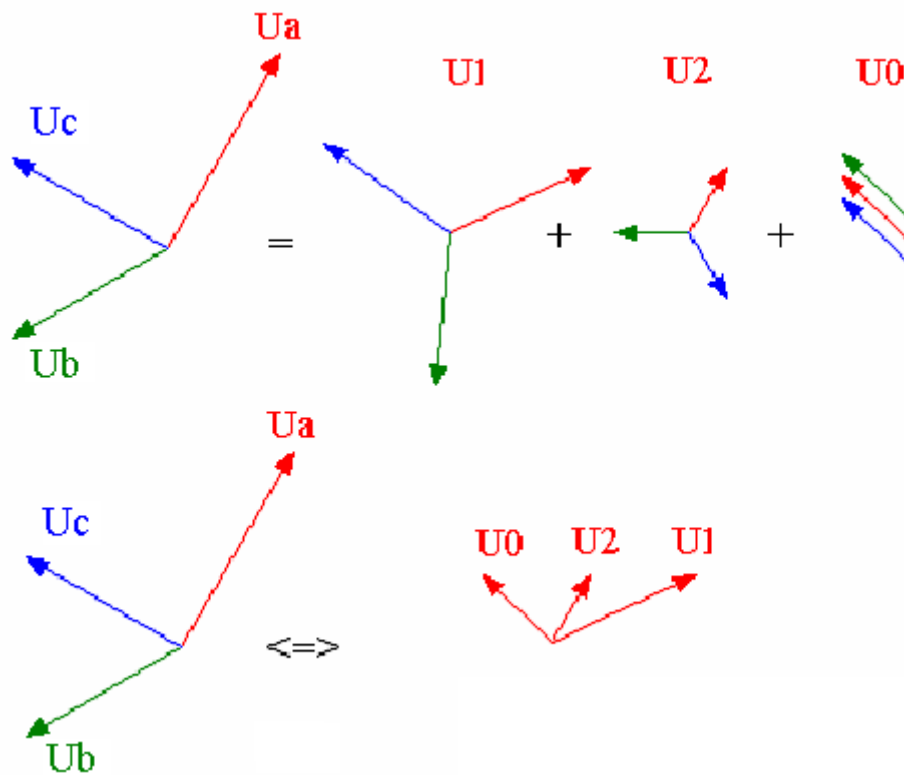


B./ A három időfüggvény azonos:



Ha az áramkör szimmetrikus és lineáris elemekből áll, akkor minden esetben felbontható olyan összetevőkre, hogy az előző két eset egyikét lehessen alkalmazni.

Az alábbi példában az U_a , U_b , U_c vektorokat 3 db háromfázisú vektorrendszerre bonthatjuk. Az U_a vektor összetevőit elnevezzük U_0 , U_1 , U_2 vektoroknak.



Az U_a , U_b , U_c és U_0 , U_1 , U_2 vektorok között kölcsönösen egyértelmű összefüggés van:

$$\begin{vmatrix} U_0 \\ U_1 \\ U_2 \end{vmatrix} = \frac{1}{3} \begin{vmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{vmatrix} * \begin{vmatrix} U_a \\ U_b \\ U_c \end{vmatrix} \quad \text{és} \quad \begin{vmatrix} U_a \\ U_b \\ U_c \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{vmatrix} * \begin{vmatrix} U_0 \\ U_1 \\ U_2 \end{vmatrix}$$

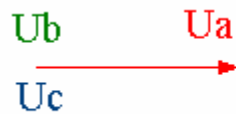
Ahol „a” 120°-os komplex (forgató) egységvektor

$$a = e^{j120^\circ} = e^{j2/3\pi} = \cos 120^\circ + j \sin 120^\circ = -0.5 + j\sqrt{3}/2 = -0,5 + j0,866$$

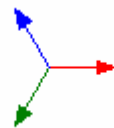
$$a^2 = e^{j240^\circ} = e^{-j2/3\pi} = \cos 240^\circ + j \sin 240^\circ = -0.5 - j\sqrt{3}/2 = -0,5 - j0,866$$

Példák a felbontásra:

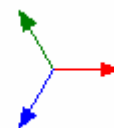
1. $U_b=U_c=0$



U1



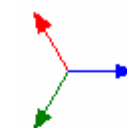
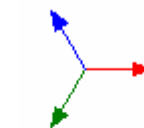
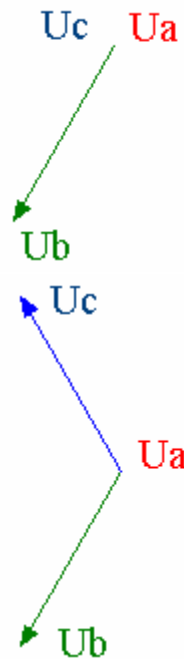
U2



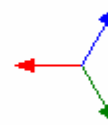
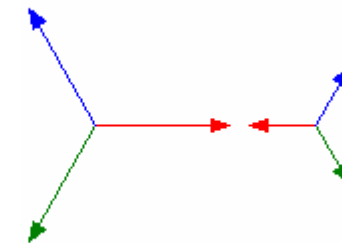
U0



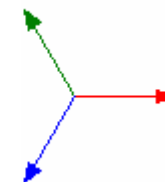
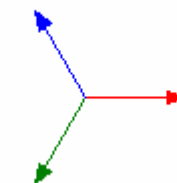
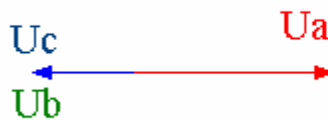
2. $U_a=U_c=0$



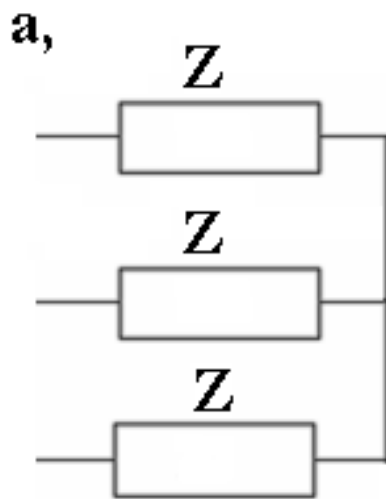
3. $U_a=0$



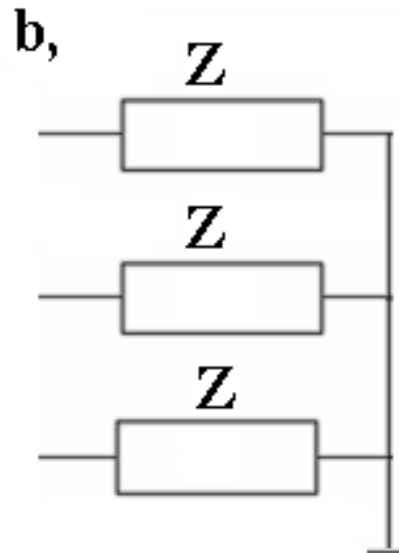
4. $U_b=U_c=-U_a/2$



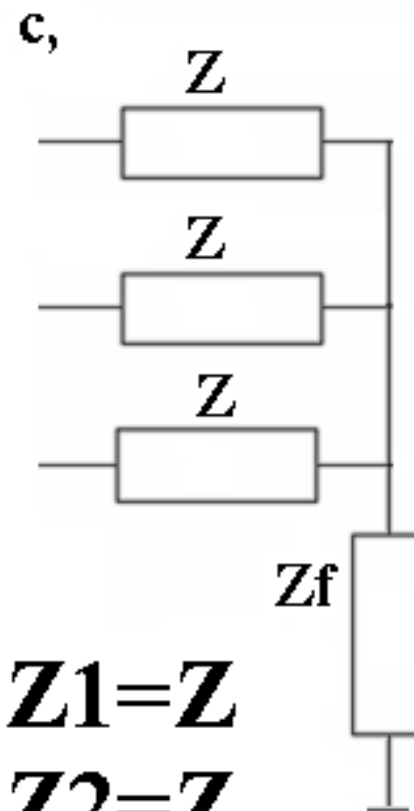
Hálózati elemek sorrendi impedanciái I.



$$\begin{aligned} Z_1 &= Z \\ Z_2 &= Z \\ Z_0 &= \infty \end{aligned}$$

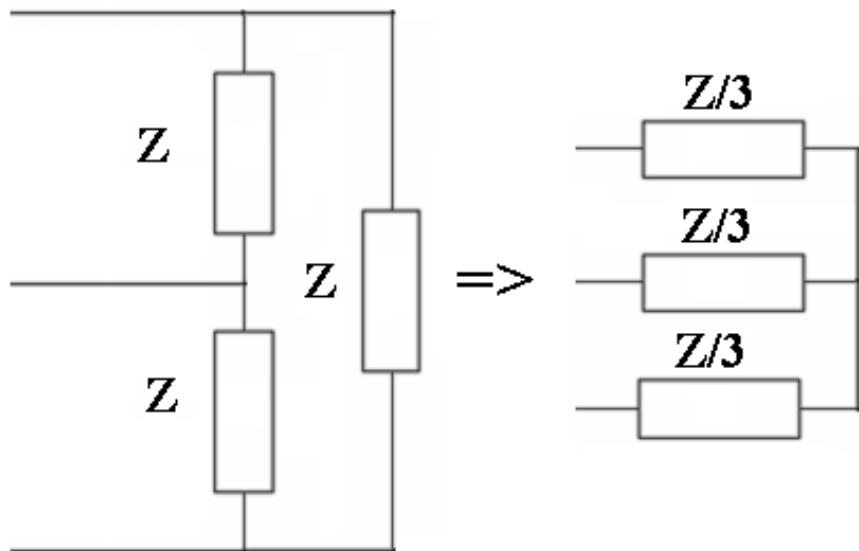


$$\begin{aligned} Z_1 &= Z \\ Z_2 &= Z \\ Z_0 &= Z \end{aligned}$$



$$\begin{aligned} Z_1 &= Z \\ Z_2 &= Z \\ Z_0 &= Z + 3Zf \end{aligned}$$

Hálózati elemek sorrendi impedanciái II.



$$\frac{U_v^2}{Z_\Delta} = \frac{U_f^2}{Z_Y}$$

$$\frac{(\sqrt{3}U_f)^2}{Z_\Delta} = \frac{U_v^2}{Z_\Delta} = \frac{U_f^2}{Z_Y}$$

$$Z_Y = \frac{Z_\Delta}{3}$$

A delta kapcsolást átalakítjuk csillaggá, és az értékeket meghatározzuk. Az átalakítást legegyszerűbben a teljesítmények azonossága alapján tehetjük meg.

$$Z_1 = Z_2 = Z / 3$$

$$Z_0 = \infty,$$

Transzformátor sorrendi impedanciája I.

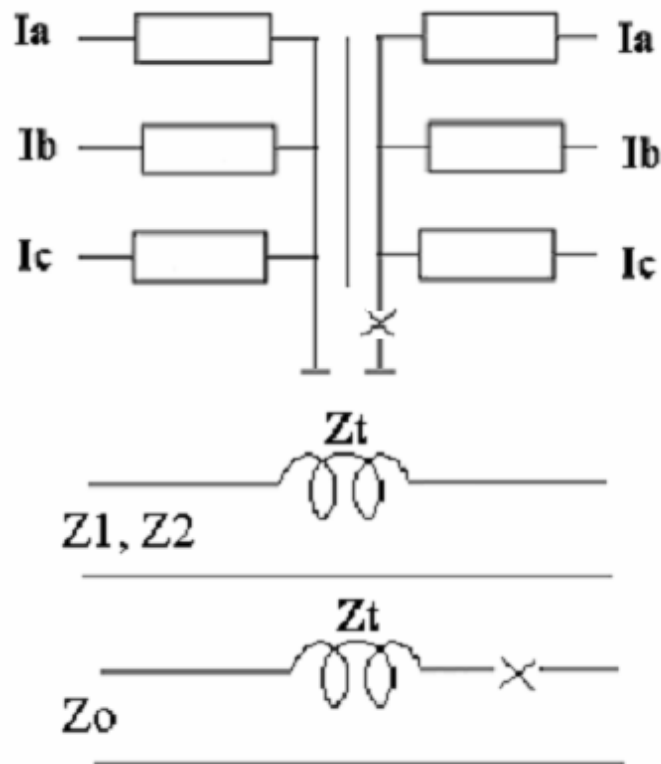
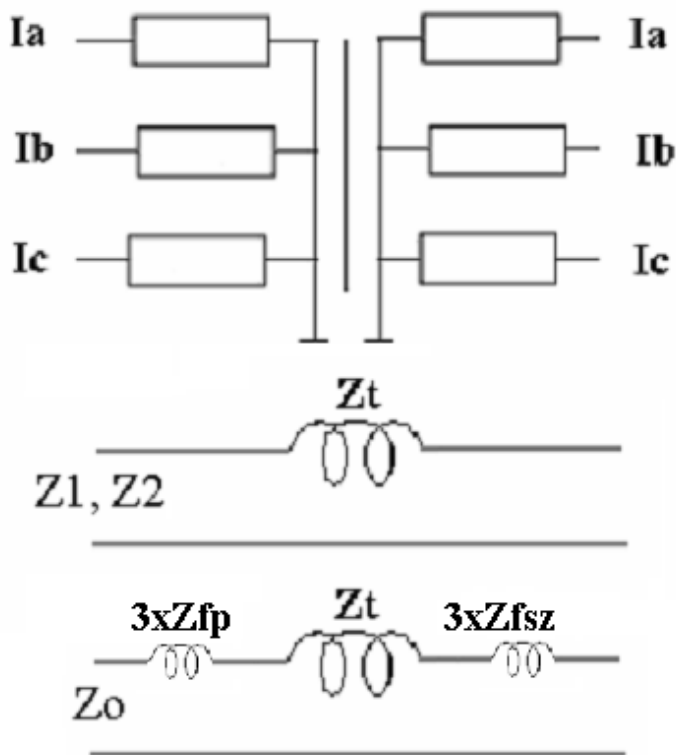
$$\varepsilon, S_{3f}, U_p, U_{sz}$$

$$Z_t = \varepsilon * \frac{U_{nv}^2}{S_{3f}}$$

U_{nv} névleges vonali feszültség

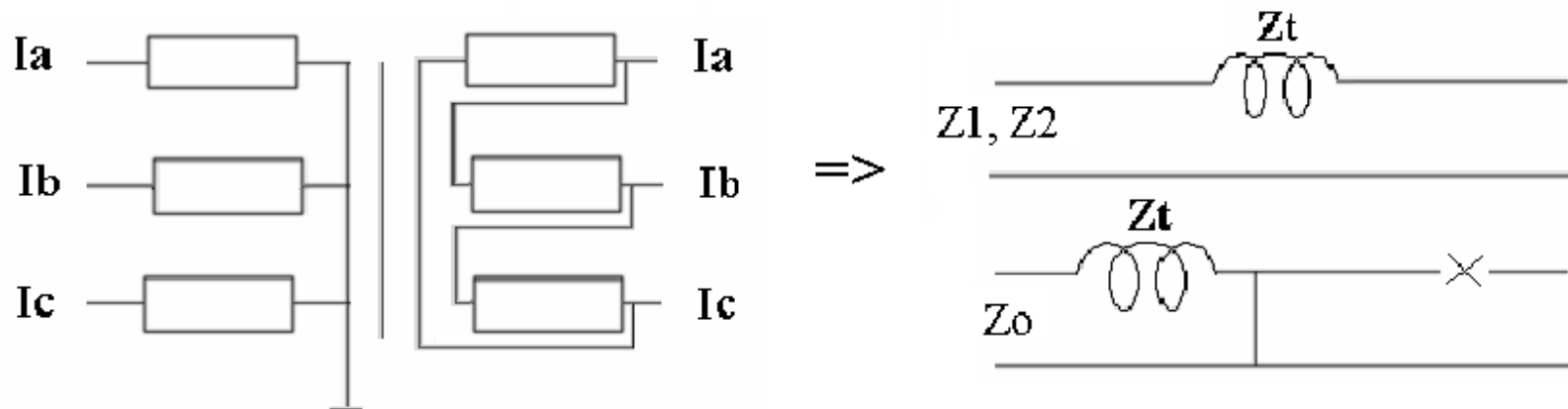
S_{3f} háromfázisú névleges teljesítmény

ε transzformátorjellemző

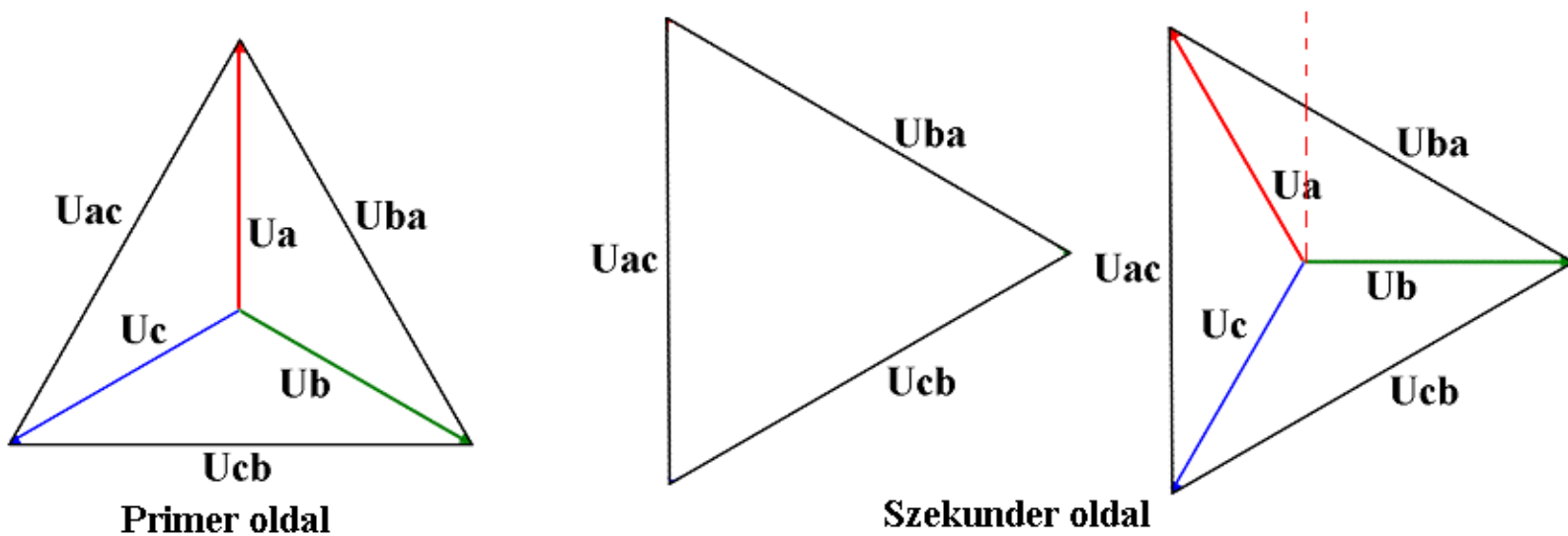


Z_f a csillagpontba tett földági impedancia ha van

Transzformátor sorrendi impedanciája II.



Yd11 transzformátor kapcsolás vektor ábrája



Sorrendi összetevők mérésének problémái

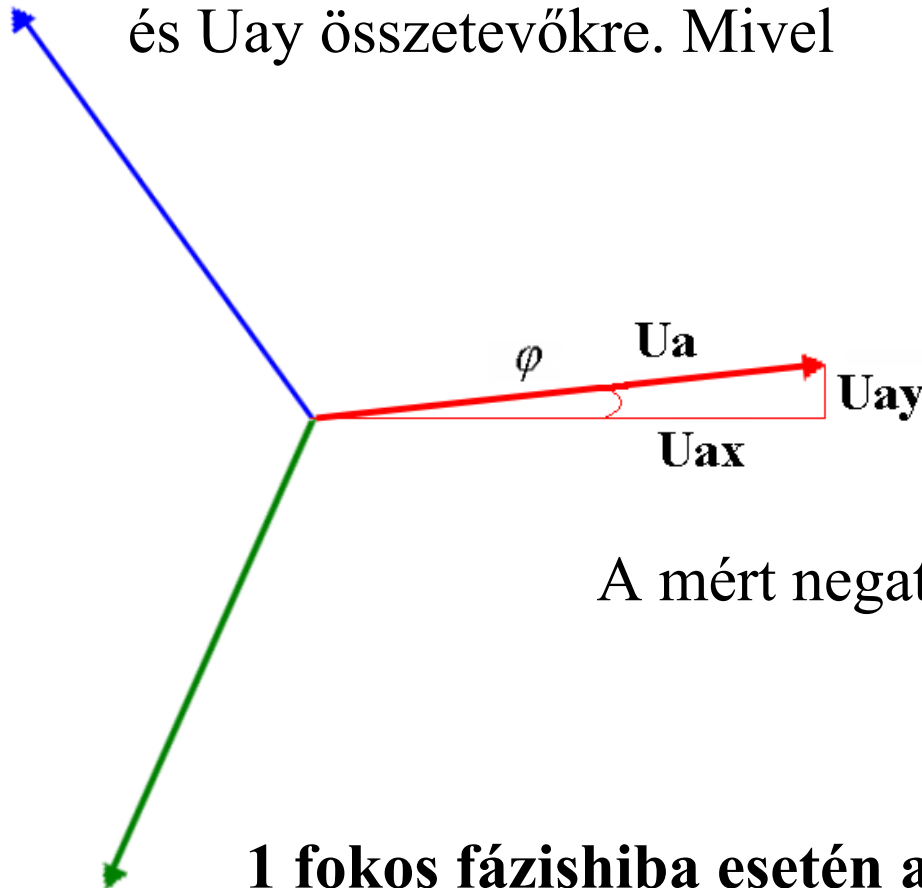
1. A negatív és zérus sorrendű összetevők sokszor kis értékűek a pozitív sorrendű értékhez képest, a negatív sorrendű összetevő értéke általában 1% körül van. Ennek pontos (Pl. 1%-os) mérése analóg áramköröktől nagyon nagy pontosságot, digitális mérés esetén nagyobb felbontást igényel.
2. A zérus sorrendű összetevő mérése egyszerűen az időfüggvények összeadásával valósítható meg. (Pl.: összegző mérőváltó)
3. A sorrendi mennyiségek definíció szerint egy adott frekvenciájú jelre értelmezettek, ezért harmonikusokat tartalmazó jelek esetén az alap harmonikusokkal kell számolni.
4. Sorrendi összetevők mérése igen érzékeny a fázishibákra.

Sorrendi összetevők mérésének fázisérzékenysége

Ha egy tisztán pozitív sorrendű rendszer egyik fázisát φ fázishibával mérjük, akkor az U_a vektor felbontható U_{ax} és U_{ay} összetevőkre. Mivel

$$U_{ax} \approx U_a \quad \text{és}$$

$$U_{ay} = U_a * \sin \varphi$$

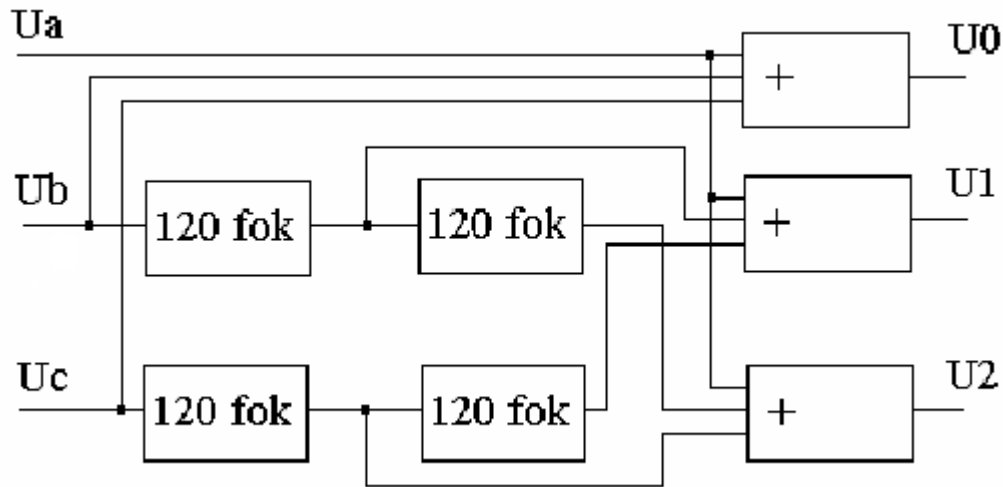


A mért negatív sorrendű összetevő értéke:

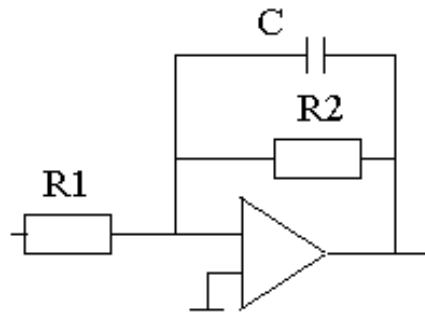
$$U_2 = U_a * \sin \varphi / 3$$

1 fokos fázishiba esetén a mért negatív sorrendű összetevő értéke 0.58 %!!!

Sorrendi összetevők mérése analóg módszerrel



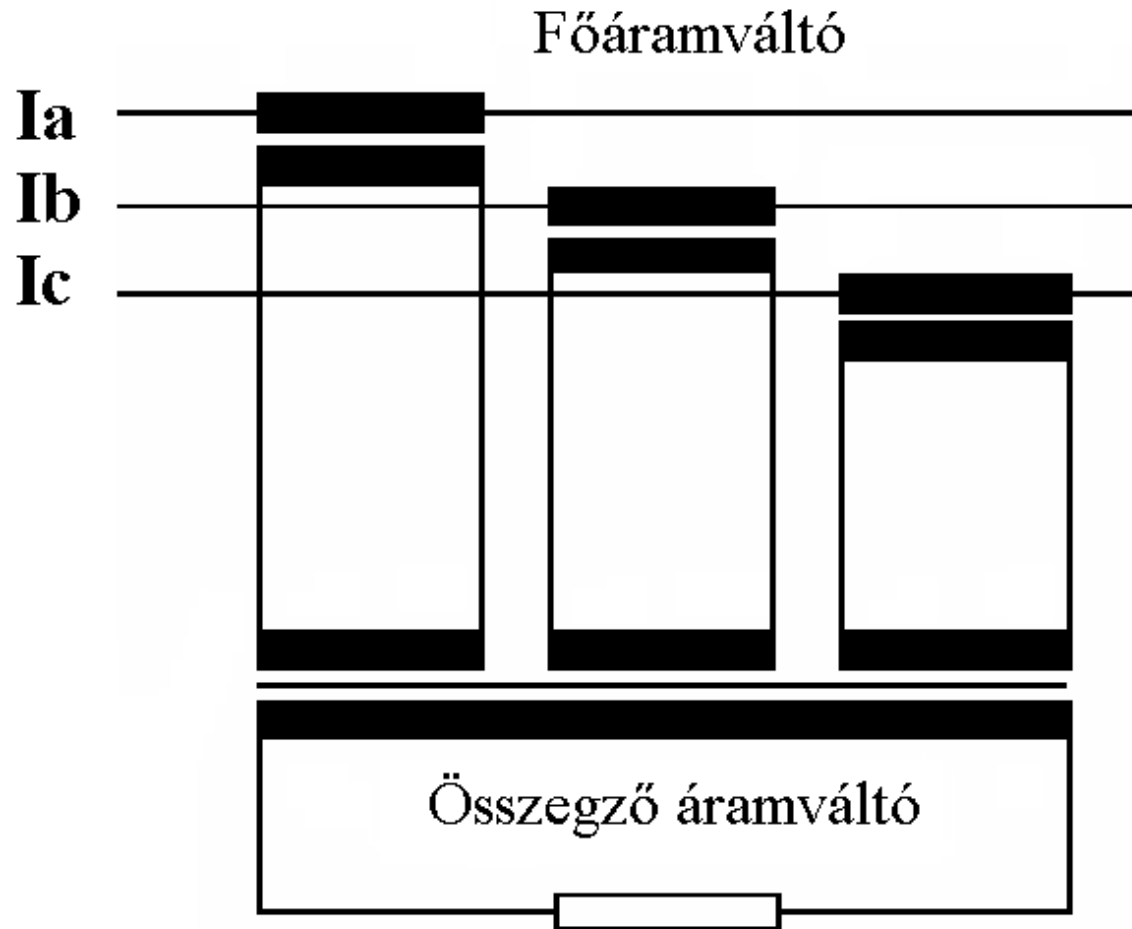
A méréshez 120 fokos fázistoló áramkört használunk.



A fázistolás 120 ($-180-60=-240=120$) fokos, ha $R2=2R1$ és az $R2, C$ visszacsatolás könyökpointi frekvenciája

$$f_0 = 50/\sqrt{3} = 28.86\text{Hz}$$

Összegző áramváltó zérus sorrendű áram mérésre



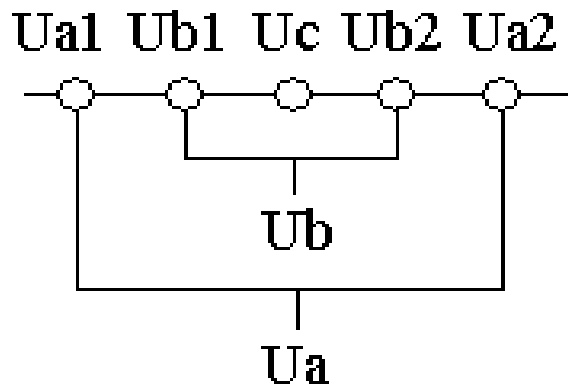
Sorrendi összetevők mérése digitális módszerrel

Kis értékek miatt nagyobb felbontást igényel.

A mintavételekből adódó fázishiba kompenzálása történhet:

- Egyidejű mintavétellel (Pl.: S/H) áramkörök alkalmazásával
- Többszörös mintavétellel

A fázisokból U_a, U_b, U_c, U_b, U_a sorrendben veszünk mintákat.



$$U_a = \frac{U_{a1} + U_{a2}}{2}$$

$$U_b = \frac{U_{b1} + U_{b2}}{2}$$

- A fázishiba kompenzálása szoftver módszerekkel.

Digitális mérési algoritmusok

1. Fázisforgatás digitális szűrővel.

A 120 fokos fázisforgatást mindent áteresztő szűrővel valósítjuk meg. (Erősítés a teljes frekvenciatartományban 1)

$$A = \frac{1 - P}{1 + P}$$

$$A = -\frac{1 - P}{1 + P}$$

A fázisforgatás:

$2 \cdot -60 = -120 = 240$
fokos lesz ha $f_0 = 28.86$ Hz

$2 \cdot -30 = -60$
fokos lesz ha $f_0 = 86.6$ Hz

A fázisforgatás:

$180 - 2 \cdot 60 = 60$
fokos lesz ha $f_0 = 28.86$ Hz

$180 - 2 \cdot 30 = 120$
fokos lesz ha $f_0 = 86.6$ Hz

A negatív sorrendű összetevő mérési algoritmus:

$$U_2 = \frac{1}{3} \sqrt{\frac{1}{N} \sum_{n=1}^N [U_a(n) + U_{b_{-120}}(n) + U_{c_{120}}(n)]^2}$$

ahol $U_a(n)$ az a fázis, $U_{b_{-120}}$ a b fázis -120 fokkal elforgatott n. mintája, U_2 a negatív sorrendű összetevő effektív értéke.

A mintavételekből eredő fázishiba korrigálható ha ezt beépítjük a fázisforgatásba. Pl. ha a mintavételek sorrendje U_a , U_b , U_c és két minta között 1 fok fáziseltolás (a mintavételi idő 55.5 μsec) lép fel, akkor negatív sorrend mérése esetén az U_b fázist 239=-121 fokkal, az U_c fázist 118 fokkal forgatjuk el.

A mérési algoritmus a harmonikusok hatását nem küszöböli ki, a minták tárolására nincs szükség.

2. Fázisforgatás ciklikus tár segítségével.

A mintákat ciklikus tárban helyezük el. A 120 fokos fázisforgatást az $N/3$ -mal korábbi minták felhasználásával képezzük.

A negatív sorrendű összetevő mérési algoritmus:

$$U_2 = \frac{1}{3} \sqrt{\frac{1}{N} \sum_{n=1}^N [U_a(n) + U_b(n - N/3) + U_c(n - 2N/3)]^2}$$

Az algoritmus két ciklikus tár használatát kívánja meg. A mintaszámnak 3-mal oszthatónak kell lenni.

A mintavételi fázishiba és a harmonikusok hatása szoftver úton nem kompenzálható.

3. Szorzás 120 fokkal eltolt szinusz értékekkel

Felhasználjuk a következő összefüggéseket:

$$\sin \alpha * \sin \beta = \frac{1}{2} [\cos(\alpha - \beta) - \cos(\alpha + \beta)]$$

$$\sin \alpha * \cos \beta = \frac{1}{2} [\sin(\alpha - \beta) + \sin(\alpha + \beta)]$$

Összegezzük egy periódusra a mintavett szinuszos jel és egy szinusz függvény szorzatát:

$$\sum_{i=1}^N A \sin(\omega t_i + \varphi) * \sin(\omega t_i + \gamma) = \sum_{i=1}^N \frac{1}{2} A \cos(\varphi - \gamma) = \frac{N}{2} A \cos(\varphi - \gamma)$$

$$\sum_{i=1}^N A \sin(\omega t_i + \varphi) * \cos(\omega t_i + \gamma) = \sum_{i=1}^N \frac{1}{2} A \sin(\varphi - \gamma) = \frac{N}{2} A \sin(\varphi - \gamma)$$

Az eredmény a mintavett jel - γ szöggel elforgatott vektorának valós és képzetes összetevői.

A három bemenő fázis mintáit az alábbiak szerint megszorozzuk a sint szinusz táblázatból (ciklikus módon) vett értékekkel, ekkor megkapjuk a negatív sorrendű összetevő értékét.

$$U_{2x} = \frac{1}{3} \left(\frac{2}{N} \sum_{i=1}^N (U_a(i) * \sin t[i] + U_b(i) * \sin t[i + N/3] + U_c(i) * \sin t[i + 2N/3]) \right)$$

$$U_{2y} = \frac{1}{3} \left(\frac{2}{N} \sum_{i=1}^N (U_a(i) * \sin t[i + 3N/12] + U_b(i) * \sin t[i + 7N/12] + U_c(i) * \sin t[i + 11N/12]) \right)$$

$$U_2 = \sqrt{U_{2x}^2 + U_{2y}^2}$$

Az eljárás egy szinusz táblázatot és sorrendi mennyiségeként két összegző tárat igényel.

A mintavételek számának 12-vel oszthatónak kell lennie.

A módszer nem érzékeny a felharmonikus tartalomra, de érzékeny a mintavételi fázishibára.

Az alábbi táblázat megadja, hogy az egyes sorrendekhez tartozó összegző tárákat 12 minta esetén a szinusz táblázat mennyivel eltolt elemeivel (ciklikus tár) képezzük:

sorrend:	0	1	2	0	1	2
tárák:	U0x	U1x	U2x	U0y	U1y	U2y
Ua	0	0	0	0	3	3
Ub	0	8	4	0	11	7
Uc	0	4	8	0	7	11

A mintavételi fázishiba kompenzálható, ha minden fázis külön szinusz táblázattal rendelkezik, amelyek az adott fázishoz tartozó szöggel vannak korrigálva.

4. Alapharmonikus vektorok mérése.

Meghatározzuk az egyes fázisok alapharmonikus értékeit mint vektorokat az effektív érték mérésénél megismert módon:

$$\bar{U} = \frac{1}{N} \sum_{n=0}^{N-1} U(n) \cos\left(n \frac{2\pi}{N}\right) - j \frac{1}{N} \sum_{n=0}^{N-1} U(n) \sin\left(n \frac{2\pi}{N}\right)$$

$$\bar{U} = U_x + j^* U_y$$

A sorrendi mennyiségeket az egyes vektorok elforgatásával számíthatjuk ki.

$$\bar{U}_0 = \frac{1}{3} (\bar{U}_a + \bar{U}_b + \bar{U}_c)$$

$$\bar{U}_1 = \frac{1}{3} (\bar{U}_a + a\bar{U}_b + a^2\bar{U}_c)$$

$$\bar{U}_2 = \frac{1}{3} (\bar{U}_a + a^2\bar{U}_b + a\bar{U}_c)$$

Az előző összefüggést kifejtve, figyelembe véve, hogy

$$a = e^{j2/3\pi} = -0.5 + j\sqrt{3}/2 = -0.5 + 0.86j$$

$$a^2 = e^{j4/3\pi} = -0.5 - j\sqrt{3}/2 = -0.5 - 0.86j$$

A negatív sorrendű összetevő számítható:

$$U_{2x} = \frac{1}{3} (U_{ax} - 0.5U_{bx} - 0.5U_{cx} + 0.86U_{by} - 0.86U_{cy})$$

$$U_{2y} = \frac{1}{3} (U_{ay} - 0.5U_{by} - 0.5U_{cy} - 0.86U_{bx} + 0.86U_{cx})$$

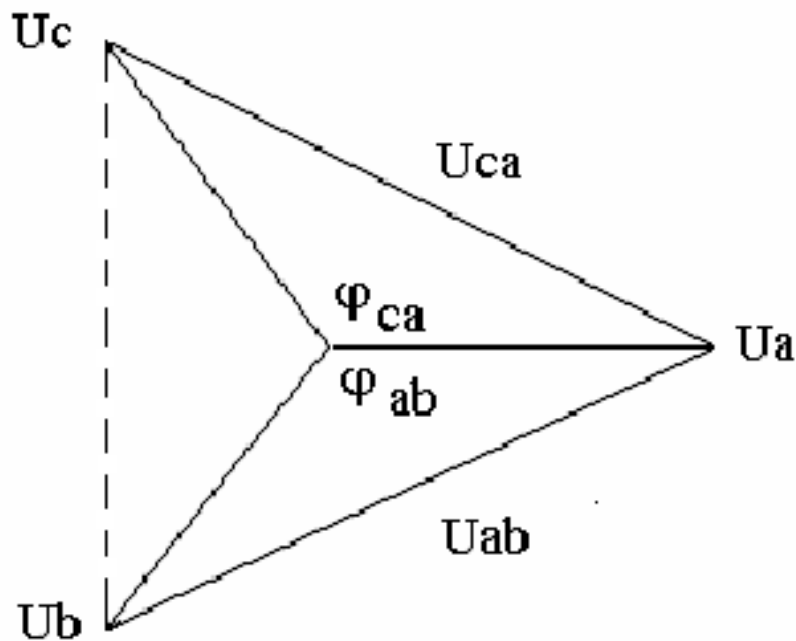
$$U_2 = \sqrt{U_{2x}^2 + U_{2y}^2}$$

A mintavételből eredő fázishiba kiküszöbölhető, ha az a és a^2 értékét megfelelően módosítjuk. Pl. ha a fázisok mintavétele között 1 fok van, akkor az alábbi értékekkel számolunk:

$$a_k = e^{j118^\circ} = -0.469 + 0.883j, \quad a_k^2 = e^{j239^\circ} = -0.515 - 0.857j$$

5. Háromszög módszer

A három fázis és legalább két vonali effektív értékből a fázisszögek a cosinus tétel ($c^2 = a^2 + b^2 - 2ab \cos \varphi$) segítségével meghatározhatóak és ezekből a sorrendi mennyiségek az előző módszer alapján számolhatóak:



$$\cos \varphi_{ab} = \frac{U_a^2 + U_b^2 - U_{ab}^2}{2U_a U_b}$$

$$\cos \varphi_{ca} = \frac{U_c^2 + U_a^2 - U_{ca}^2}{2U_a U_c}$$

Az egyes fázisokhoz tartozó vektorok:

$$\bar{U}_a = U_a e^{j0}$$

$$\bar{U}_b = U_b e^{-j\varphi_{ab}}$$

$$\bar{U}_c = U_c e^{j\varphi_{ca}}$$

A módszer nem érzékeny a mintavételek közötti fázis eltolásokra, mert az effektív érték nem függ a mérés kezdeti fázisától.

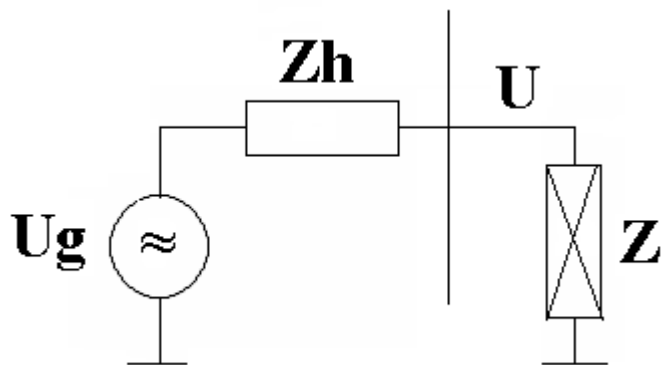
A harmonikusok jelenléte mérési hibát okoz ez az alapharmonikus effektív értékek mérésével küszöbölhető ki.

A flicker (villogás) fogalma és mérési módja

Irodalom: Dán András, Tersztyánszky Tibor, Varjú György
Villamosenergia minőség
Kiadó: INVEST-MARKETING Bt.

A flicker jelenség

A flicker (villogás) a hálózati feszültség kismértékű, alacsonyabb frekvenciájú (főleg az 5-15 Hz közötti) ingadozása. Az ingadozó feszültség a világító eszközök, elsősorban az izzólámpák fénykibocsátásának ingadozását, villogását okozza, amely kellemetlen, hosszabb idő alatt fejfájáshoz vezethet.

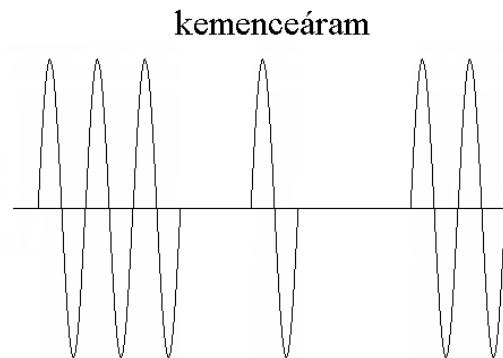
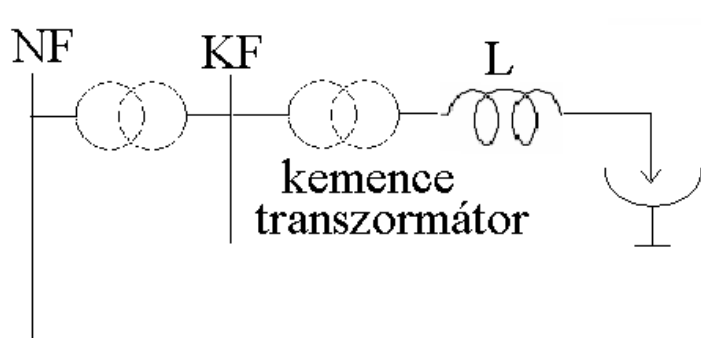


Kialakulásához kell a mögöttes hálózati impedancia

$$U = U_g \frac{Z}{Z + Z_h}$$

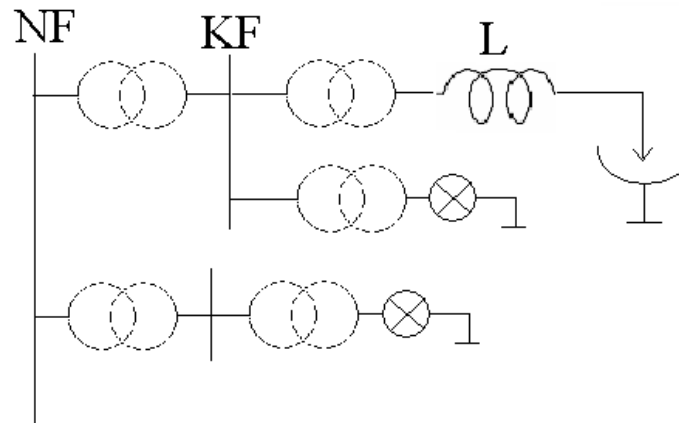
Z_h induktív, de kisfeszültségen inkább ohmos

Ívkemence

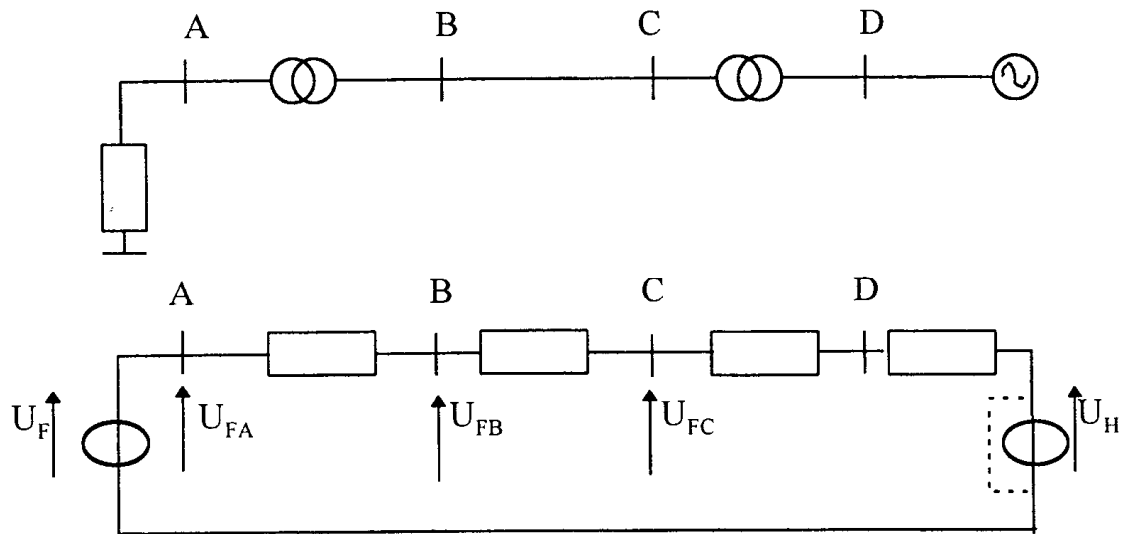


Olvadás, vas összeesik, ív megszakad, majd újra indul.
Elején induktív, aztán wattos.

Melyik lámpa villog jobban?



A villogás terjedése a kiefeszültségű oldalról



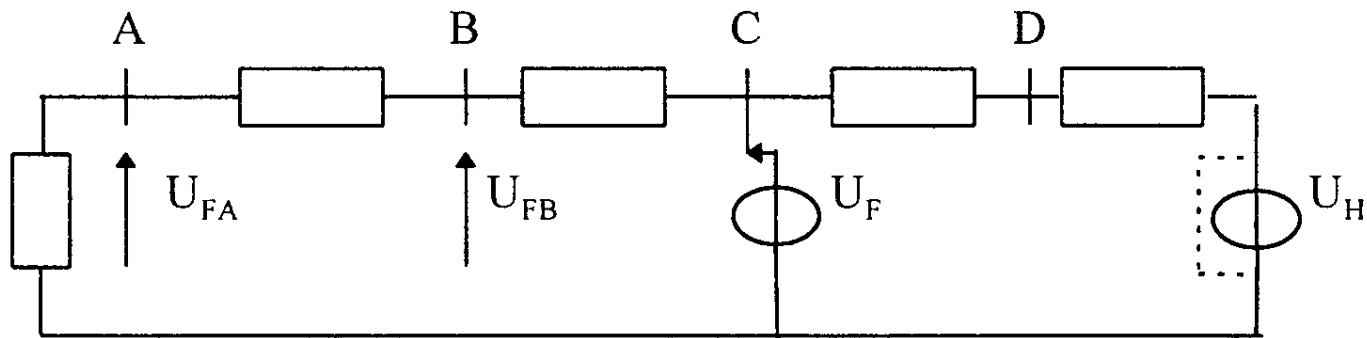
U_F a flikkerforrás a kiefeszültségű oldalon

B, C és D pontok közép illetve nagyfeszültségű gyűjtősínek

U_H feszültség a hálózat Thevenin feszültsége

A zavar hatása a betáplálási oldalon kisebb mértékben jelenik meg, mivel az U_F feszültség leosztódik a táphálózat felé haladva a soros impedanciákon.

A villogás terjedése közép vagy nagyfeszültségű oldalon



U_H a hálózat Thevenin feszültsége

B, C és D gyűjtősínek, mérési pontok közép illetve nagyfeszültségű oldalon

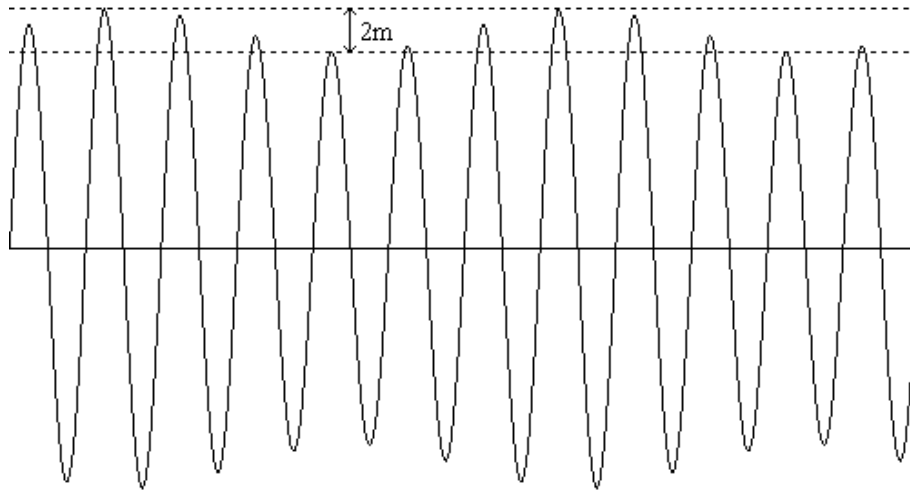
U_F a flikkerforrás a középfeszültségű oldalon

Mivel az A és C pontok közötti reaktancia sokkal kisebb mint az A pont és föld közötti, ezért U_F feszültsége szinte változatlanul megjelenik az A pontban.

A flicker jelenség matematikai leírása

A hálózati feszültség egy adott frekvenciával való ingadozása amplitudó modulált jellel írható le. Az ω_m frekvenciájú „m” modulációs mélységű hálózati feszültség az alábbi módon írható le:

$$U(t) = U_p * \sin(1 + m * \sin(\omega_m t)) * \sin(\omega_{50} t)$$



Ahol $2m$ a modulációs mélység csúcstól csúcsig.

A határérték meghatározása

A világítás intenzitásának változása az emberre az ingadozás frekvenciájától és amplitudójától is függ. A legerősebben a 8 Hz körüli ingadozások hatnak. A modulációs mélység és a frekvencia közötti összefüggést kísérleti alapon határozták meg. Azt a modulációs mélységet tekintjük határértéknek, amely az adott frekvenciával modulált hálózati feszültséggel táplált 230 V-os 60 W-os izzólámpa által megvilágított helységben a kísérletben résztvevő személyek fele érzékeli a fény változását.

A villogás keletkezése

A hálózati feszültség ingadozását a nagy teljesítményű gyorsan változó terhelések okozzák. Ilyenek elsősorban az ívkemencék, hegesztő berendezések, hengerművek.

A flicker mérése

A feszültség ingadozás mértékét az erre a célra kifejlesztett mérő berendezés segítségével mérhetjük. A mérőberendezés részletes kialakítását az IEC 868 közlemény alapján az MSZ EN 61000-4-15 szabja meg.

A mérőberendezés blokksémája az alábbi ábrán látható. Az egyes blokkok kialakítását a továbbiakban részletesen ismertetjük.

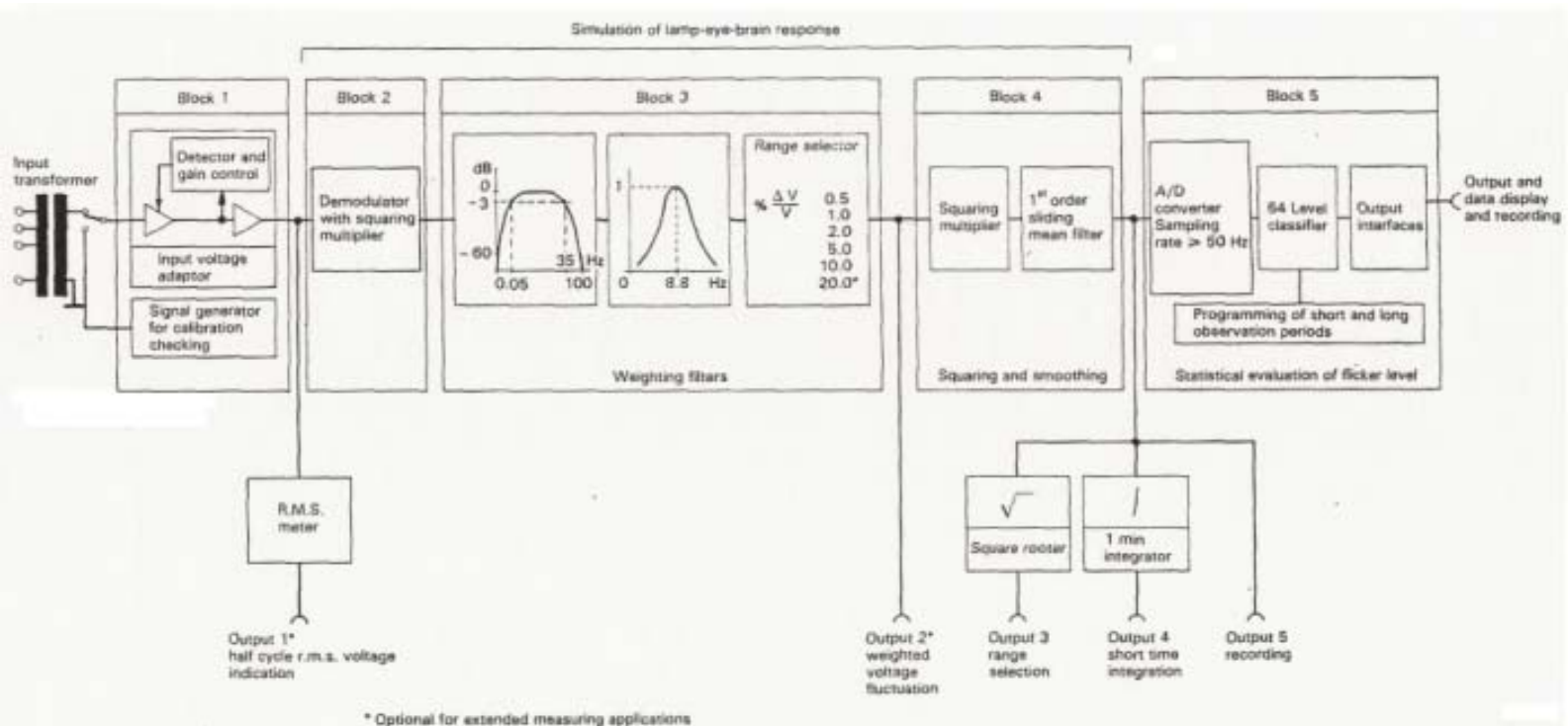
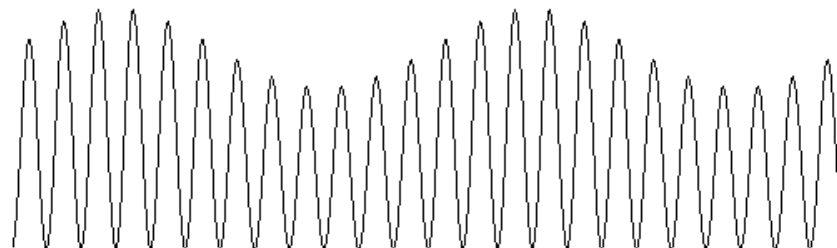
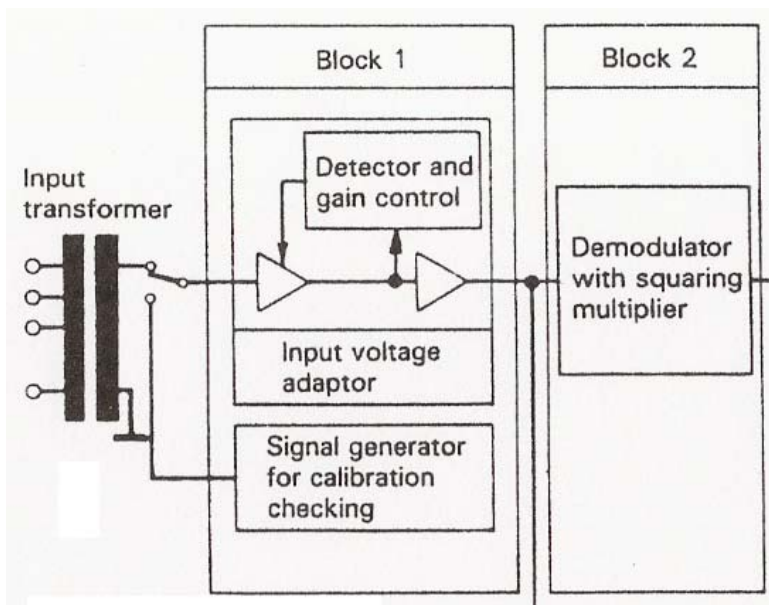


FIG. 1. — Functional diagram of UIE flickermeter.

Az 1. és a 2. blokk

Az 1. blokk tartalmazza a mérendő jel leválasztását és a szükséges erősítést. Tartalmazhat továbbá egy jelgenerátort a készülék kalibrálása és ellenőrzése céljából.

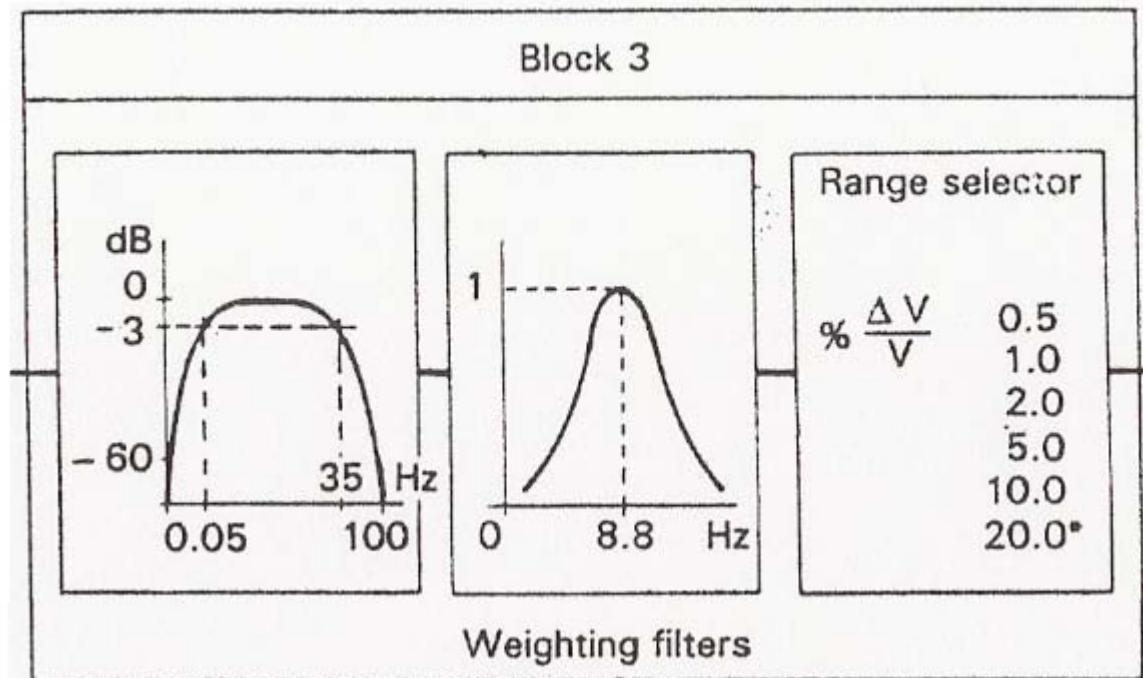
A 2. blokk tartalmazza az effektív érték méréséhez szükséges négyzetes egyenirányítót (demodulátort). A 2. blokk kimenő jele az ábrán látható.



A 3. blokk

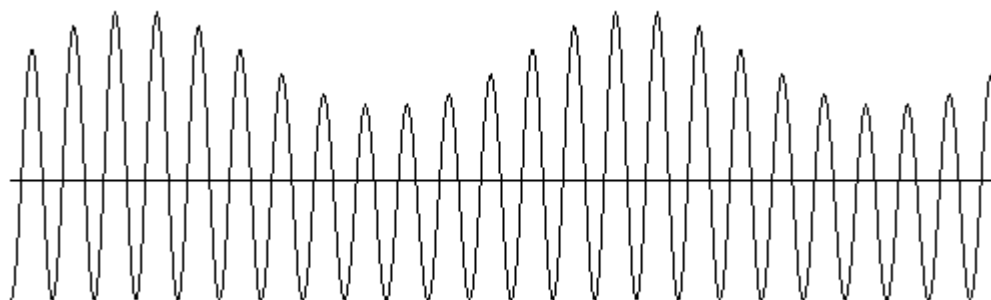
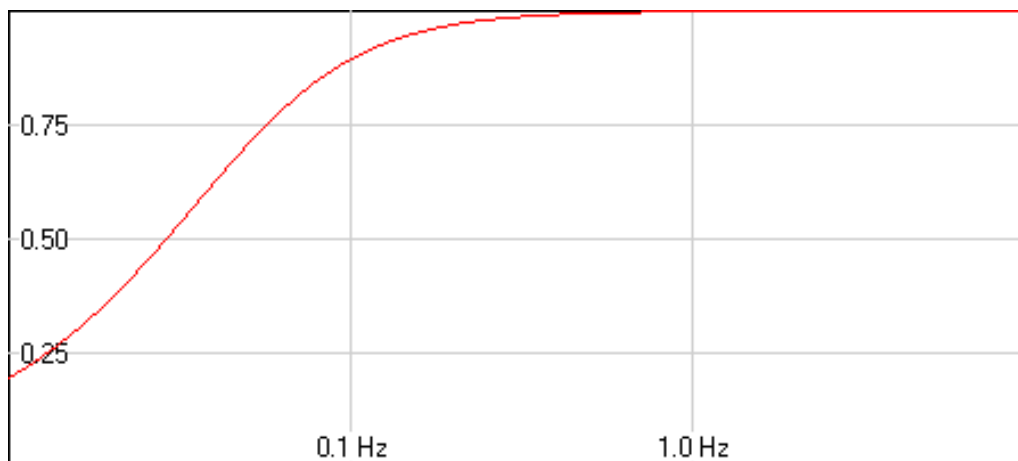
A 3. blokk feladata a moduláló jel leválasztása és a kísérletileg meghatározott emberi érzékenységi görbe megvalósítása.

Az előző fokozat jeléből egy felüláteresztő és egy aluláteresztő szűrő választja le az egyen összetevőt és a moduláló jelet, ez után következik az érzékenységnek megfelelő súlyozási görbe.



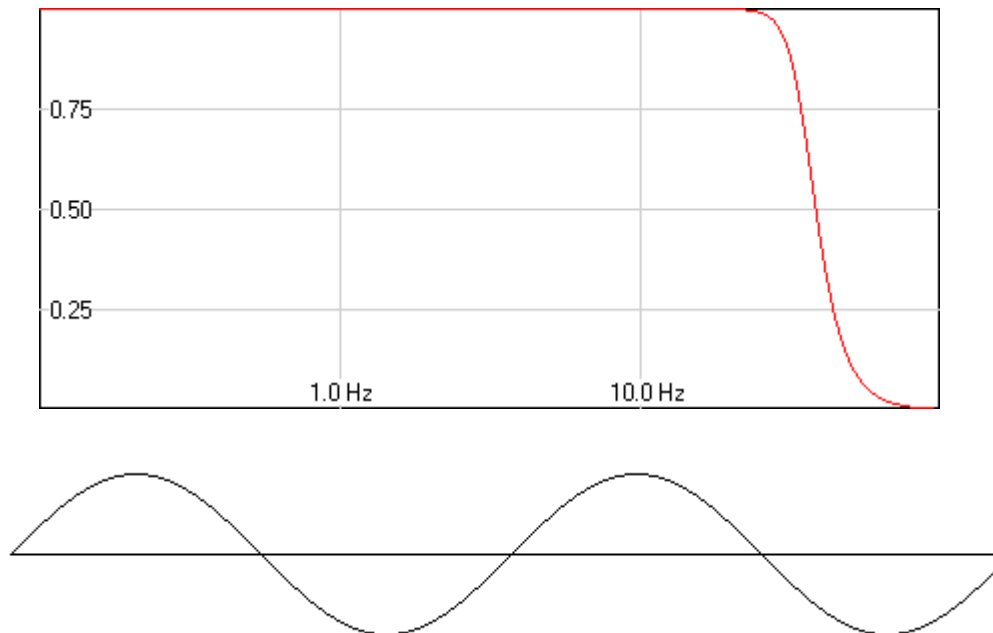
A 3. blokk megvalósítása (1.)

A 0.05 Hz-es elsőfokú felül áteresztő szűrő leválasztja az egyen összetevőt. A szűrő átviteli karakterisztikája és a kimenő jel alakja látható az alábbi ábrán.



A 3. blokk megvalósítása (2.)

A következő fokozat egy 35 Hz könyökpon­tú hatodrendű Butterworth típusú aluláteresztő szűrő. Ez leválasztja a magasabb frekvenciájú, elsősorban 100 Hz-es összetevőket. A szűrő karakterisztikája és a fokozat kimenő jele az ábrán látható. A kimenet már csak a moduláló jelet tartalmazza.



A 3. blokk megvalósítása (3.)

Az emberi érzékenységnek megfelelő súlyozási görbét a szabvány a következő képlettel írja elő:

$$F(s) = \frac{k\omega_1 s}{s^2 + 2\lambda s + \omega_1^2} * \frac{1 + s/\omega_2}{(1 + s/\omega_3) * (1 + s/\omega_4)}$$

ahol

$$k = 1.74802$$

$$\lambda = 2\pi * 4.05981$$

$$\omega_1 = 2\pi * 9.15494$$

$$\omega_2 = 2\pi * 2.27979$$

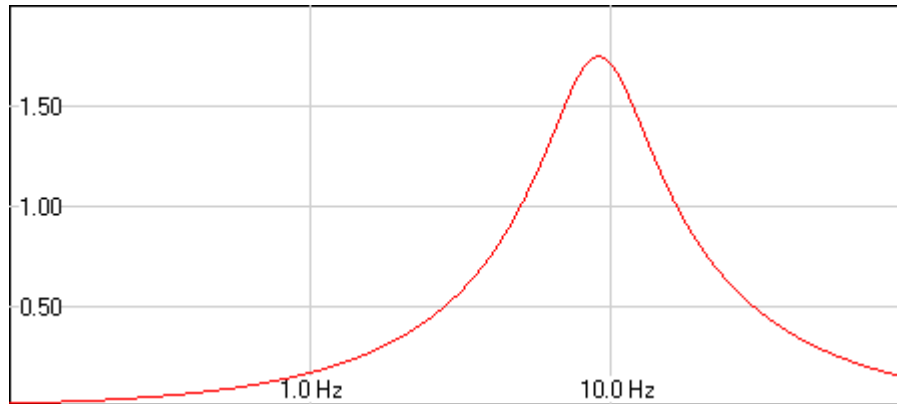
$$\omega_3 = 2\pi * 1.22535$$

$$\omega_4 = 2\pi * 21.9$$

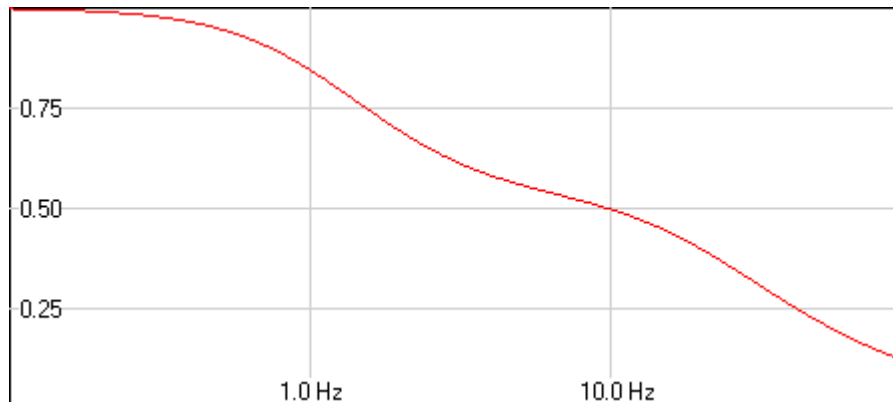
A 3. blokk megvalósítása (4.)

A súlyozási görbe első tagja egy sávszűrő, ahol

$$f_0 = 9.15 \text{ Hz} \quad Q = 1.127 \quad k = 1.748$$

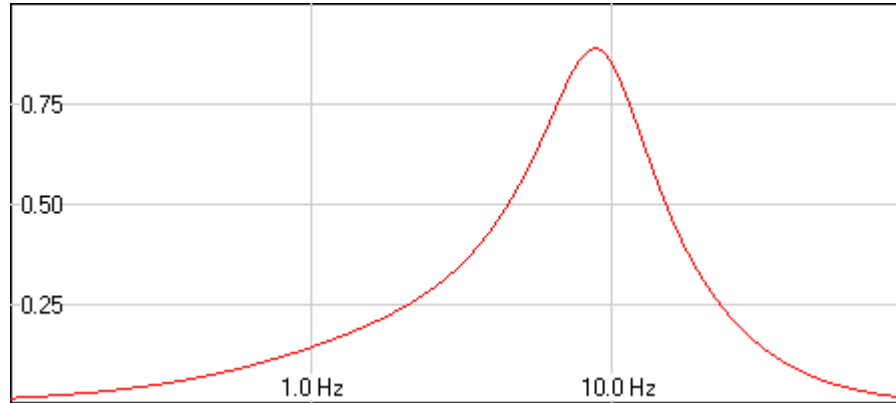


A súlyozási görbe második tagja egy 2.28 Hz-től emelkedő és két elsőfokú aluláteresztő (1.22 és 21.9 Hz) szűrő kombinációja.



A 3. blokk megvalósítása (5.)

A súlyozási görbe eredő karakterisztikája az alábbi:



A 3. blokk összesen 4 féle különböző szűrőt tartalmaz, ezek összes fok-száma 11. Ezeknek a szűrőknek megvalósítása analóg módszerrel igen terjedelmes áramköröket igényel és az előírt karakterisztika megvalósítása a kellő pontossággal alig lehetséges. A készülék teljes blokkvázlata csak az 5. blokkban tartalmaz mintavételi fokozatot (AD konvertert), azonban célszerű már a 2. blokkot és a 3. blokk szűrőit is digitális módon létrehozni.

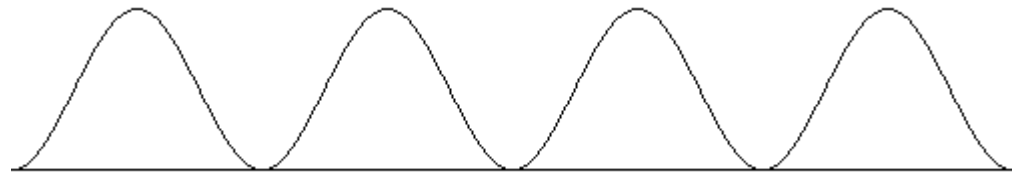
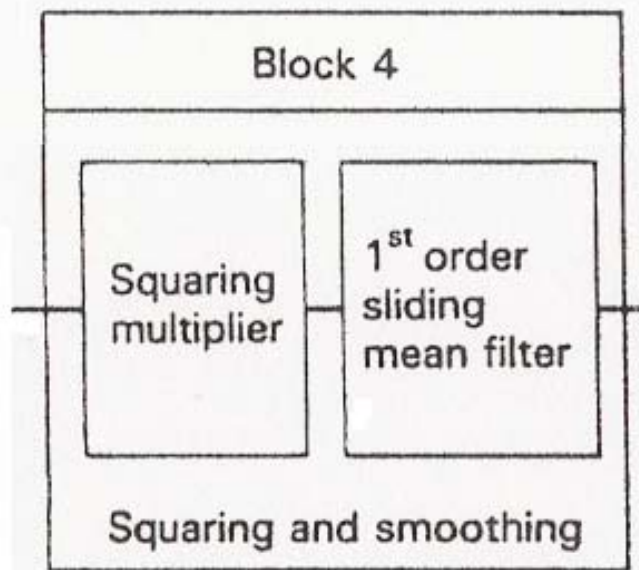
Szabvány szerinti kimeneti értékek

A szabvány szerint a készüléknek az alábbi táblázatban lévő modulációs frekvenciáknál a megadott modulációs mélység mellett azonos egységnyi eredményt kell szolgáltatnia $\pm 5\%$ pontossággal.

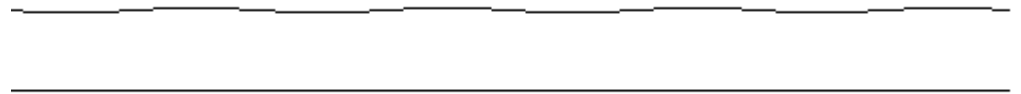
Hz	Voltage fluctuation (%)	Hz	Voltage fluctuation (%)
0.5	2.340	9.5	0.254
1.0	1.432	10.0	0.260
1.5	1.080	10.5	0.270
2.0	0.882	11.0	0.282
2.5	0.754	11.5	0.296
3.0	0.654	12.0	0.312
3.5	0.568	13.0	0.348
4.0	0.500	14.0	0.388
4.5	0.446	15.0	0.432
5.0	0.398	16.0	0.480
5.5	0.360	17.0	0.530
6.0	0.328	18.0	0.584
6.5	0.300	19.0	0.640
7.0	0.280	20.0	0.700
7.5	0.266	21.0	0.760
8.0	0.256	22.0	0.824
8.8	0.250	23.0	0.890
		24.0	0.962
		25.0	1.042

A 4. blokk

A 4. blokk a moduláló jel effektív értékét képezi. A 3. blokk kimenő jelét négyzetre emeli, majd ezt egy 300 msec időállandójú elsőrendű aluláteresztő szűrő simítja. A 4. blokk kimenete állandó moduláció esetén a modulációs mélységgel arányos egyenfeszültség.



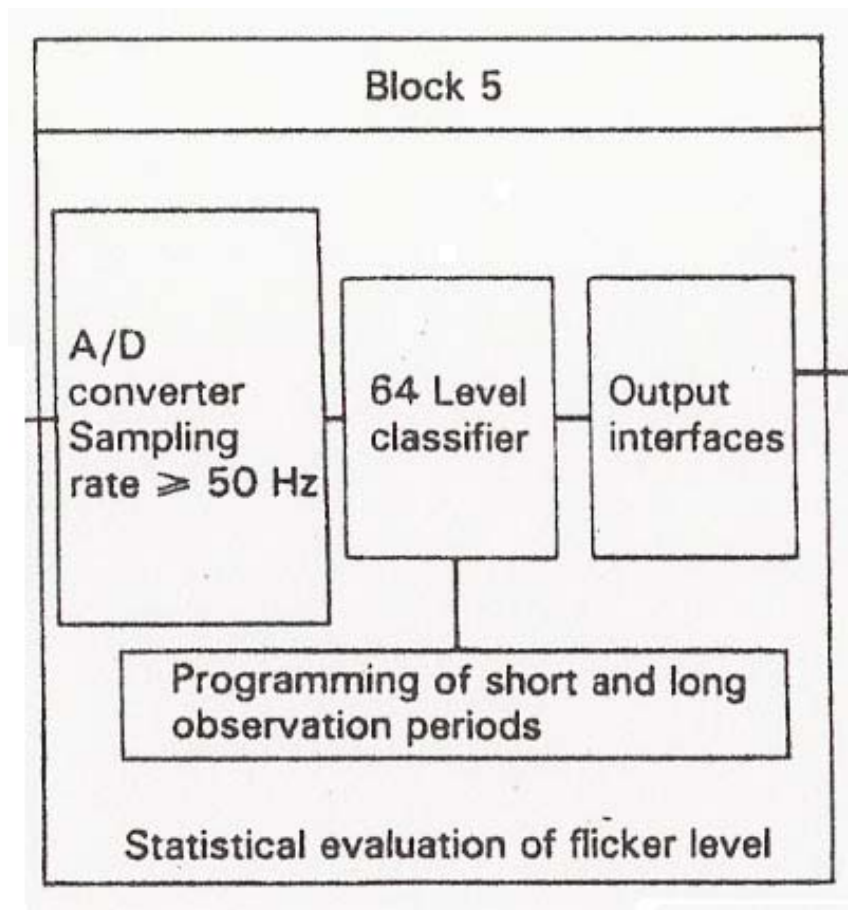
A négyzetes egyenirányító kimenete



A 4. blokk kimenőjele

Az 5. blokk

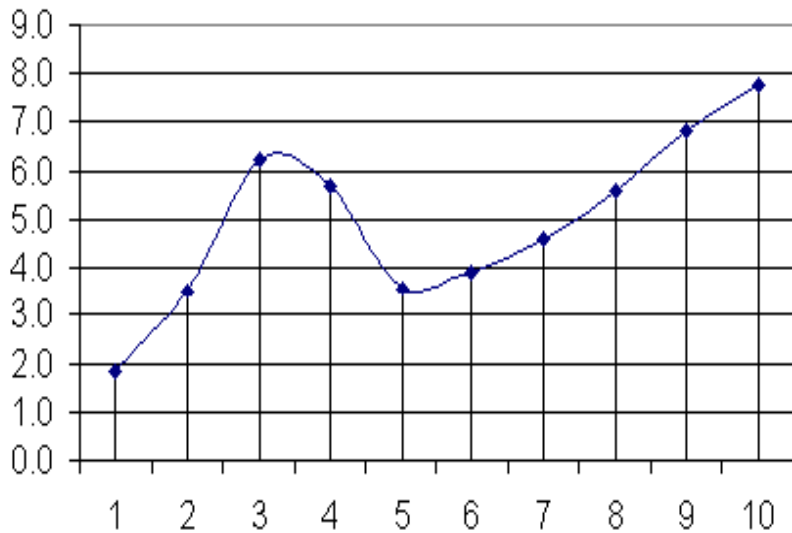
Az 5. blokk végzi a feszültség ingadozás statisztikai feldolgozását és megadja a flicker szabvány szerinti mérésének eredményét



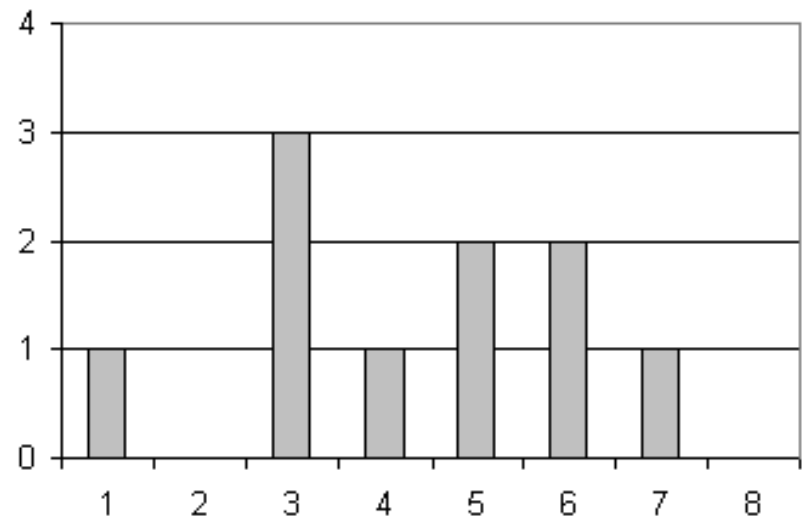
A 5. blokk megvalósítása (1.)

A 4. blokk kimenőfeszültsége az időben változó modulációnak megfelelően változik. Ezt a feszültséget mintavételezzük, az azonos értékeket összegyűjtve "dobozokba" helyezzük. Az egyes dobozok tartalma megadja hogy a doboznak megfelelő értékből hány mintát találtunk.

Az alábbi példában 10 mintát veszünk a feszültségből, amely 1 és 8 között változik, és előfordulásuk számát az ennek az értéknek megfelelő dobozokba helyezzük.



Mintavételezés az időfüggvényből



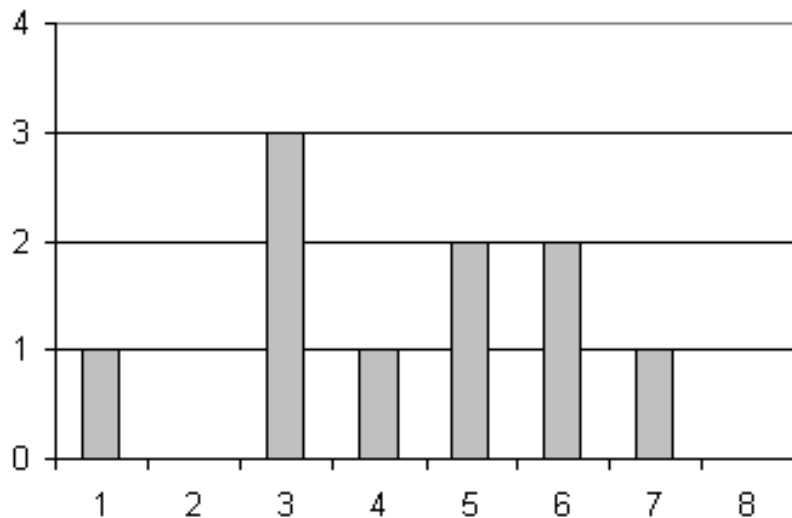
A dobozok az előfordulások számával

A 5. blokk megvalósítása (2.)

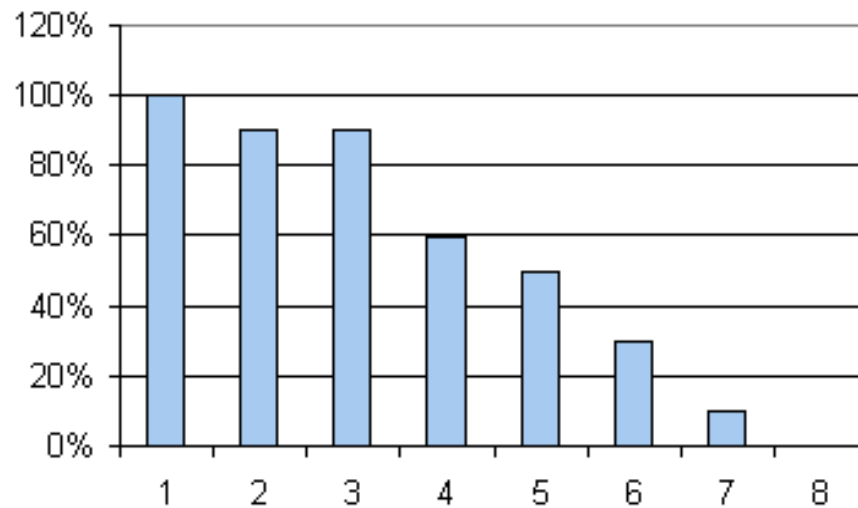
A mintavételezés frekvenciája min. 50 Hz, de célszerű nagyobb pl. 400 Hz-es mintavételt választani. A dobozok száma min. 64, azonban itt is ajánlott nagyobb számot választani. Mivel az ingadozás mértéke jelentős szórást mutat, a kis és nagy értékek egyforma pontosságú mérése érdekében a doboz értékek kiválasztásakor logaritmikus léptéket alkalmazhatunk.

Miután a mintákat a megfelelő dobozokba elhelyeztük, következik az adatok statisztikai feldolgozása.

Első lépésként képezzük azt a függvényt, amely minden dobozhoz megadja hogy a minták hány százaléka azonos vagy nagyobb mint a dobozhoz tartozó érték (eloszlásfüggvény).



A dobozok az előfordulások számával



A dobozokhoz tartozó eloszlásfüggvény

A 5. blokk megvalósítása (3.)

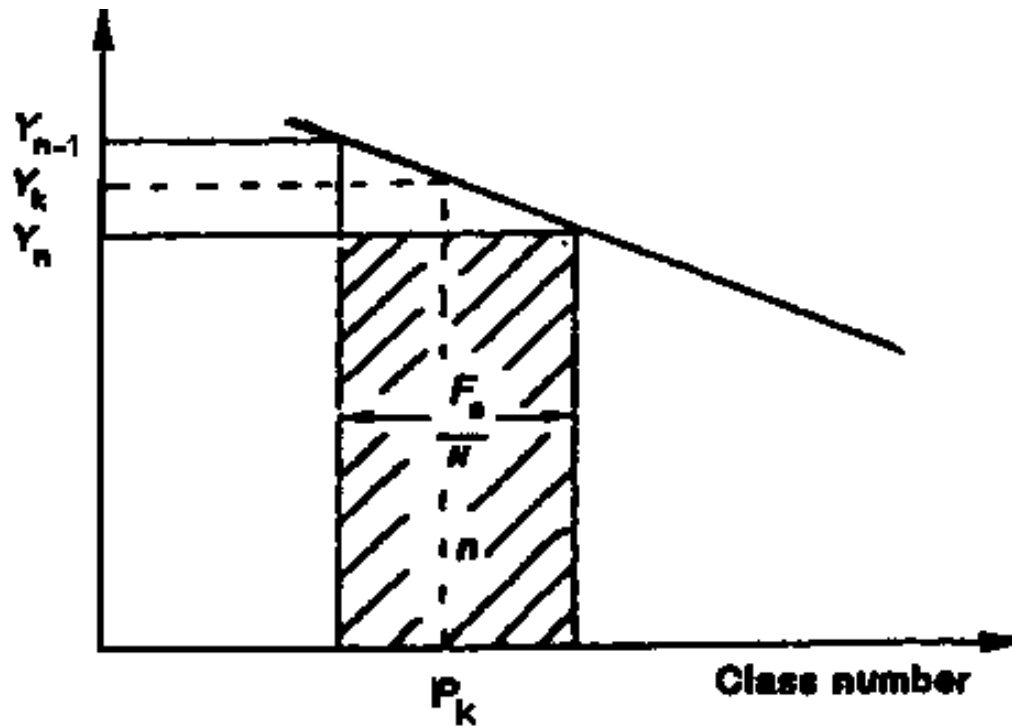
Jelöljük P_k -val annak a doboznak az értékét, amelynél az eloszlás függvény értéke k %. Például P_{50} az az érték, amelynél az eloszlás függvény értéke 50%, vagyis a 4. blokk kimenetéből vett minták 50%-a egyenlő vagy nagyobb mint P_{50} .

Az előző példában a P_{50} értéke pont az 5-ös doboz, mert az alatta lévő értékek amelyek ennél kisebbek $1+0+3+1=5$, a vele azonos vagy nagyobb érték pedig $2+2+1=5$.

Amennyiben a keresett százalék érték pontosan egyik helyen sem található, ilyen például a P_{70} , akkor a pontos eloszlásértéket interpolációval határozhatjuk meg. Az IEC 868 közlemény háromféle interpolációs eljárást ad meg.

A 5. blokk megvalósítása (4.)

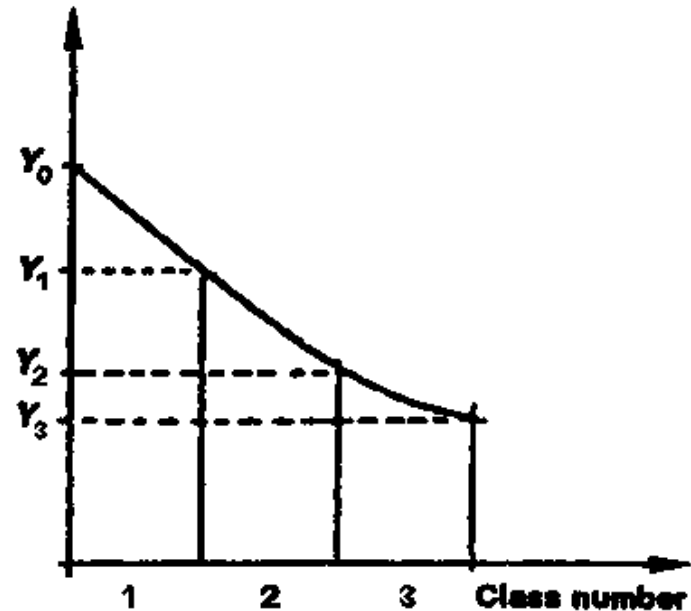
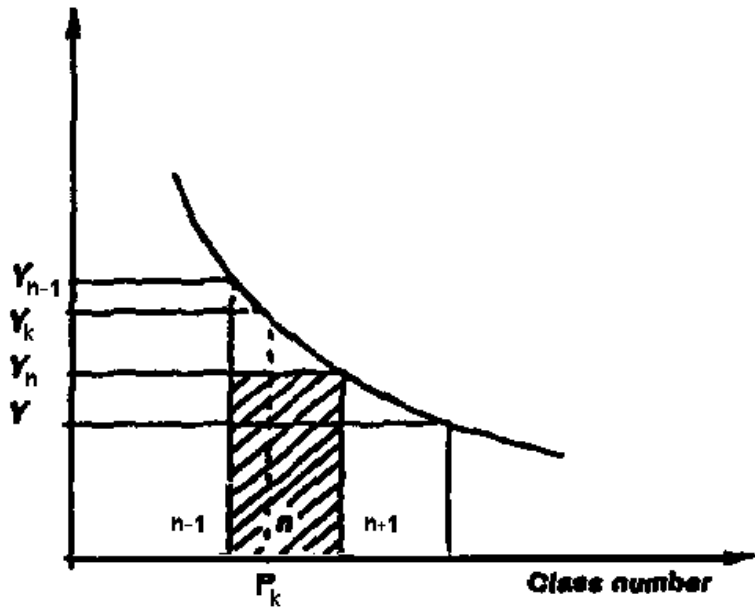
1. Lineáris interpoláció:



A 5. blokk megvalósítása (5.)

2. Nemlineáris interpoláció:

3. Pszeudo zérus interpoláció



Az eloszlásfüggvény alakjától függően célszerű választani az egyes interpolációs módszerek közül. A közlemény megadja az interpolációs módszerekhez tartozó képleteket.

A flicker szint meghatározása

Az eredő rövid idejű flicker szint a P_{st} (st = short time) meghatározásához 10 percig kell mintákat venni és az eloszlás függvény következő pontjait kell meghatározni:

$$P_{0.1} \quad P_1 \quad P_3 \quad P_{10} \text{ és } P_{50}$$

Az egyes P értékek meghatározása az alábbiak szerint történik:

$$P_1 = (P_{0.7} + P_1 + P_{1.5})/3$$

$$P_3 = (P_{2.2} + P_3 + P_4)/3$$

$$P_{10} = (P_6 + P_8 + P_{10} + P_{13} + P_{17})/5$$

$$P_{50} = (P_{30} + P_{50} + P_{80})/3$$

Az eredő flicker szint ezekből az alábbi képlettel számítható:

$$P_{st} = \sqrt{0.0314P_{0.1} + 0.0525P_1 + 0.0657P_3 + 0.28P_{10} + 0.08P_{50}}$$

Hosszú idejű flicker mérése

A hosszú idejű flickert, jelölése P_{lt} (long time), a hosszabb időn keresztül mért P_{st} (10 perces) értékekből számíthatjuk ki. A mérés ideje általában egy hét. Egy hét alatt összesen $N=1008$ ($168*6$) P_{st} értéket kapunk, ezekből a P_{lt} a következő képlettel számítható:

$$P_{lt} = \sqrt[3]{\frac{\sum_{i=1}^N P_{st_i}^3}{N}}$$

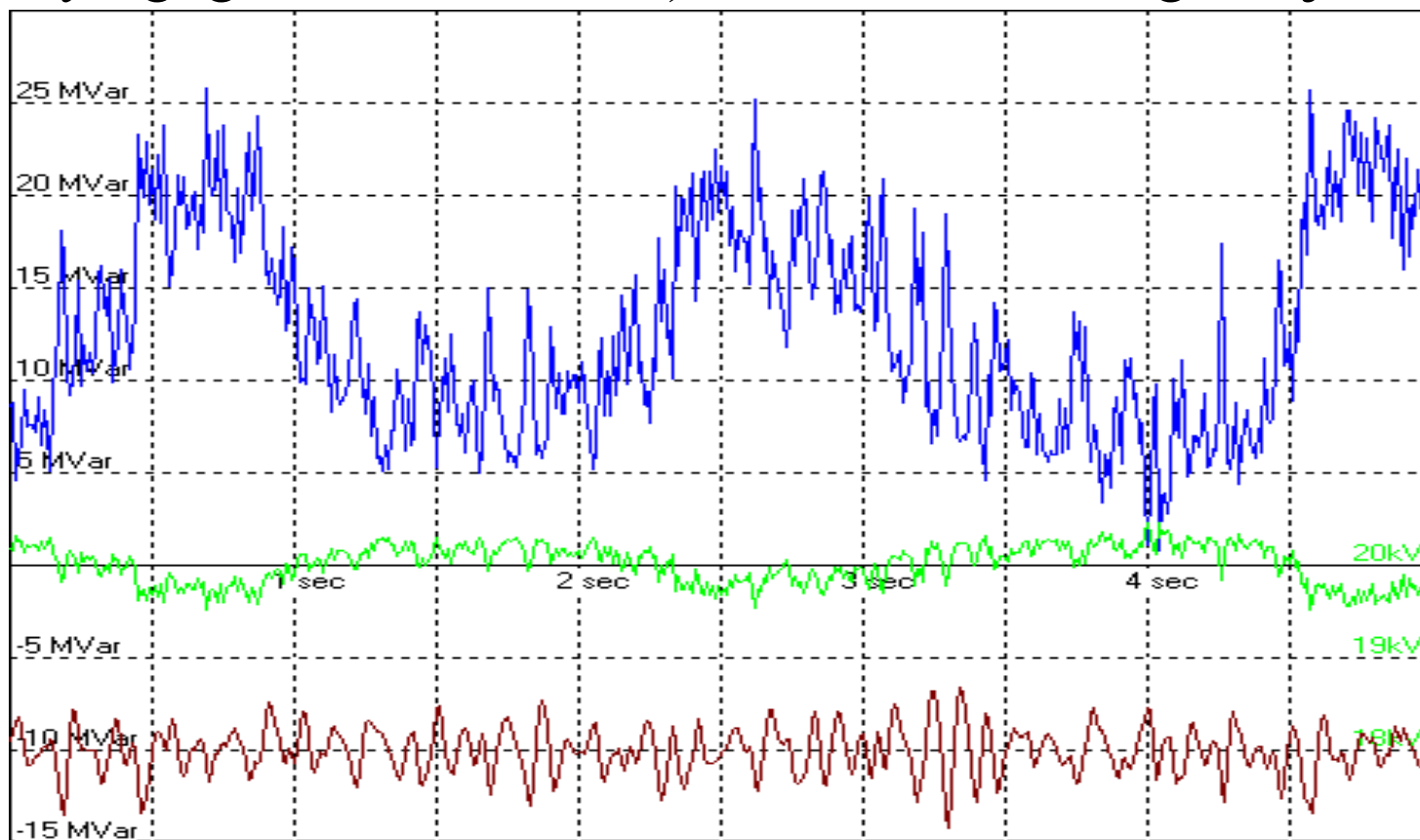
A mérőberendezés hitelesítése, vizsgálata

A teljes mérőberendezés az alábbi táblázat szerinti négyszögjellel modulált vizsgáló feszültségek esetén $P_{st}=1\pm 5\%$ értéket kell szolgáltatnia.

Változás percenként	Változás dU/U %
1	2,724
2	2,211
7	1,459
39	0,906
110	9,725
1620	0,402

Ívkemence 3 fázisú meddőteljesítmény felvétele

Az alábbi ábrán egy ívkemence jellegzetes 3 fázisú meddőteljesítmény felvétele (kék vonal) látható a beolvasztási periódusban. A zöld vonal a feszültség ingadozását mutatja, a barna vonal pedig a flicker mérő berendezés 3. blokkjának (az érzékenységi görbe szerint szűrt) kimeneti feszültségét adja meg.



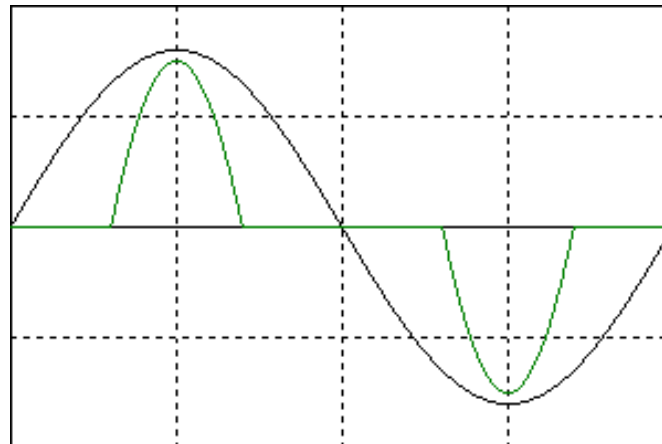
Harmonikusok

Harmonikusok keletkezése villamos hálózaton

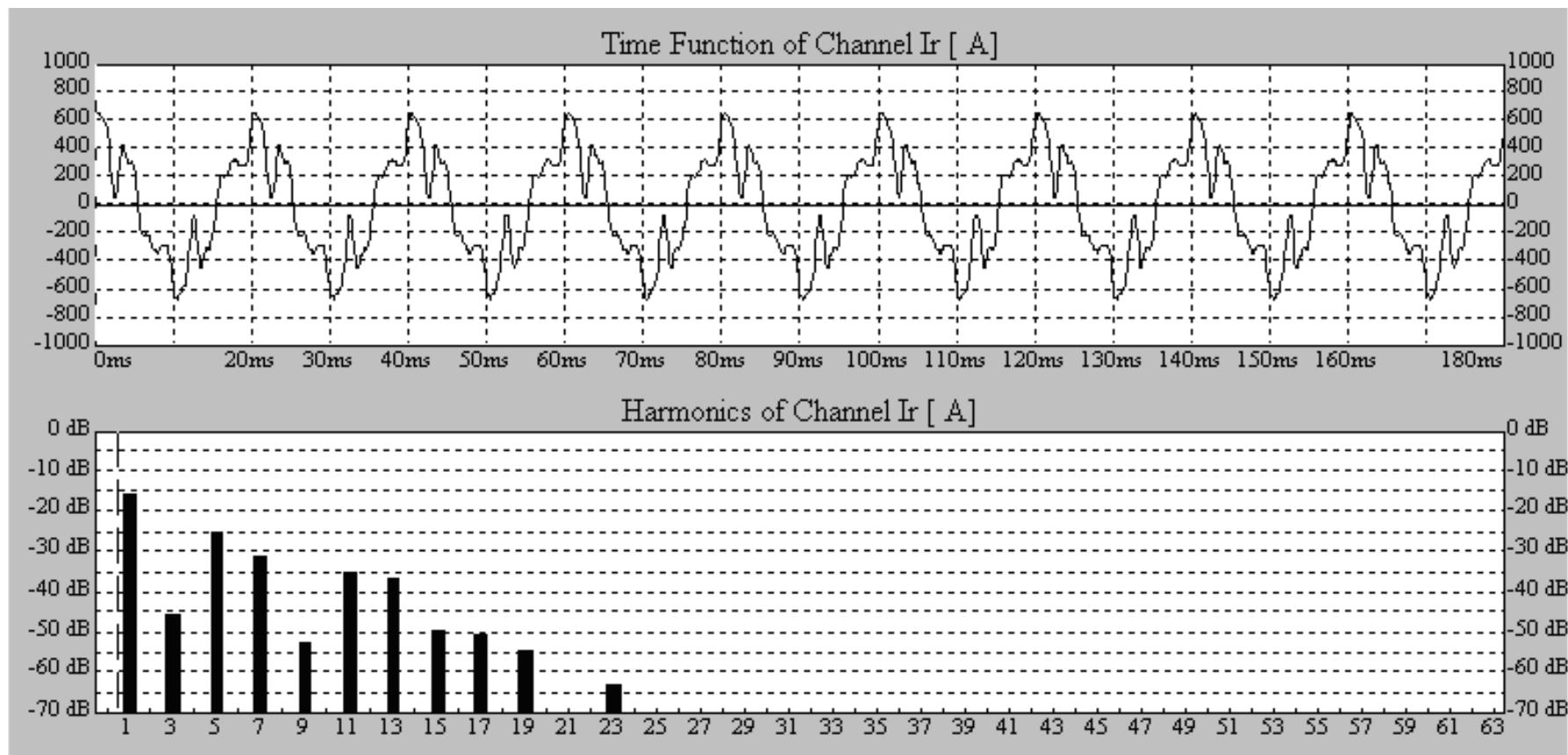
Harmonikusokat nemlineáris fogyasztók hoznak létre:

- Nagyteljesítményű egyenirányítók
- Hajtások
- Ívkemencék
- Villamos mozdonyok
- Transzformátorok
- Tápegységek

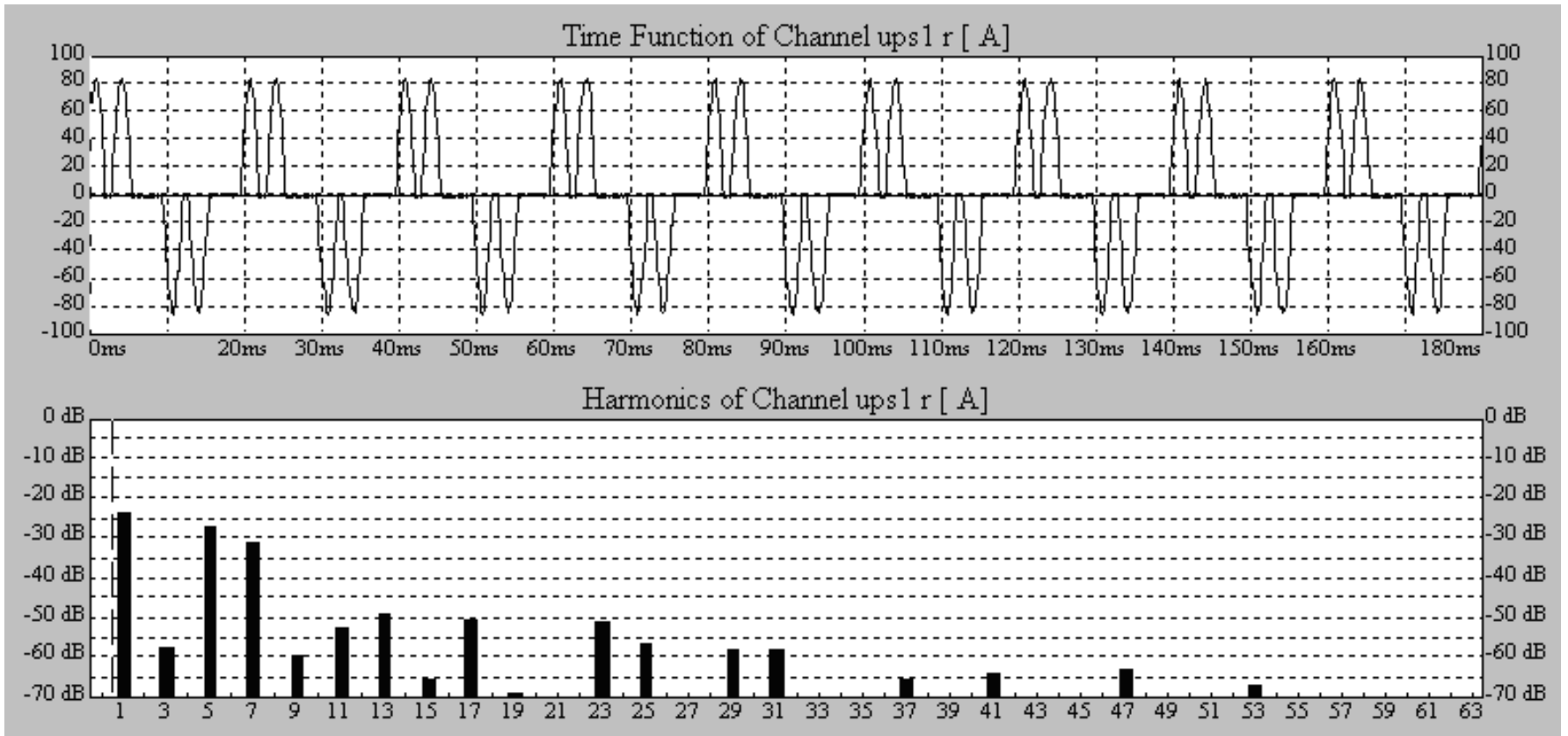
Tápegységek feszültség és áram alakja:



Irodaház betáp árama

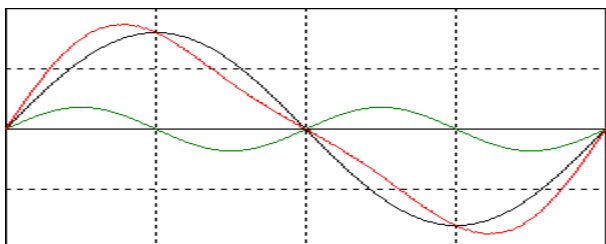


Szünetmentes hálózati oldali tápáram

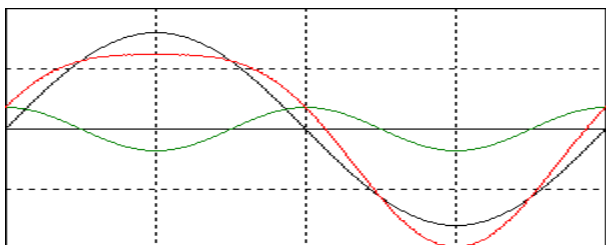


Időfüggvények harmonikusokkal

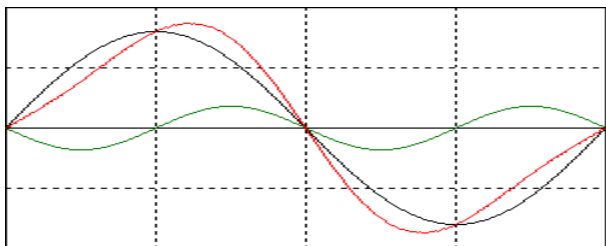
2. harmonikus



$f_i=0$

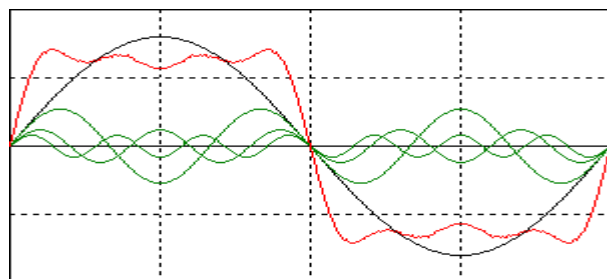
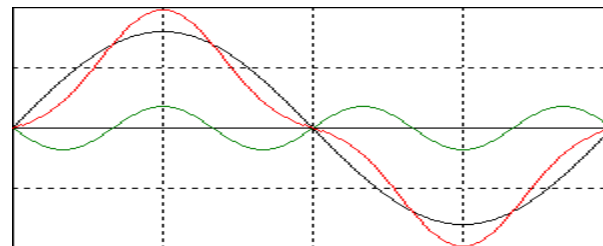
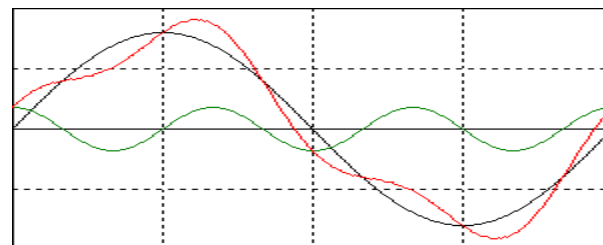
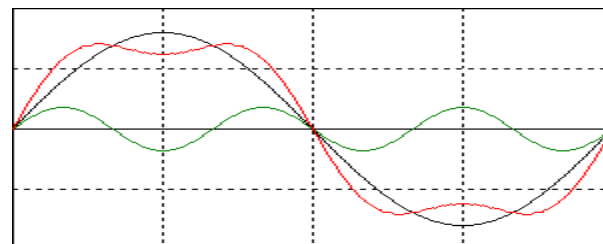


$f_i=90$



$f_i=180$

3. harmonikus



3.+5.+7. harmonikus

Harmonikusok mérése analóg módszerrel

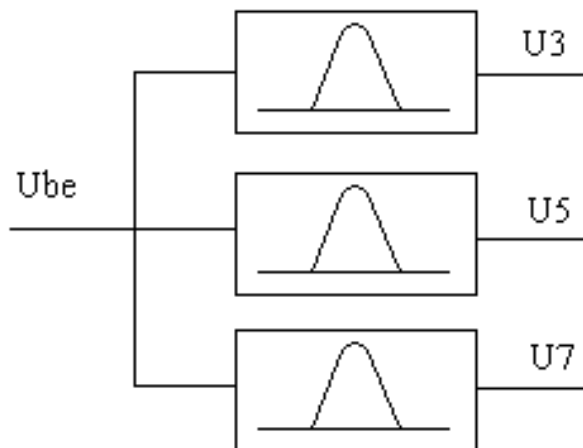
Egyedi harmonikus mérhető a harmonikus frekvenciára hangolt sávszűrővel.

Több harmonikus méréséhez több sávszűrőt kell alkalmazni.

A módszer hátránya:

Harmonikusokat általában az 63. rendszámig mérjük, ez igen sok szűrőt igényelne.

Magasabb harmonikusok esetén igen nagy jósági tényezőjű szűrők kellene. Pl. a 17. harmonikus szűrő esetén $Q=90$ szükséges ahhoz, hogy a 19. harmonikus elnyomása 40 dB legyen.



Szelektív erősítő

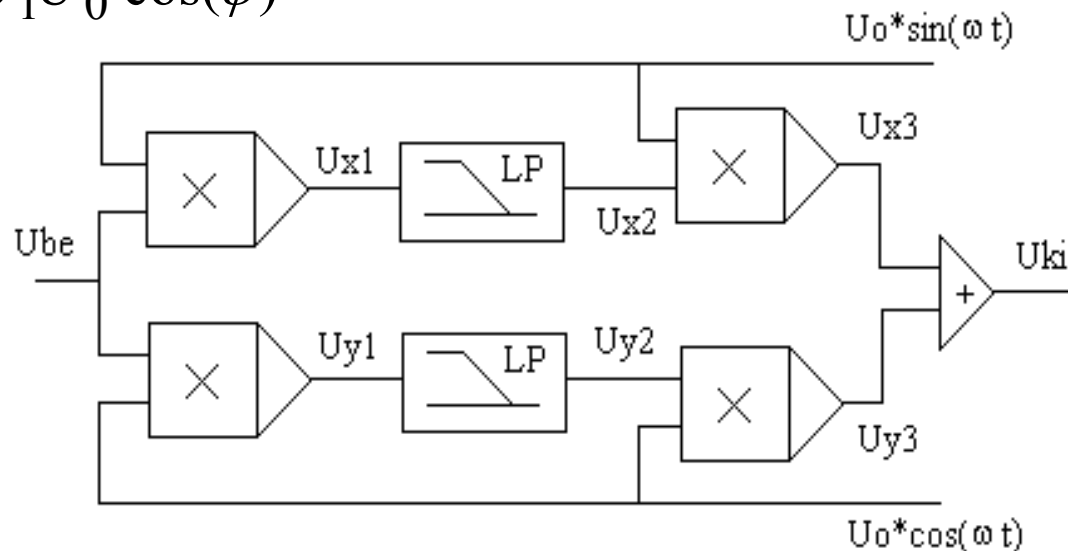
Szorzók és aluláteresztő szűrővel igen keskeny sáv szélességű szelektív erősítő készíthető, ezzel magasabb harmonikusokat ill. harmonikusok közelében is lehet mérni.

Ha a bemeneti jel $U_{be} = U_1 \sin(\omega t + \varphi)$, akkor:

$$U_{x1} = U_1 \sin(\omega t + \varphi) * U_0 \sin(\omega t) = \frac{1}{2} U_1 U_0 (\cos(\varphi) - \cos(2\omega t + \varphi))$$

Az aluláteresztő szűrő kimenetén:

$$U_{x2} = \frac{1}{2} U_1 U_0 \cos(\varphi)$$



A második szorzó kimenete:

$$U_{x3} = \frac{1}{2} U_1 U_0^2 \cos(\varphi) * \sin(\omega t)$$

A másik ágon hasonló eljárással kapjuk:

$$U_{y3} = \frac{1}{2} U_1 U_0^2 \sin(\varphi) * \cos(\omega t)$$

Az összegző erősítő kimenete:

$$U_{ki} = U_{x3} + U_{y3} = \frac{1}{2} U_1 U_0^2 \sin(\omega t + \varphi)$$

A kimeneti feszültség a bemeneti feszültséggel arányos.

Ha a bemeneti feszültség frekvenciája eltér az ω frekvenciától, akkor a kimenő feszültség oly mértékben csökken mint az aluláteresztő szűrő frekvencia karakterisztikája. Az aluláteresztő szűrő határfrekvenciája akár 1 Hz alatt is lehet.

Digitális jelfeldolgozás esetén a második szorzó áramkör elmaradhat, az U_{x2} és U_{y2} jelek közvetlenül feldolgozhatók.

DFT (Digital Fourier Transformation) algoritmus

Elméleti alapok

Az $f(t)$ időfüggvény spektrumát a Fourier integrál segítségével számíthatjuk ki:

$$F(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

Periodikus jel esetén elég az integrálást egy periódus (T) hosszan kel elvégezni:

$$F(\omega) = \frac{1}{T} \int_0^T f(t) e^{-j\omega t} dt$$

Mintavételezett jelek esetén az integrálás helyett összegzést végzünk

$$F(\omega) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-j\omega * nT / N}$$

A k. harmonikus értéke (az alap harmonikus itt $f_1=1/T$)

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-jk\omega_1 * nT / N}$$

Átalakítva adódik:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

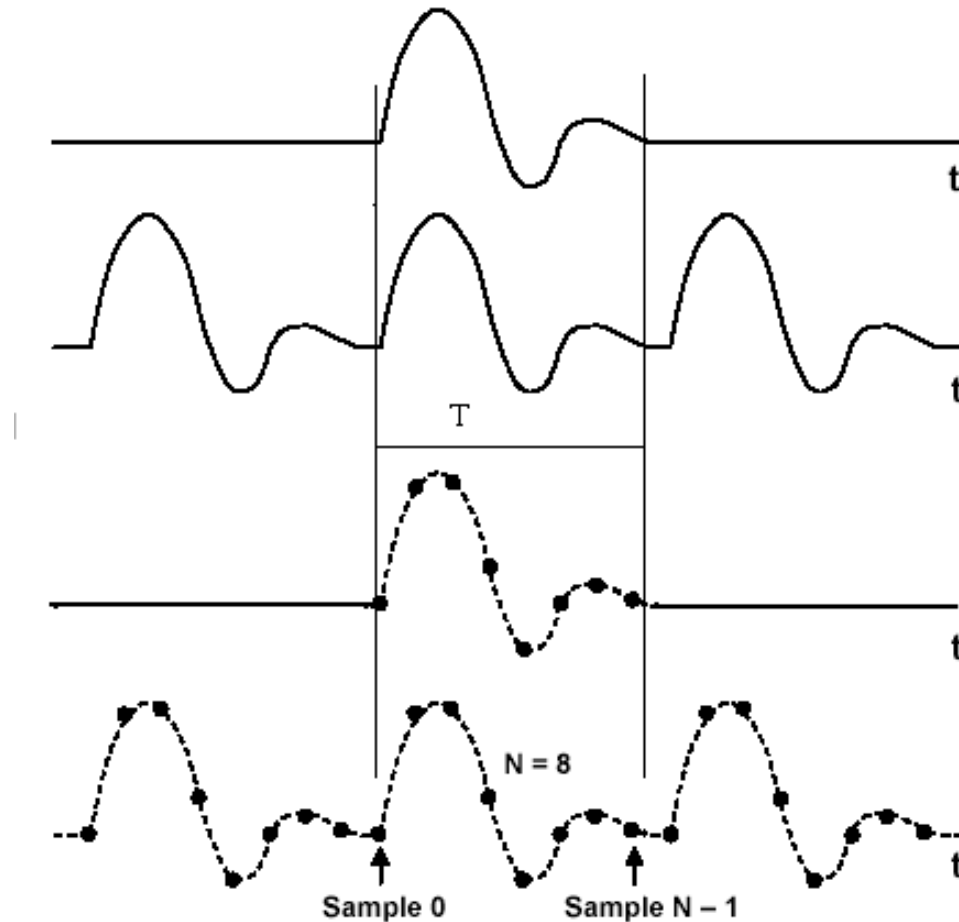
ahol $W_N = e^{-j\frac{2\pi}{N}}$ az N. komplex egységgyök

Periodikus jel mintavételezése

A legalacsonyabb előforduló frekvencia $f_1=1/T$

Az előforduló harmonikusok $f_k=k*f_1$

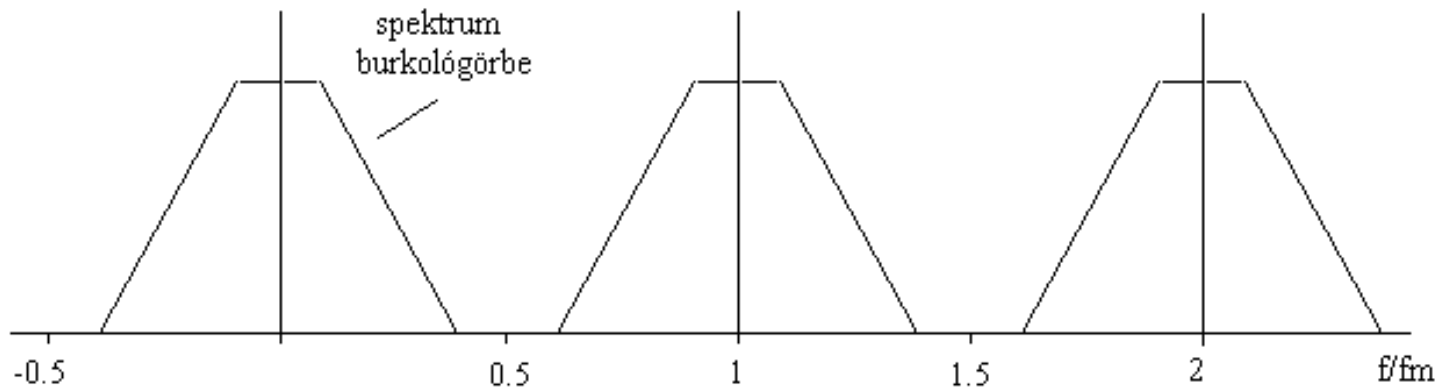
A legmagasabb frekvenciájú harmonikus értéke $k=N/2-1$



A spektrumvonalak $fm/2$ értékre tükrözve és fm egész számú többszöröseinél megismétlődnek, mivel

$$W_N^{n(N+k)} = W_N^{nN} * W_N^{nk} = W_N^{kn}$$

$$W_N^{n(N-k)} = W_N^{nN} * W_N^{-nk} = W_N^{-kn}$$



Digitális algoritmus N mintával.

A mérendő k. harmonikus: $U(t) = C_k \sin(k\omega t + \varphi)$

DFT integrálással:

$$A = \frac{1}{2\pi} \int_0^{2\pi} C_k \sin(k\omega t + \varphi) * \sin(k\omega t) = \frac{1}{2\pi} \int_0^{2\pi} \frac{C_k \cos(\varphi) - C_k \cos(2k\omega t + \varphi)}{2} = \frac{C_k \cos(\varphi)}{2}$$

$$B = \frac{1}{2\pi} \int_0^{2\pi} C_k \sin(k\omega t + \varphi) * \cos(k\omega t) = \frac{1}{2\pi} \int_0^{2\pi} \frac{C_k \sin(\varphi) - C_k \sin(2k\omega t + \varphi)}{2} = \frac{C_k \sin(\varphi)}{2}$$

Ebből adódik: $C_k = \sqrt{(2A)^2 + (2B)^2}$

Felhasznált trigonometrikus összefüggések:

$$\sin(\alpha) * \sin(\beta) = \frac{\cos(\alpha - \beta) - \cos(\alpha + \beta)}{2}$$

$$\sin(\alpha) * \cos(\beta) = \frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2}$$

$$\sqrt{\sin^2(\alpha) + \cos^2(\beta)} = 1$$

Az integrálást összegzéssel felcserélve kapjuk DFT algoritmust:

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos\left(kn \frac{2\pi}{N}\right) - j \frac{1}{N} \sum_{n=0}^{N-1} x(n) \sin\left(kn \frac{2\pi}{N}\right)$$

A mintákat a megfelelő szinusz és koszinusz függvényekkel szorozva és két összegző tárban összegezve megkapjuk az $X(k)$ komplex amplitúdó valós és képzetes részét. Ebből a spektrum összetevő kiszámítható:

$$A_k = \operatorname{Re}(X_k) / 2 \quad B_k = \operatorname{Im}(X_k) / 2$$

$$C_k = \sqrt{(2A_k)^2 + (2B_k)^2}$$

A fenti algoritmus nagyon számításigényes, egy spektrum összetevő kiszámítása $2N$ szorzást igényel, a teljes algoritmus szorzásigénye N^2 nagyságrendű.

A számításigény csökkenthető ha nem a teljes spektrumot hanem csak néhány harmonikust számítunk ki.

Az algoritmus alkalmas az alap harmonikus összetevő kiszámítására is.

Teljes harmonikus torzítás

A THD a harmonikusok effektív értékének viszonya az alapharmonikus effektív értékhez.

$$\text{THD} = \frac{\sqrt{\sum_{k=2}^n U_k^2}}{U_1}$$

A THD mérésének másik módja:

Az egyes harmonikusok ismeretében a teljes harmonikus torzítás THD (Total Harmonic Distortion) számítható:

$$\text{THD} = \frac{\sqrt{\sum_{k=2}^n U_k^2}}{U_1} = \frac{\sqrt{\sum_{k=1}^n U_k^2 - U_1^2}}{U_1} = \frac{\sqrt{U_{\text{rms}}^2 - U_1^2}}{U_1}$$

FFT (Fast Fourier Transformation) algoritmus

A k . harmonikus számítása DFT módszerrel:

N minta esetén a k . harmonikus komplex alakját az alábbi kifejezéssel számíthatjuk, ahol W_N az N . komplex egységgyök:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad k = 0 \rightarrow N-1$$

ahol $W_N = e^{-j\frac{2\pi}{N}}$ az N . komplex egységgyök

($1/N$ -t elhanyagoltuk, mert csak léptékezés kérdése.)

Az előbbi kifejezést felbontjuk a minta szerinti páros és páratlan részre:

$$X(k) = \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{(2n+1)k}$$

$$= \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1)W_N^{2nk}$$

$$k = 0 \rightarrow N - 1$$

Végezzük el az alábbi behelyettesítéseket:

$$W_N^{2nk} = \left(e^{-j2\pi/N} \right)^{2nk} = \left(e^{-j2\pi/(N/2)} \right)^{nk} = W_{N/2}^{nk}$$

$$x_1(n) = x(2n)$$

$$x_2(n) = x(2n+1)$$

ahol $x_1(n)$ a páros és $x_2(n)$ a páratlan minták

$$\begin{aligned} X(k) &= \sum_{n=0}^{N/2-1} x_1(n) W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x_2(n) W_{N/2}^{nk} = \\ &= Y(k) + W_N^k Z(k) \end{aligned}$$

$$k = 0 \rightarrow N-1$$

$Y(k)$ és $Z(k)$ a páros és páratlan $N/2$ pontos mintákból képzett komplex harmonikusok.

A $k+N/2$ harmonikusok számításához felhasználjuk, hogy

$$W_N^{k+N/2} = W_N^k * W_N^{N/2} = -W_N^k$$

$$X(k) = Y(k) + W_N^k Z(k)$$

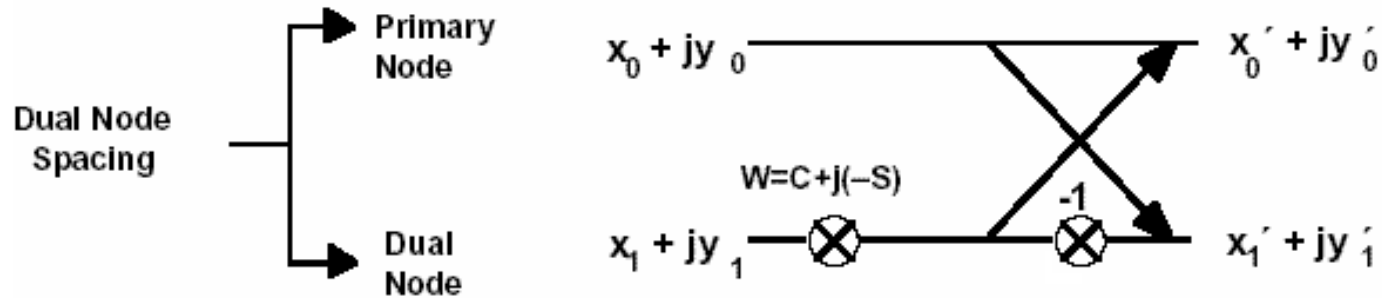
$$X(k + N/2) = Y(k) - W_N^k Z(k)$$

$$k = 0 \rightarrow N/2 - 1$$

Ezzel az N pontos algoritmust helyettesíthetjük két darab $N/2$ pontos algoritmussal

Egy alapelem (Butterfly) megvalósítása

Az elemek komplex mennyiségek:



$$x'_0 = x_0 + [(C)x_1 - (-S)y_1]$$

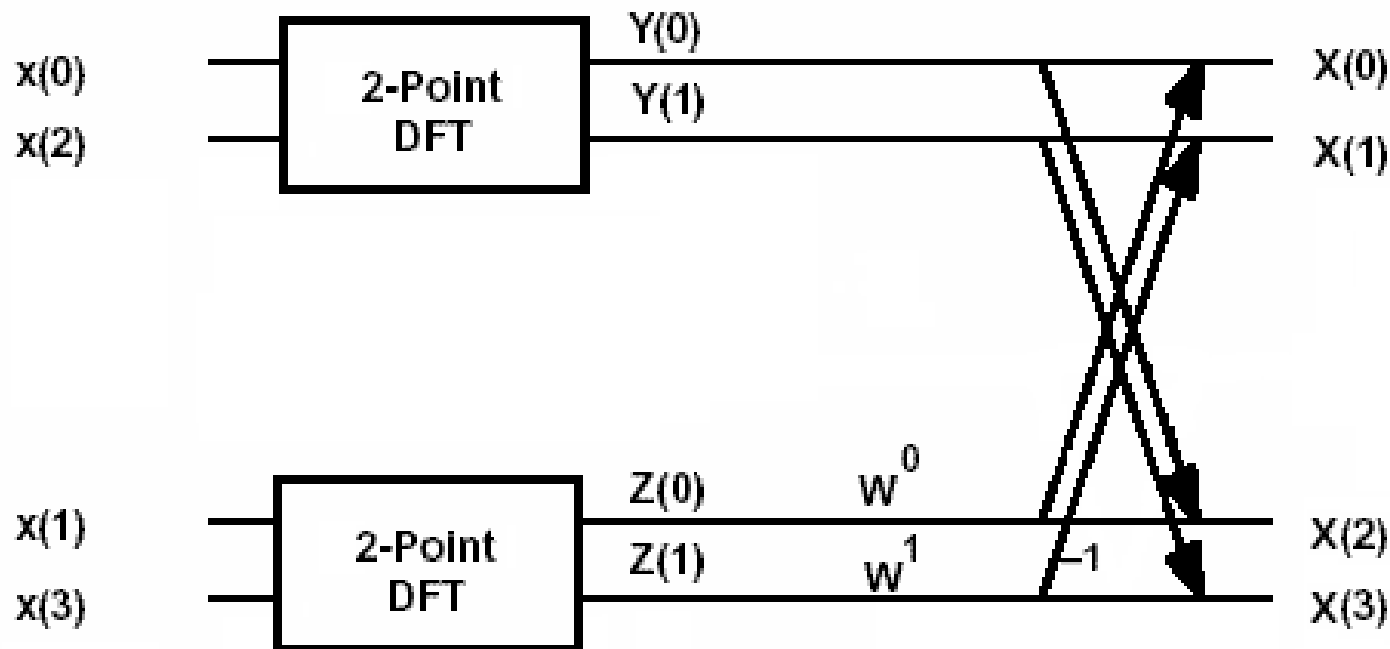
$$y'_0 = y_0 + [(C)y_1 + (-S)x_1]$$

$$x'_1 = x_0 - [(C)x_1 - (-S)y_1]$$

$$y'_1 = y_0 - [(C)y_1 + (-S)x_1]$$

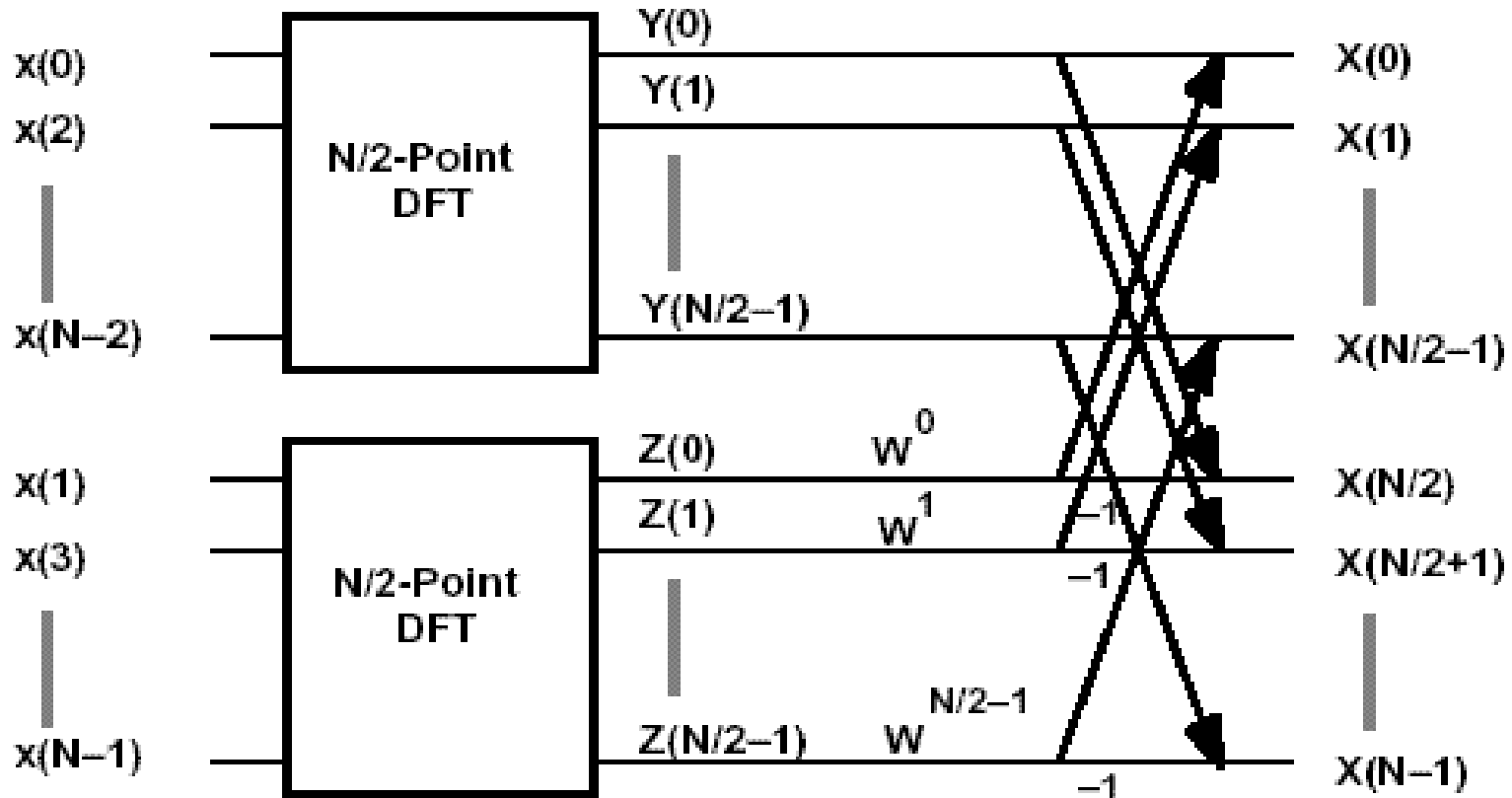
FFT algoritmus felbontása I.

Első lépésben veszek 2 db 2 pontos algoritmust, abból kapok egy 4 pontos kimenetet.



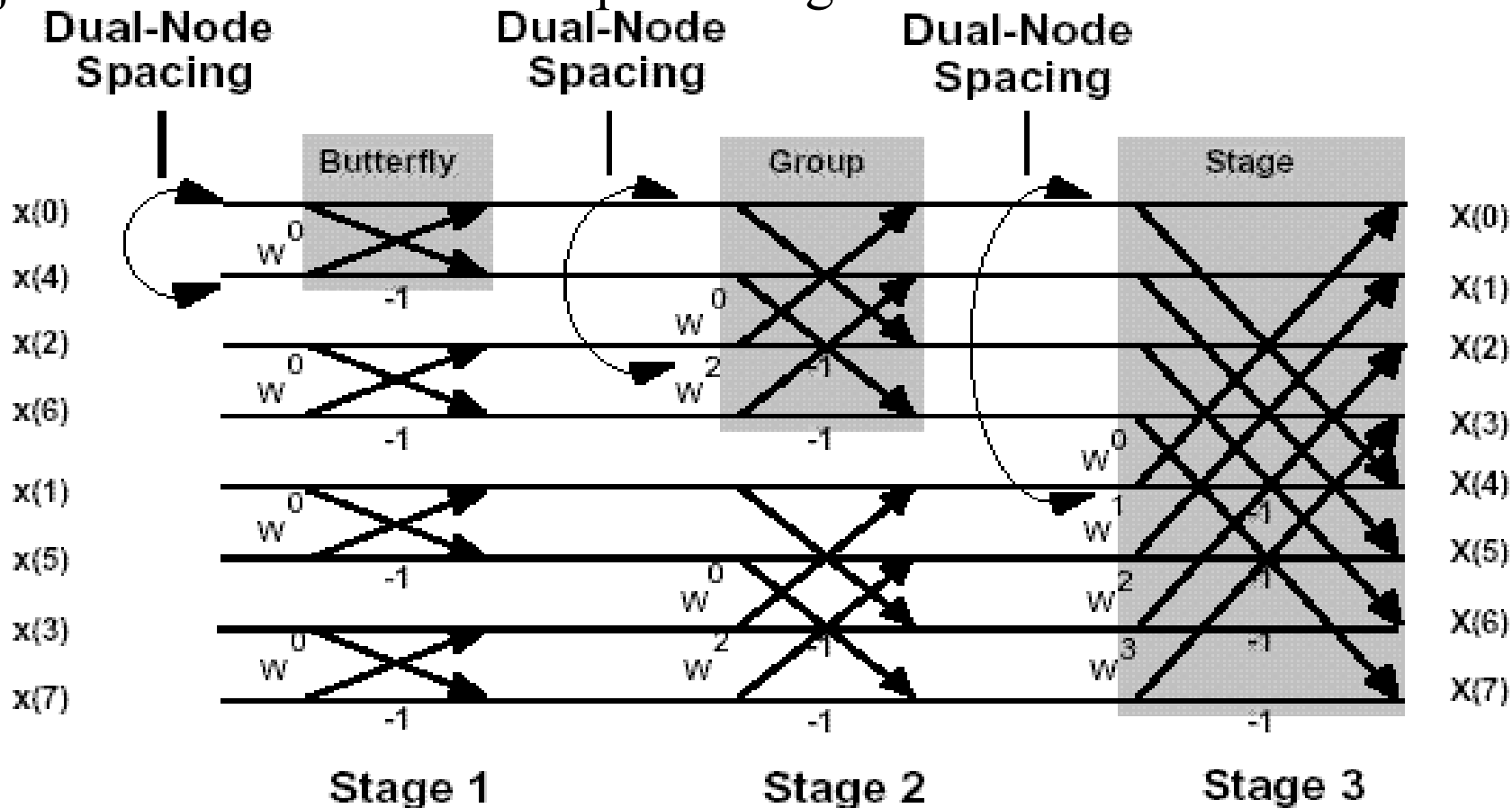
FFT algoritmus felbontása II.

Második lépésben veszek 2 db 4 pontos algoritmust, abból kapok egy 8 pontos kimenetet, és így tovább ...



8 pontos FFT algoritmus felbontása

A 8 pontos FFT algoritmusból először 2 db négy pontos algoritmus, majd tovább bontva 4 db két pontos algoritmus keletkezik.



A minták elrendezése

16 pontos FFT algoritmus felbontása és a minták bit fordított (bit reversed) elhelyezése az algoritmus bemenetén.

16 pontos FFT felbontása				bit fordított		normál	
0	0	0	0	0	0000	0	0000
1	2	4	8	8	1000	1	0001
2	4	8	4	4	0100	2	0010
3	6	C	C	C	1100	3	0011
4	8	2	2	2	0010	4	0100
5	A	6	A	A	1010	5	0101
6	C	A	6	6	0110	6	0110
7	E	E	E	E	1110	7	0111
8	1	1	1	1	0001	8	1000
9	3	5	9	9	1001	9	1001
A	5	9	5	5	0101	A	1010
B	7	D	D	D	1101	B	1011
C	9	3	3	3	0011	C	1100
D	B	7	B	B	1011	D	1101
E	D	B	7	7	0111	E	1110
F	F	F	F	F	1111	F	1111

Az N pontos FFT algoritmus megvalósítása

- 1. Az elemezni kívánt jelből N mintát veszünk, ahol $N=2^k$
- 2. A mintákat bit fordított elrendezésben elhelyezzük az N hosszúságú valós tárban.
- 3. A képzetes tárat feltöltjük 0-val.
- 4. Elvégezzük az $N/2$ darab komplex ‘butterfly’ kiszámítását az aktuális fokozatnak megfelelő módon.
- 5. Megismételjük a 4. pont szerinti számítást összesen k -szor.
- 6. Az eredmény valós és képzetes részét a két tárban kapjuk meg.
- 7. Az egyes spektrum összetevők abszolút értékét és fázisát kiszámítjuk a valós és képzetes részből.

FFT algoritmus hatékonysága a DFT algoritmushoz képest

Az FFT algoritmus főleg nagy mintaszám esetén hatékony.

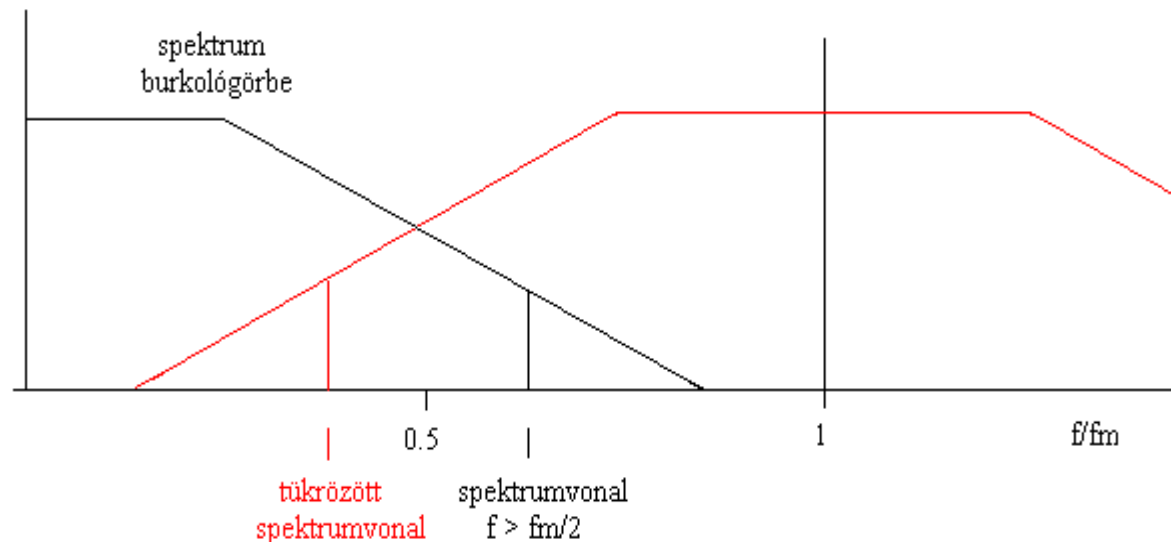
- ◆ DFT: N^2 Complex Multiplications
- ◆ FFT: $(N/2) \log_2(N)$ Complex Multiplications

N	DFT Multiplications	FFT Multiplications	FFT Efficiency
256	65,536	1,024	64 : 1
512	262,144	2,304	114 : 1
1,024	1,048,576	5,120	205 : 1
2,048	4,194,304	11,264	372 : 1
4,096	16,777,216	24,576	683 : 1

Aliasing jelenség

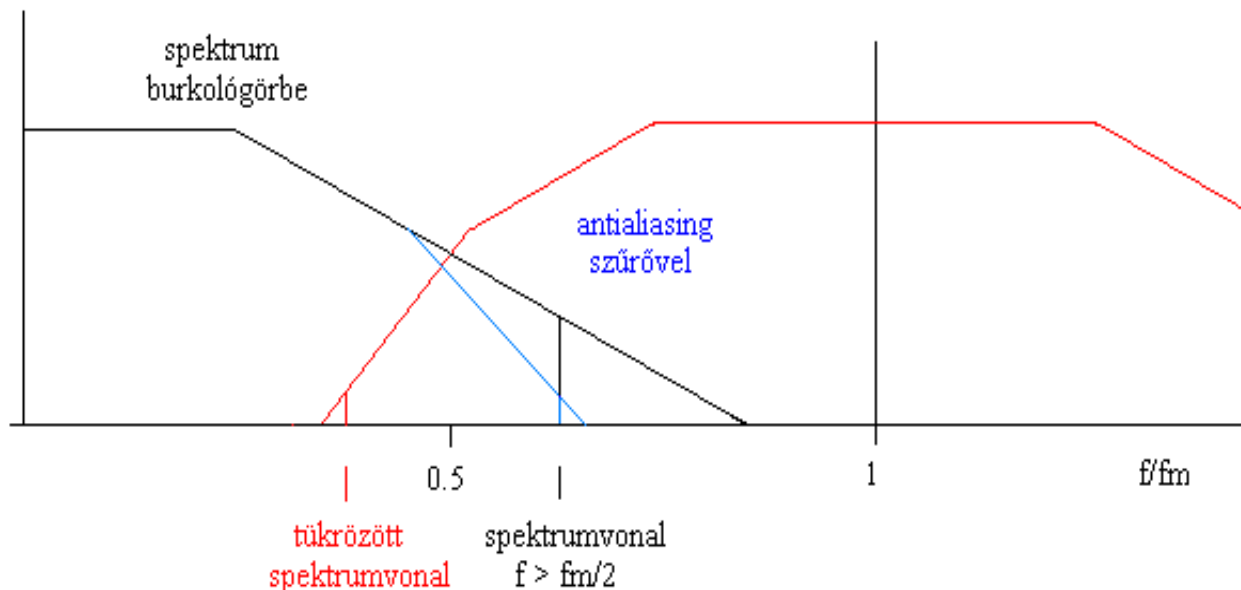
Ha a bemenő jel tartalmaz az $f_m/2$ -nél nagyobb frekvenciájú összetevőket, akkor a spektrumban megjelennek hamis vonalak.

Ha $f > f_m/2$, akkor az $f_m - f$ frekvencián is kapunk spektrum vonalat (az $f_m/2$ frekvenciára tükrözve)



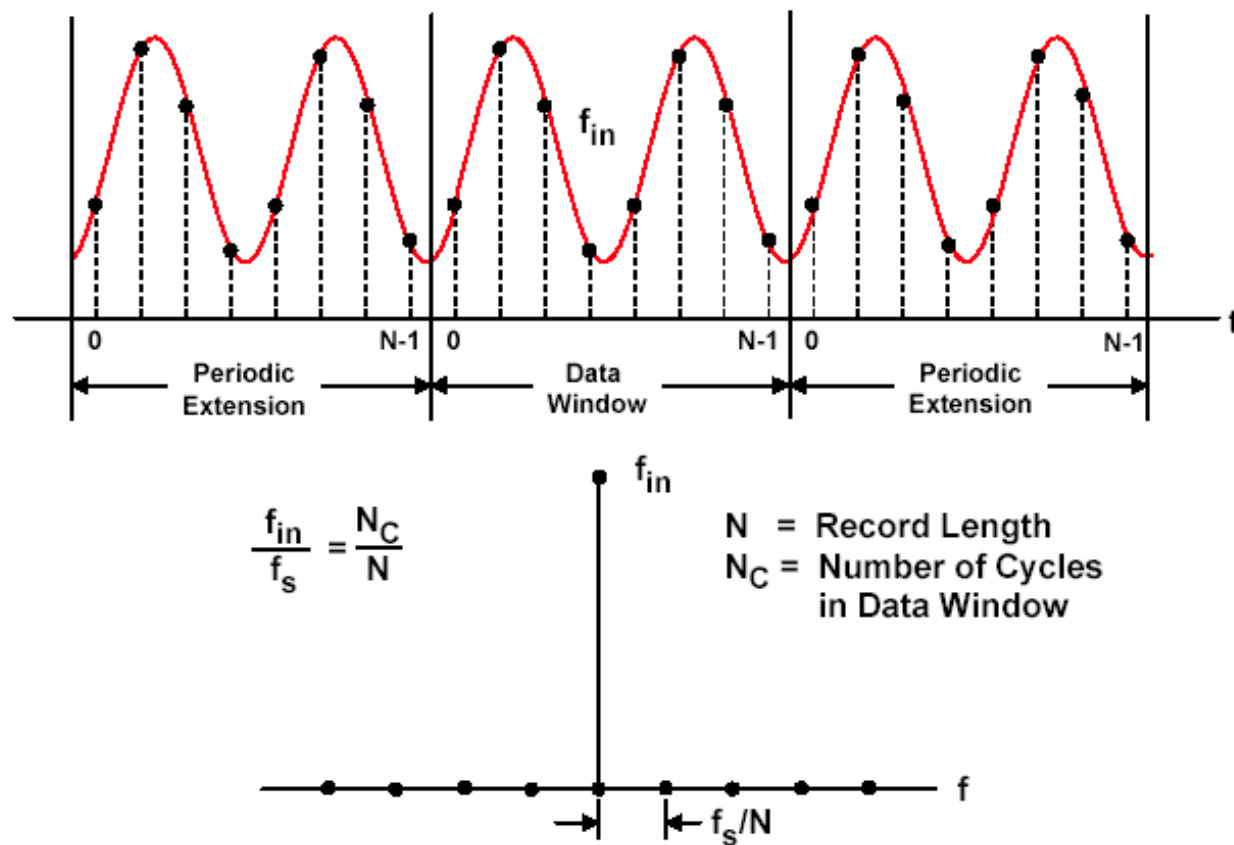
Aliasing jelenség korlátozása

1. Nagyobb mintavételi frekvencia alkalmazása, ha lehetséges.
2. A mintavételi frekvencia változtatásával megállapítható, hogy jelen vannak-e hibás spektrumvonalak.
3. Az $f_m/2$ -nél nagyobb összetevőket a bemenő jelből eltávolítjuk aluláteresztő szűrővel (antialiasing szűrő). Ez a szűrő természetesen csak az analóg oldalon, az AD konverter előtt lehet, és általában magas fokszámú szűrőket szokás alkalmazni.

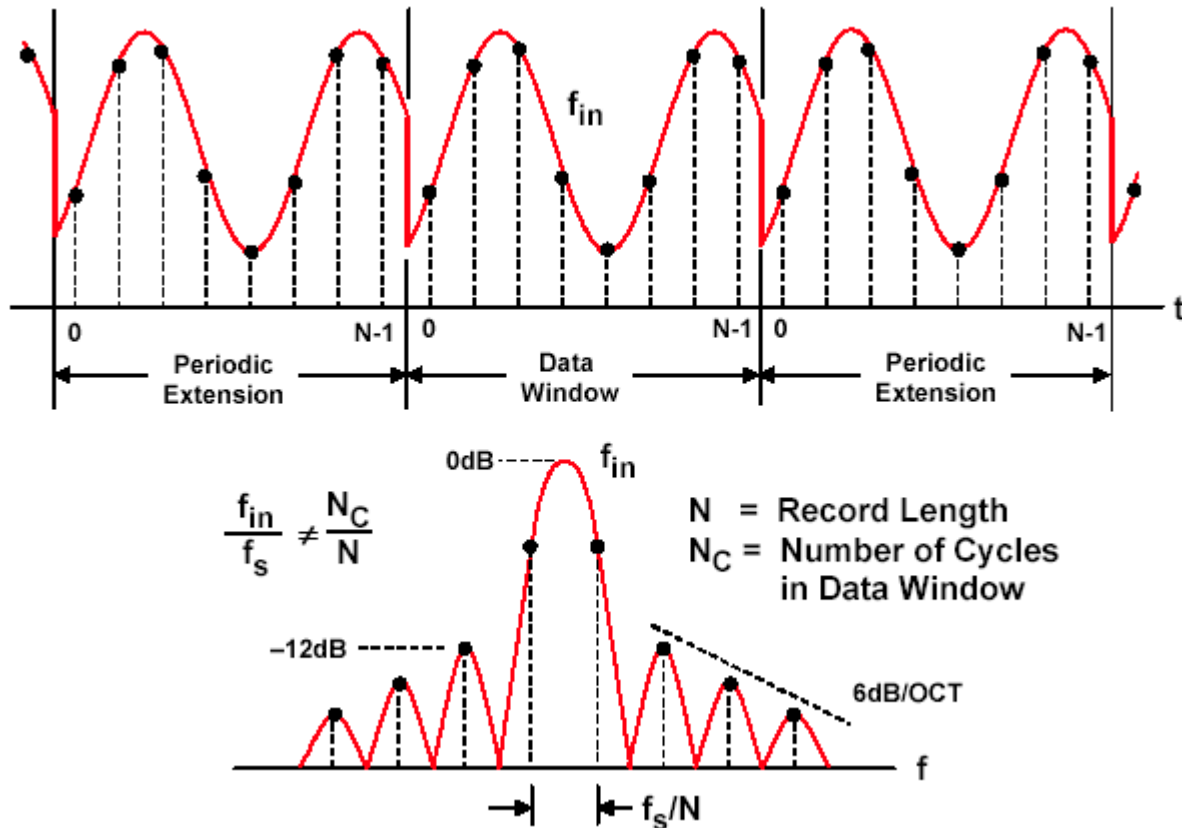


Mintavételezés nem egész számú periódusból

Ha szinuszos jelből egész számú periódusból veszünk mintát, akkor egy diszkrét spektrumvonalat kapunk:

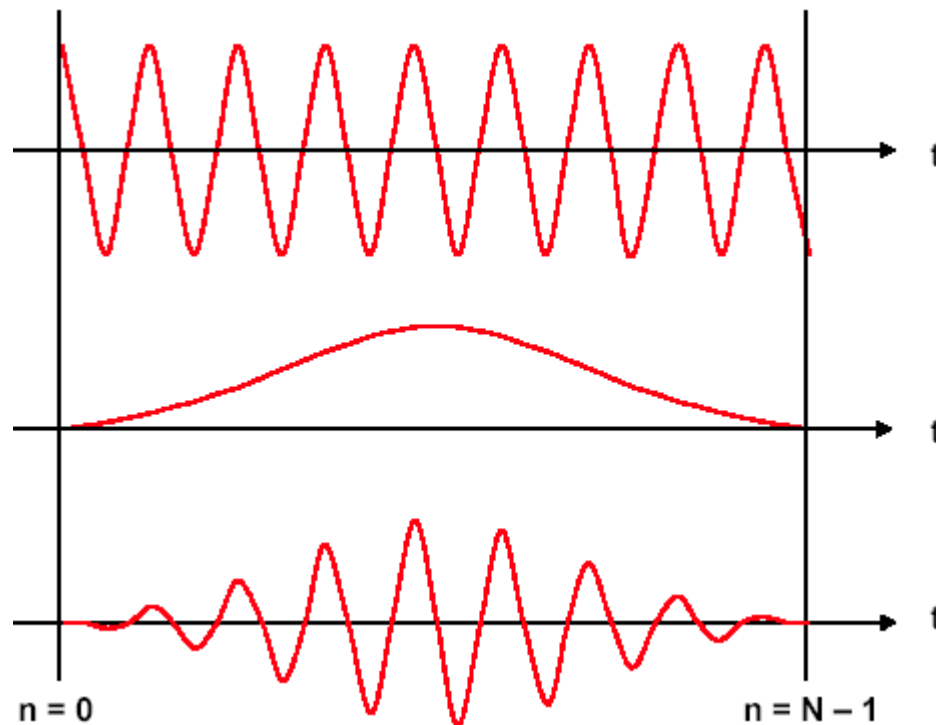


Ha a mintavétel hossza nem egész számú többszöröse a periódusnak, akkor a fő spektrumvonal mellett további mellék vonalak jelennek meg:



A mellék vonalak nagysága csökkenthető, ha a mintavételezett jelet egy célszerűen megválasztott függvénnyel az ún. ablakfüggvénnyel megszorozzuk a spektrumanalízis előtt.

Az ablakfüggvény a mintavett szakasz két végén 0 értékű, ezáltal nem jön létre ugrás a vizsgált időfüggvényben.



Néhány szokásos ablak függvény az alábbi

Megjegyezzük, hogy ha nem alkalmazunk ablakfüggvényt, ezt gyakran négyszög ablaknak nevezik.

$$\text{Hamming: } w(n) = 0.54 - 0.46 \cos \left[\frac{2\pi n}{N} \right]$$

$$\text{Blackman: } w(n) = 0.42 - 0.5 \cos \left[\frac{2\pi n}{N} \right] + 0.08 \cos \left[\frac{4\pi n}{N} \right]$$

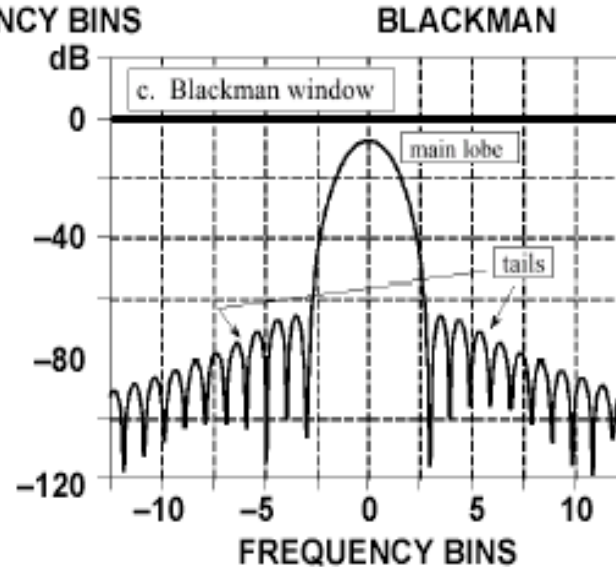
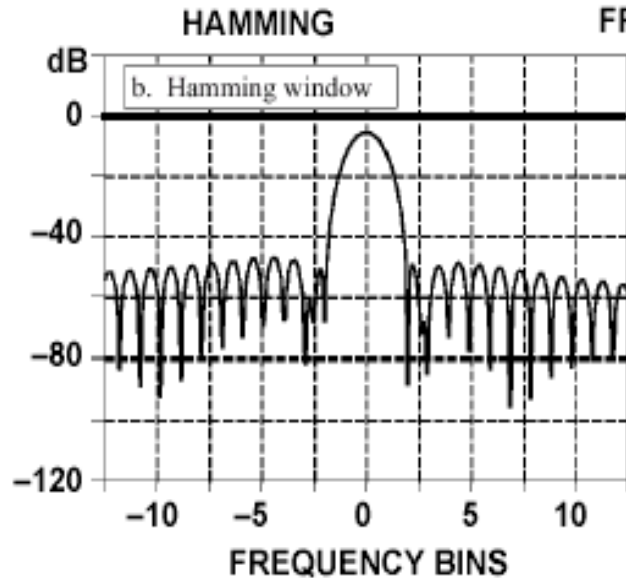
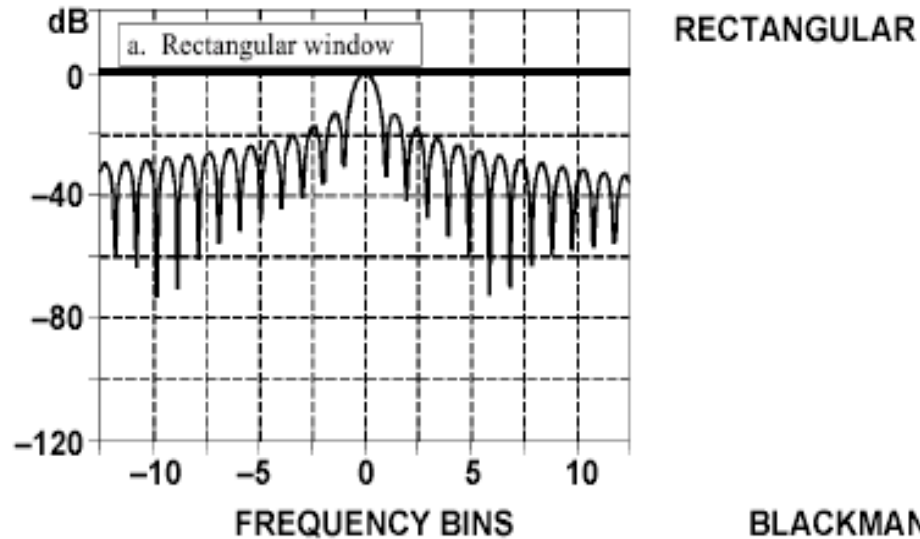
$$\text{Hanning: } w(n) = 0.5 - 0.5 \cos \left[\frac{2\pi n}{N} \right]$$

Az ablakfüggvénnyel a mintavett jeleket megszorozzuk:

$$x'(n) = x(n) * w(n)$$

Oldalsávok elnyomása ablakfüggvény alkalmazásakor

$$\text{BIN WIDTH} = \frac{f_s}{N}$$



Hálózati frekvencia mérése

A hálózati frekvencia értéke

A hálózati frekvencia névleges értéke (nálunk) 50 Hz. A frekvencia pillanatnyi értéke folyamatosan változik a rendszer terhelésének függvényében.

Az MSZ EN 50160 szabvány szerint a frekvencia megengedett értékei $50 \text{ Hz} \pm 1\%$ az idő 99.5 %-ában és $50 \text{ Hz} +4\%/-6\%$ az idő 100 %-ában.

A gyakorlatban jelenleg a hálózati frekvencia általában az $50 \text{ Hz} \pm 50 \text{ mHz} (\pm 0.1\%)$ értéken belül van.

A frekvencia meghatározásának módjai

1. A frekvencia pillanatértéke a fázis változásának differenciál hányadosa:

$$f(t) = \frac{1}{2\pi} * \frac{d\varphi(t)}{dt}$$

A fázis időfüggvényének mérése nehezen megvalósítható, ezért ez a módszer nem használható.

2. Ha a fázis változását egy periódus ideig lineárisnak tekintjük, akkor a frekvencia ez alatt a periódus alatt:

$$f = \frac{1}{T}$$

ahol T egy periódus ideje, amely két azonos irányú nulla átmenet között eltelt időként határozható meg.

3. A frekvencia az 1 másodperc alatti periódusok száma.

Alacsony frekvencián (50 Hz) a mérési hiba (± 1 Hz) igen nagy.

A hálózati frekvencia változása

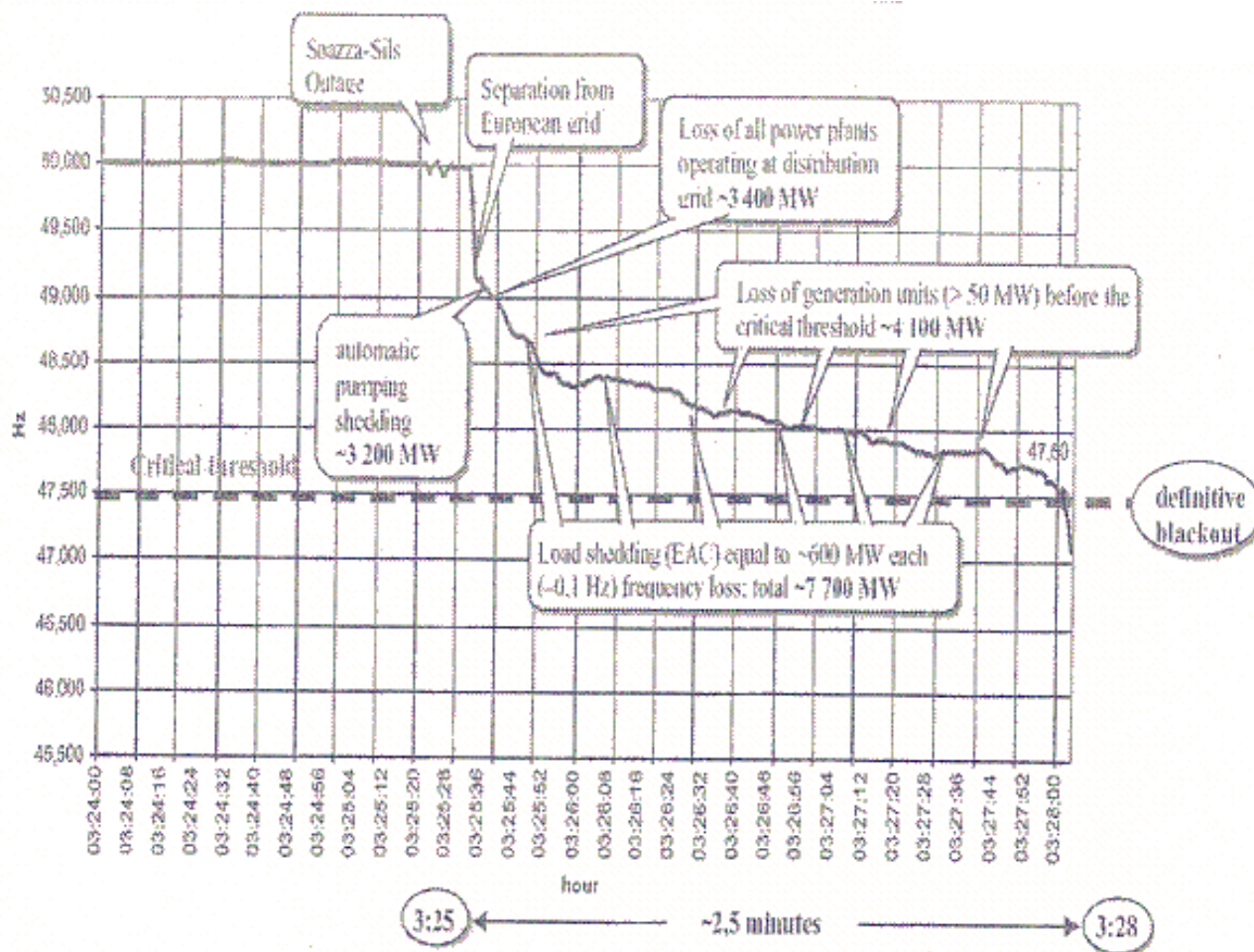
A rendszer teljesítmény egyensúlyának gyors változása a hálózati frekvencia gyors változását eredményezi. A alábbi ábrán a rendszerben hirtelen fellépő kb. 1000 MW erőmű kiesés hatására fellépő frekvencia változás látható:



<http://www.mavir.hu> – adatpublikáció – Aktuális VER adatok - Frekvencia

http://portal.mavir.hu/portal/page/portal/Mavir/Adatpublikacio/VER_aktualis_adatok/HZGadatok

Az alábbi ábrán a 2003. szeptember 28-án az olasz rendszer kiesésekor regisztrált frekvencia változás látható. A rendszer teljes kiesésének pillanatában a frekvencia értéke 47.5 Hz volt. (Elektrotechnika 2005/3)



A frekvencia mérésének módszerei

- Analóg mérési módszerek:

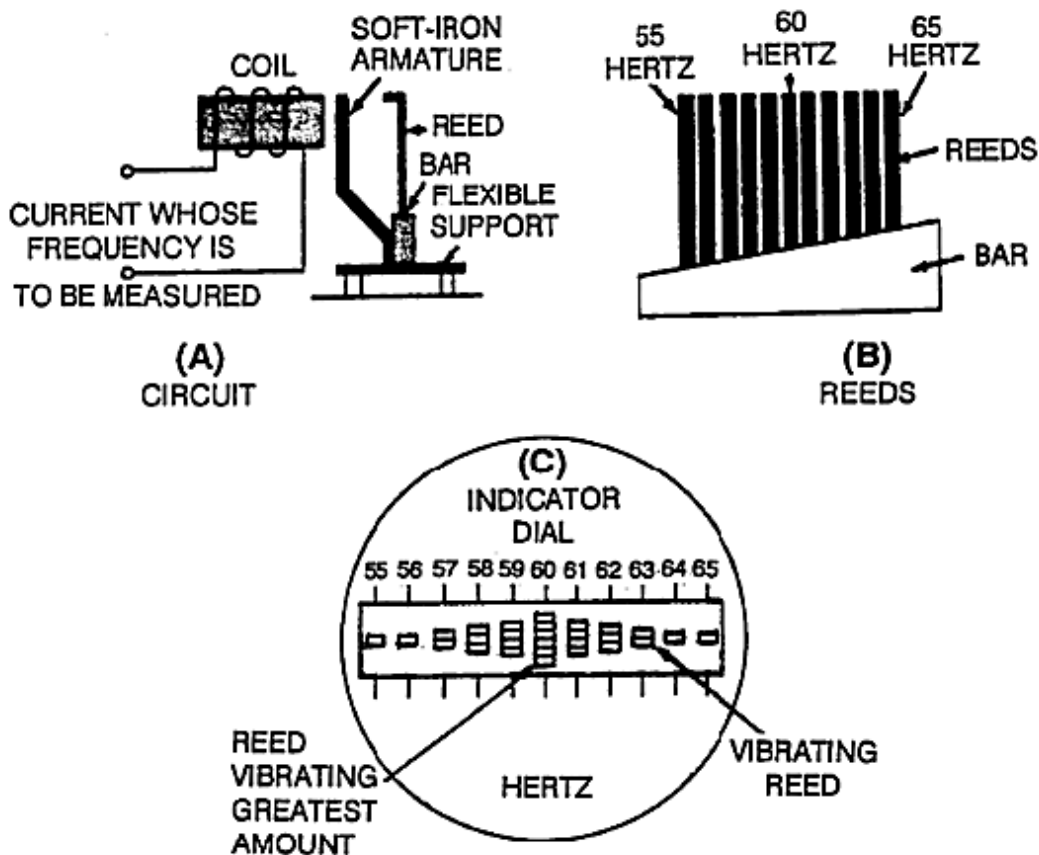
- rezgőnyelves műszer
- mozgótárcsás műszer
- sávszűrő alkalmazása

- Digitális mérési módszerek:

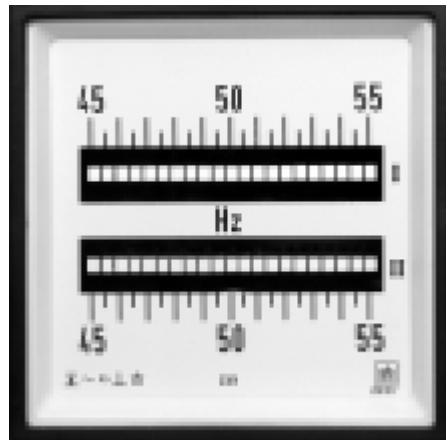
- számlálók használata
- mikrokontroller alkalmazása

Rezgőnyelvényes frekvenciamérő

Egyszerű frekvenciamérő műszer. A mérendő frekvenciájú árammal gerjesztett tekercs kétszeres frekvenciával rezgeti a lágyvasas armatúrát, amely össze van kötve több, különböző rezonanciafrekvenciára hangolt lemezzel. Az a lemez fog a legnagyobb amplitúdóval rezegni, amelynek a frekvenciája megegyezik a hálózati frekvencia kétszeresével.



Egy és kétféle rezgőnyelves frekvenciamérő táblaműszer



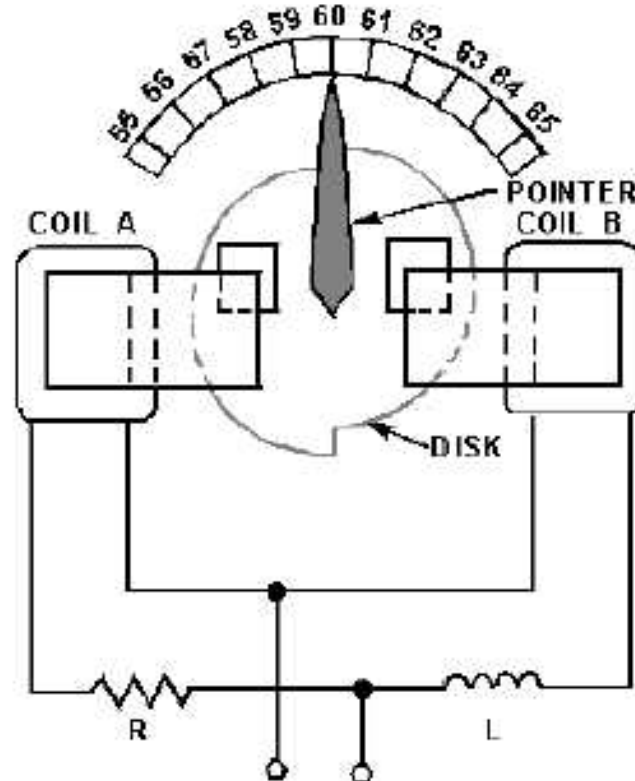
Rezgőnyelves frekvenciamérő kézi műszer:

A műszerek 50 vagy 60 Hz középfrekvenciával, ± 5 Hz méréshatárral és 0.5 Hz felbontással rendelkeznek.



Mozgó tárcsás frekvenciamérő

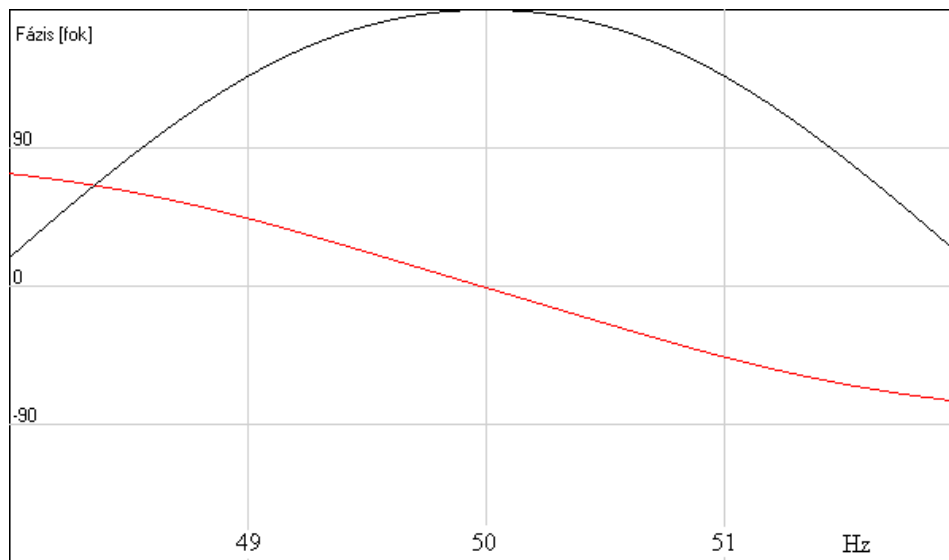
Névleges frekvencia esetén az A és B tekercs gerjesztése azonos és a tárcsa nyugalomban van. Ha a frekvencia növekszik, akkor a B tekercs gerjesztése csökken, ezért a tárcsa forogni kezd balra. Ekkor a tárcsán lévő kivágás az A tekercs alá kerül és új egyensúlyi állapot alakul ki.



Analóg mérési módszer sávszűrővel (1.)

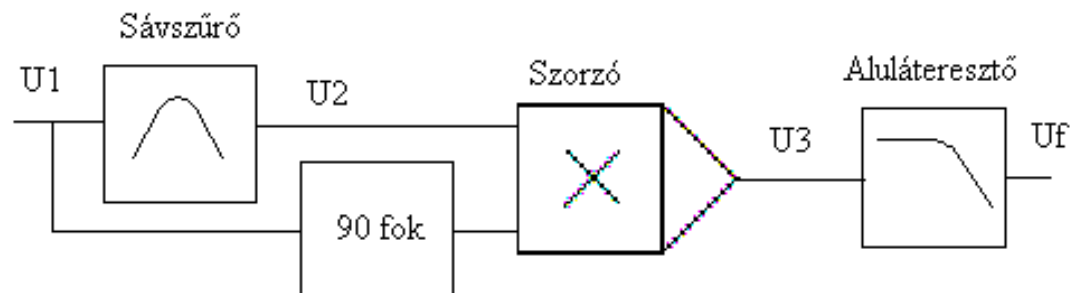
A mérendő jelet hálózati frekvenciára hangolt sávszűrőn vezetjük át. A sávszűrő amplitudó görbéje a rezonancia frekvencia környékén lapos, a fázismenet azonban közel lineáris, ezért a fázis változását használhatjuk ki a frekvencia meghatározására.

Az ábrán 50 Hz-es, $Q=25$ jósági tényezőjű sávszűrő amplitudó és fázis karakterisztikája látható:



Analóg mérési módszer sávszűrővel (2.)

A fázis meghatározása szorzó áramkört tartalmazó fázis demodulátorral történhet.



A sávszűrő fázistolása a mérendő frekvencián φ . A szorzó kimenete:

$$U_3(t) = U_1 \sin(\omega t + \frac{\pi}{2}) * U_2 \sin(\omega t + \varphi) = \frac{1}{2} U_1 U_2 (\cos(\frac{\pi}{2} - \varphi) - \cos(2\omega t + \varphi))$$

Az aluláteresztő kimenetén a fázistolással arányos feszültség jelenik meg:

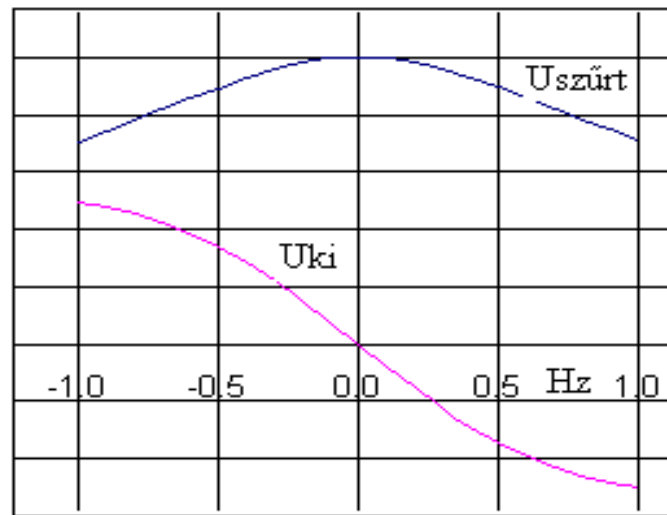
$$U_f = \frac{1}{2} U_1 U_2 \cos(\frac{\pi}{2} - \varphi) = \frac{1}{2} U_1 U_2 \sin(\varphi)$$

A frekvencia és a fázistolás közötti összefüggés a sávszűrő jósági tényezőjétől függ:

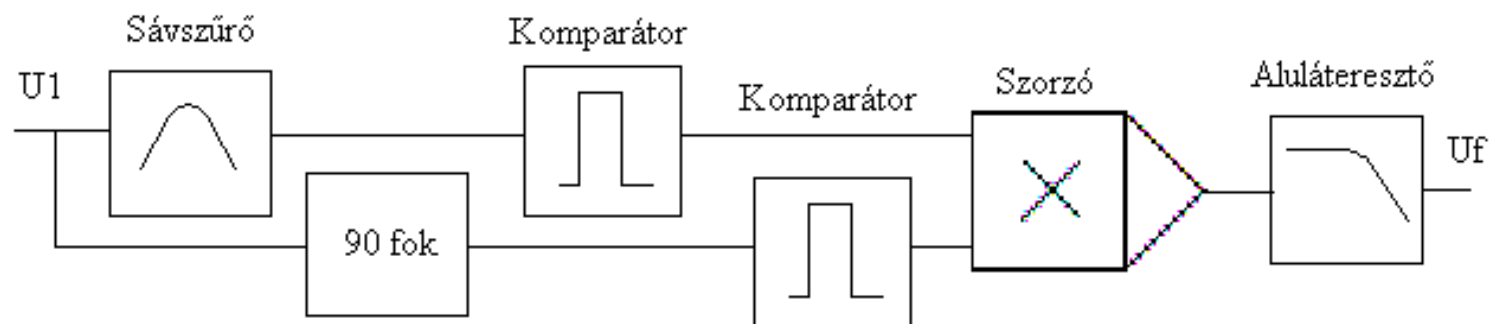
$$df = -\frac{f_0}{2Q} \operatorname{tg}(\varphi)$$

Analóg mérési módszer sávszűrővel (3.)

A sávszűrő amplitúdó karakterisztikájának hatására a kimenő feszültség-frekvencia karakterisztika nemlineáris lesz:



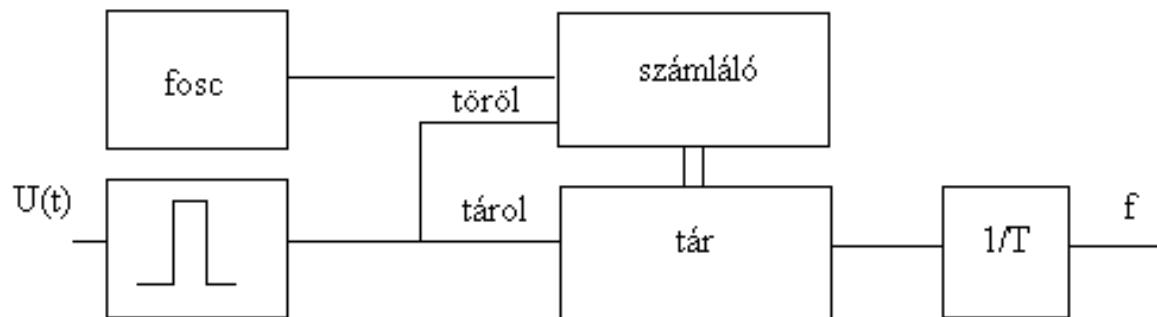
Az amplitúdó változás hatását kiküszöbölhetjük a jelek állandó amplitúdójú négyszögjellel alakításával.



Digitális mérési módszerek (1.)

A digitális mérési módszerek a periódus idő mérésén alapulnak. Ekkor periódusonként egy mérési értéket kapunk.

A mérés elvi módszere az, hogy a mérendő jel egy periódusából kapujelet képezünk, amely egy $f_{osc} \gg f$ frekvenciájú oszcillátor jelét számláló számlálót vezérel. A számlálót minden periódus kezdetén nullázza, a periódus végén pedig tárolja az értékét. A frekvenciát a periódus hosszából reciprok képző algoritmus számítja ki.



Digitális mérési módszerek (2.)

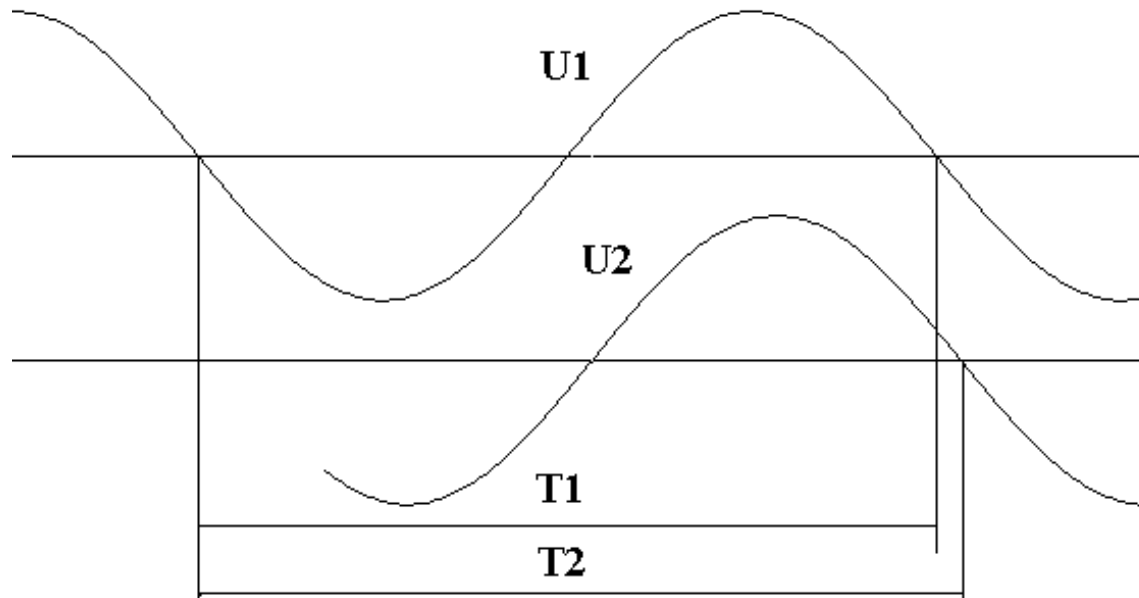
Az oszcillátor specifikációja és a számláló hossza a frekvencia mérés kívánt felbontásától függ:

Ha a		Akkor	
felbontás	pontosság	fosc	számláló
mHz	ppm	kHz	bit
100	2000	25	9
50	1000	50	10
10	200	250	13
5	100	500	14
1	20	2500	16

Az általánosan használt kvarc oszcillátorok pontossága ± 100 ppm, de a ± 20 ppm pontosságú oszcillátor sem különleges, a ± 1 ppm pontosságúak már jóval drágábbak, de beszerezhetőek.

A periódusidő meghatározása

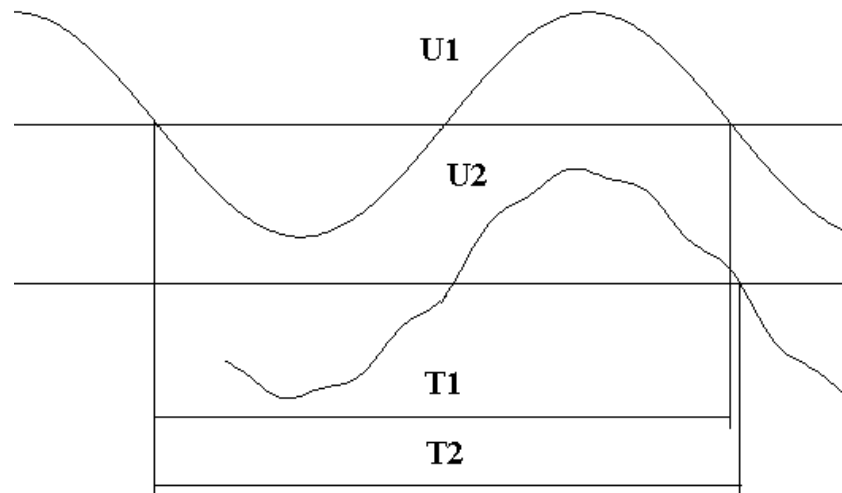
A periódusidő elvileg a feszültség két azonos irányú nulla átmenete közötti idő. A nulla átmeneteket azonban megváltoztatja a terhelések és a harmonikusok gyors változásai. Ha a terhelés az U_1 feszültség nulla átmenete után megváltozik, a feszültség U_2 helyett U_2 lesz (fázisugrás jön létre), emiatt T_1 helyett T_2 periódus időt mérünk.



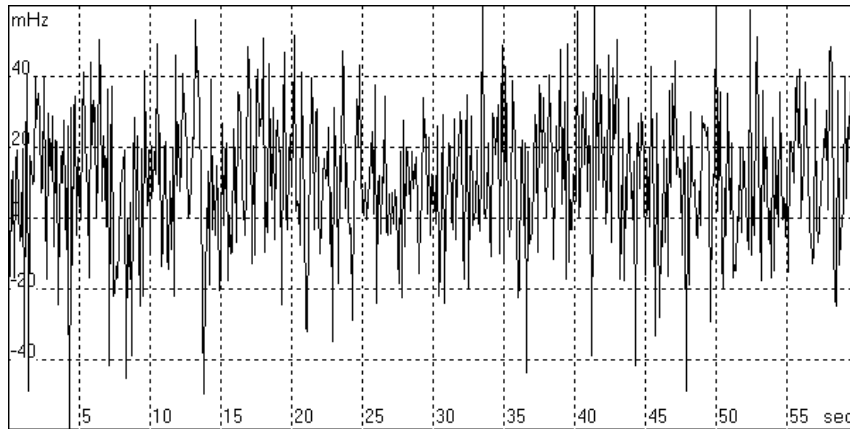
Egy leágazás névleges teljesítményének megváltozásakor keletkező frekvenciamérési hiba:

ΔP	ΔT	Δf
%	ppm	mHz
10	1 586	79
1	159	8
0,1	16	1

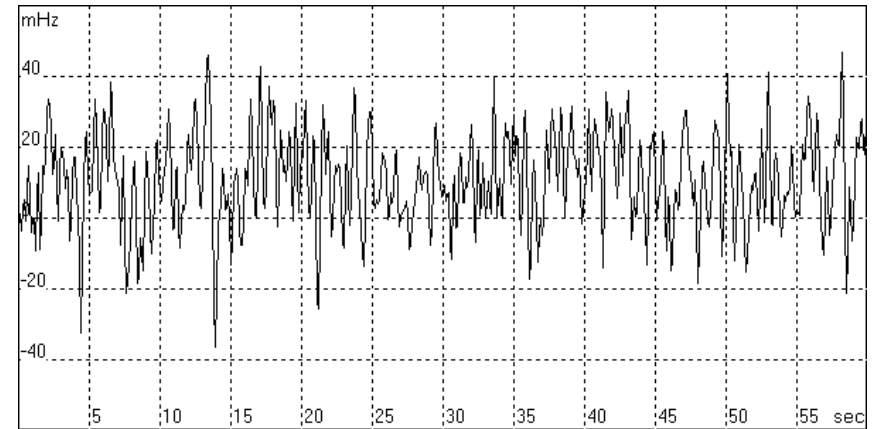
A harmonikus tartalom hirtelen megváltozása is a mért periódus idő megváltozását okozza.



Az alábbi ábrákon egy 0.4 kV-os hálózaton mért frekvencia időfüggvény látható különböző f_0 határfrekvenciájú aluláteresztő szűrők felhasználásával:



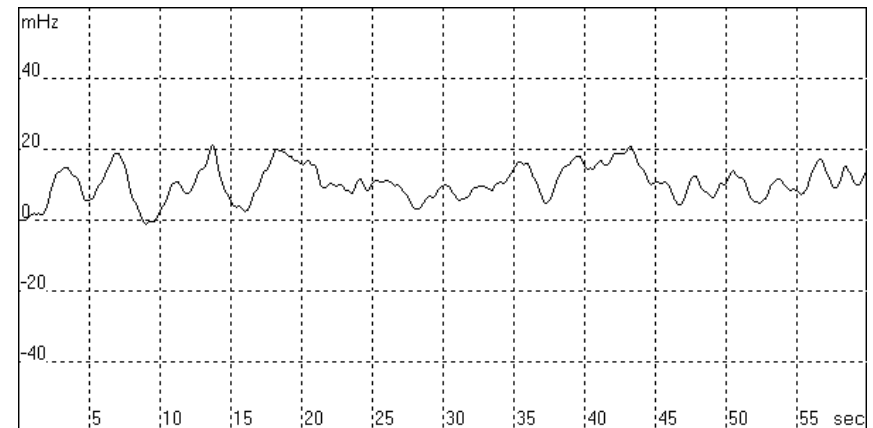
$f_0=10$ Hz



$f_0=2$ Hz



$f_0=0.5$ Hz

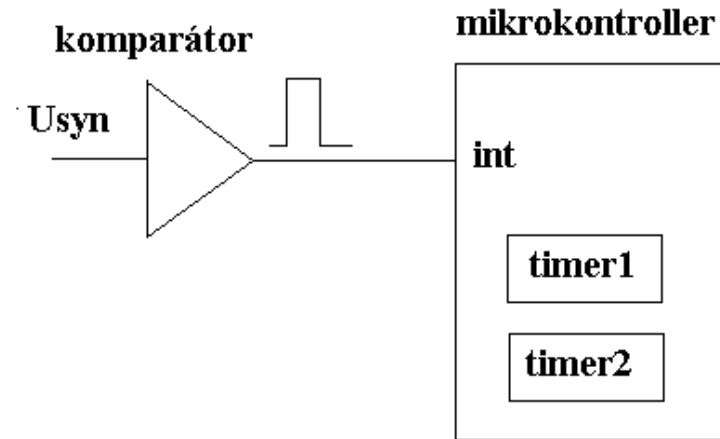


$f_0=0.2$ Hz

Frekvencia mérés mikrokontroller alkalmazásával

A frekvencia mérésére mikrokontrollert célszerű alkalmazni.

A referencia feszültséget komparátorral négyszögjellé alakítjuk, amely minden pozitív élnél megszakítást ad. A megszakításkor a timer1 számlálót kiolvassuk



A megszakítási rutin:

- megszakítás tiltás

- timer1 kiolvasása

- timer1 nullázása

- timer2 indítása 15-18 msec időzítéssel

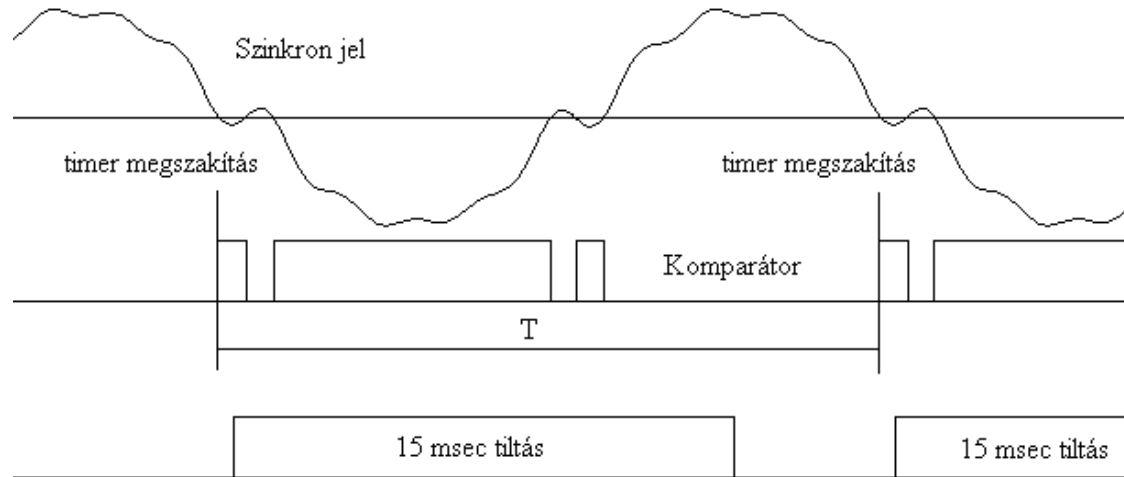
- frekvencia számítása, digitális szűrés

Timer2 rutin:

- timer2 leáll

- megszakítás engedélyezése

A timer2 feladata az, hogy a szinkron jelben a harmonikusok hatására fellépő többszörös nulla átmenetek által okozott hibát kiküszöbölje.



Frekvencia mérő algoritmus az IEC 61000-4-30 szerint A osztályú készülékek esetén:

10 másodperces mérési idő alatti egész periódusok számát és ezek teljes idejét mérjük

$$f = \frac{n}{T_n} \quad \text{ahol } n \text{ az egész periódusok száma}$$

$$T_n \text{ ezek teljes ideje}$$

Feszültség minőségének mérése

(Power Quality)

Feszültség minőségi mennyiségének mérése

A méréseket az IEC 61000-4-30 szabvány előírásai szerint kell elvégezni.

A mérendő mennyiségek:

- Frekvencia
- Feszültség
- Flicker
- Dip és swell
- Feszültség kimaradás
- Negatív sorrendű összetevő
- Harmonikusok

Mérési idő tartományok

B osztályú készülékeknél nincs előírás.

A osztályú készülékeknél a méréseket négy különböző idő tartományban kell elvégezni:

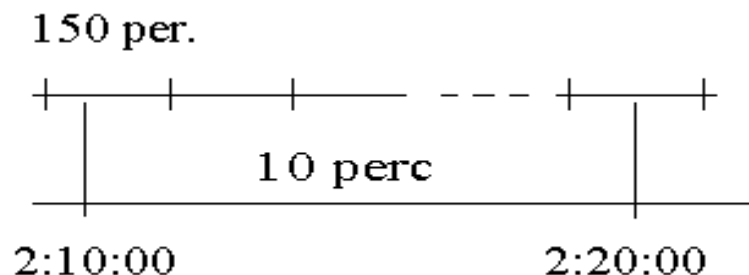
- 10/12 periódus 50/60 Hz esetén (200 ms, alapegység)
- 150/180 periódus 50/60 Hz esetén (very short time, 3 sec)
- 10 perc (short time)
- 2 óra (long time)

1. Az alapegység idő a 10 periódus. Ha valamely okból a mérés hiányzik vagy nem értelmezhető az adott mérésnél, az időtartományt meg kell jelölni (flagged).

2. A 150 periódus 15 db 10 periódus idejű mérésből áll. Ha ezek közül több mint 7 db 'flagged', akkor az egész időtartam is 'flagged'.

3. A 10 perces mérési ciklus a real time óra egész 10 perces értékénél kezdődik és tartalmazza az összes olyan 150 periódus alatt mért adatot, amely legalább részben beleesik a 10 perces intervallumba. A szélen lévő adatokat arányos súlyozással kell figyelembe venni.

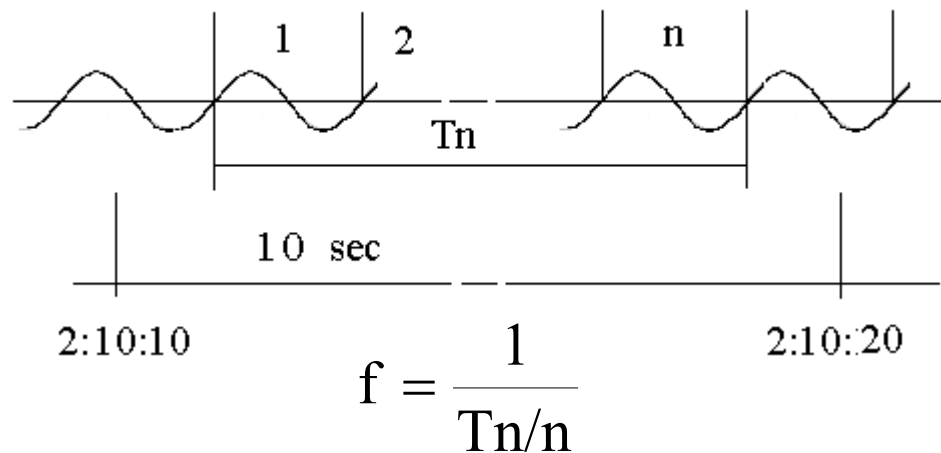
Ha a 150 periódusoknak több mint a fele 'flagged', akkor az egész 10 perces ciklus 'flagged' lesz.



4. A két órás mérési ciklus a real time óra egész 2 órás értékénél kezdődik és 12 db 10 perces ciklust tartalmaz. Ha több mint 6 db 10 perces időtartam 'flagged', akkor az egész 2 órás tartomány 'flagged' lesz.

Hálózati frekvencia mérése

A hálózati frekvencia mérése a real time idő kerek 10 másodperces időpillanata utáni a referencia fázis feszültség első pozitív nulla átmenetnél kezdődik és tartalmazza az összes egész periódust.



A frekvencia mérésének pontossága:

A osztály esetén $\Delta f < 10$ mHz,

B osztály esetén $\Delta f < 100$ mHz

Ha a mérés során dip (csökkenés), swell (növekedés), vagy feszültség kimaradás lép fel, a mérés 'flagged' lesz.

Feszültség effektív érték mérése

A mérést 10 periódus hosszan mérjük az alábbi képlet szerint:

$$U_{\text{rms_200msec}} = \sqrt{\frac{\sum_{1}^N U^2}{N}} \quad \text{ahol } N \text{ a minták száma}$$

A mérés pontossága A osztály esetén $\pm 0.1\%$

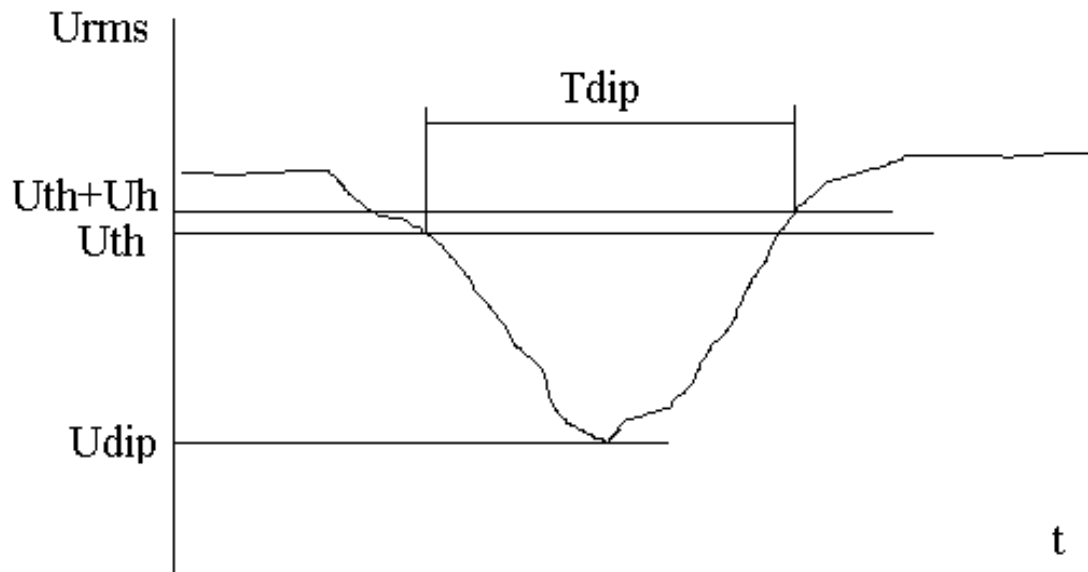
Ha a mért érték bármelyik fázisban $> 1.5U_n$ vagy $< 0.5U_n$ akkor az egész 3 fázisú mérés 'flagged' lesz.

A hosszabb idejű méréseket a megadott módon képezhetjük.

Dip és swell mérése

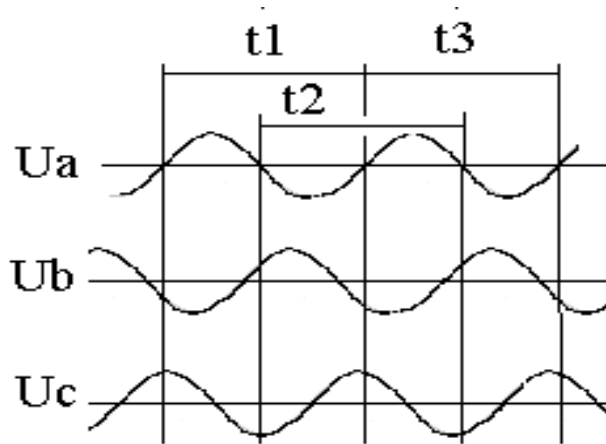
A dip rövid idejű feszültség csökkenés, a swell pedig rövid idejű feszültség növekedés.

A dip akkor kezdődik, amikor a feszültség egy meghatározott küszöbérték alá csökken és addig tart, amíg a küszöb + hiszterézis érték alatt marad és értéke a minimális feszültség.



Eredmény egy számpár, U_{dip} és T_{dip}

A feszültség mérése egy periódus ideig tart, félperiódusonként ismételve. A periódus kezdete mindegyik fázisban a referencia fázis pozitív nulla átmenete.



Az effektív érték számítása:

$$U_{\text{rms}} = \sqrt{\frac{\sum_{1}^N U^2}{N}}$$

A mintavételi frekvenciát az utolsó érvényes frekvencia mérés felhasználásával kell megállapítani.

Háromfázisú rendszerben a dip addig tart, amíg bármelyik fázisban a küszöbérték alatti, de $0.1U_n$ feletti értéket mérünk.

A mérés pontossága:

A osztály esetén $\pm 0.2\% U_n$,

B osztály esetén $\pm 2\% U_n$

Feszültség kimaradás mérése

A feszültség kimaradás mérése a dip mérésével azonos, de a feszültség a névleges érték 10%-a alá csökken.

A feszültség kimaradás esetén a nulla átmeneteket az eddigi periódus idő segítségével határozhatjuk meg.

Flicker mérése

A flickert az IEC 61000-4-15 szerint mérjük.

Harmonikusok mérése

A harmonikusokat az IEC 61000-4-7 szerint mérjük.

A méréseket az előzőekben meghatározott időtartamokra kell elvégezni. Ha dip, swell vagy kimaradás lép fel, akkor a mérés 'flagged' lesz.

Negatív sorrend mérése

A negatív sorrendű összetevőt a vonali feszültségek mérése segítségével határozzuk meg az alábbi képlet szerint:

$$\frac{|U_2|}{|U_1|} = \sqrt{\frac{1 - \sqrt{3 - 6\beta}}{1 + \sqrt{3 - 6\beta}}}$$
$$\beta = \frac{U_{ab}^4 + U_{bc}^4 + U_{ca}^4}{(U_{ab}^2 + U_{bc}^2 + U_{ca}^2)^2}$$

A vonali feszültségek alapharmonikus értékét kell használni.

A méréseket az előzőekben meghatározott időtartamokra kell elvégezni. Ha dip, swell vagy kimaradás lép fel, akkor a mérés ‘flagged’ lesz.

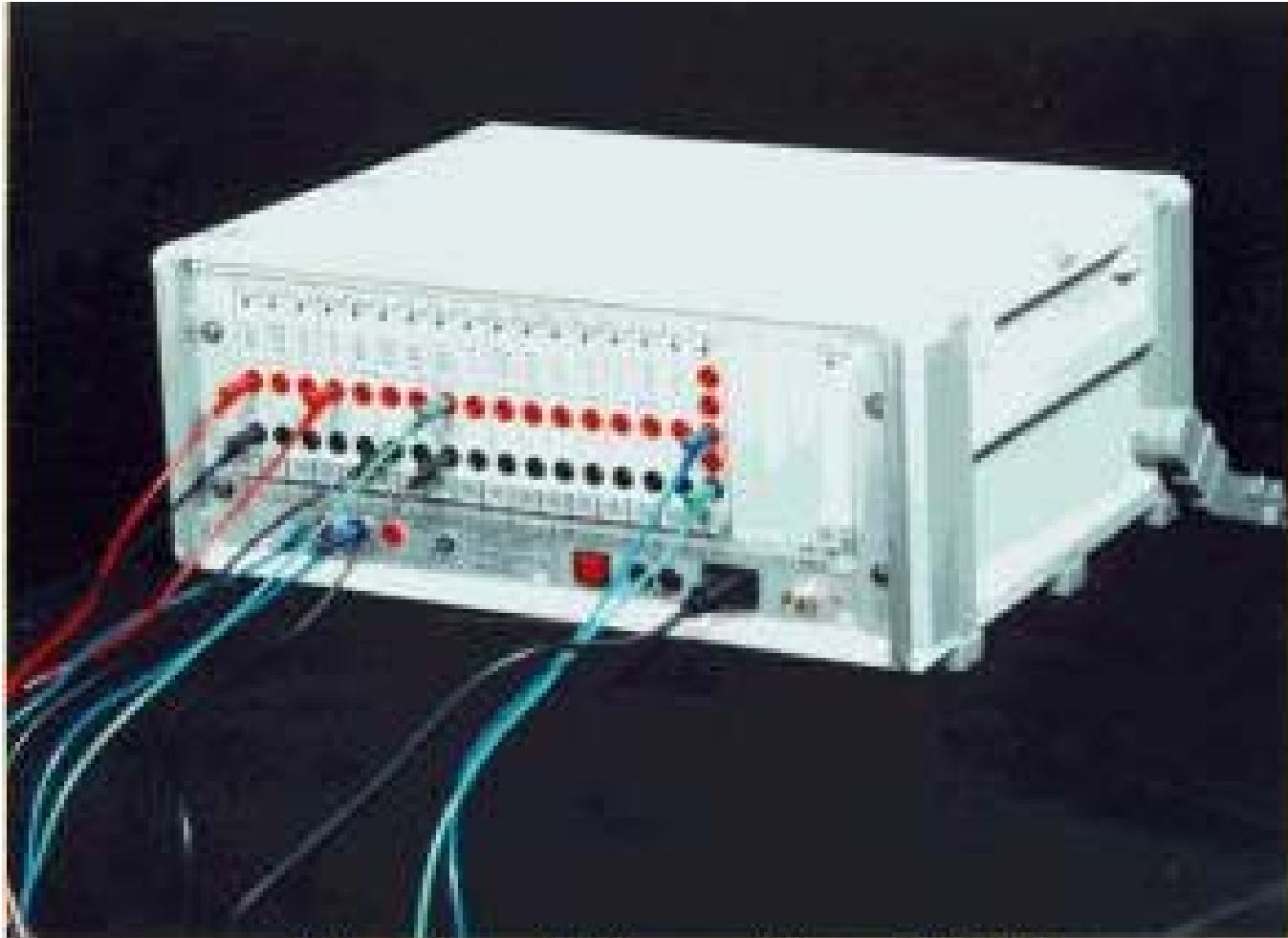
A mérés megkívánt pontossága:

A osztály esetén 0.2%, B osztály esetén 0.5%

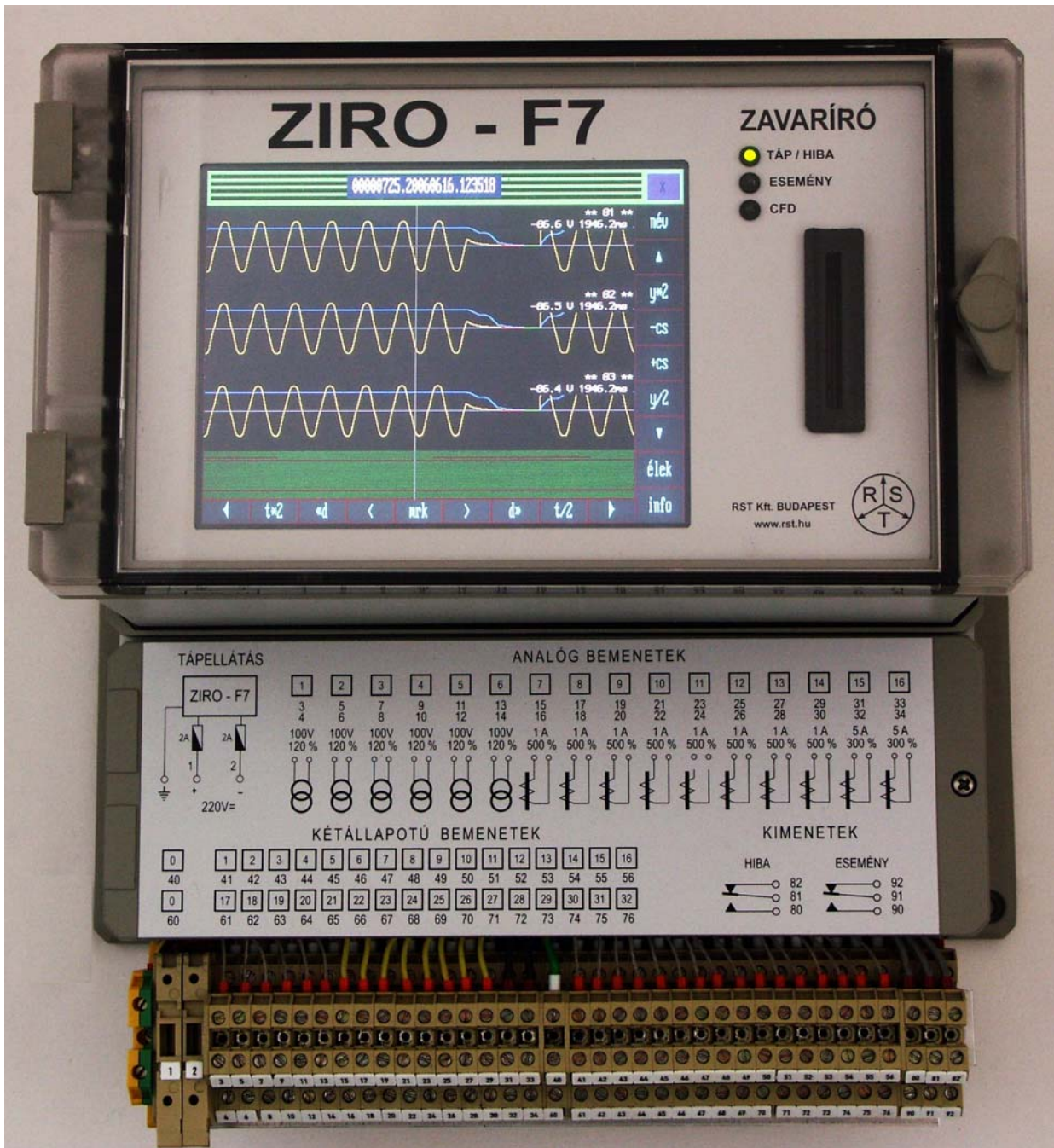
TR-16 Multirec

16 Channel Multifunction Power System Analyzer





Signal conditioner side of TR-16



Elektronikus berendezések túlfeszültség és zavarvédelme

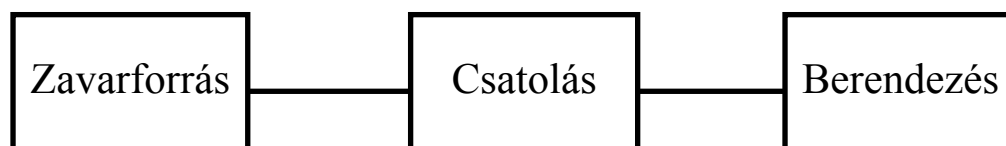
Irodalom: Dán András, Tersztyánszky Tibor, Varjú György

Villamosenergia minőség

Kiadó: INVEST-MARKETING Bt.

Az elektronikus berendezésekben külső zavarjelek hatására működési zavarok léphetnek fel, egyes esetekben ez a berendezés meghibásodásához is vezethet.

A zavarok a keletkezési helyükről csatolás útján jutnak el a berendezéshez.



Zavarforrás: villámcsapás, kapcsolási túlfeszültség, nagy áramok, harmonikusok ...
rádiófrekvenciás adók (mobil telefonok)...

Csatolás: kapacitív, induktív, ohmos

Berendezés: jelbemenetek, tápfeszültség bemenet, analóg kimenetek, soros IF ...

A zavarjelek csoportosítása hatásuk szerint

Zavart okozó esemény	Jellege	Hatás	Védelem	Példa
Túlfeszültség	Igen nagy energiatartalom	Meghibásodás	Túlfeszültségvédelem	Tápegység meghibásodás
Zavarjel	Nagy energiatartalom	Működés leáll, újra indítható	Zavarvédelem	PC lefagyás
Zavarjel	Kisebb energiatartalom	Átmenetileg hibás működés	Zavarvédelem	Hibás védelmi működés
Zavarjel	Kisebb energiatartalom	Átmenetileg specifikáción kívül	Zavarvédelem	Hibás adat egy regisztrátumban
Nem várt jelek	Tápkimaradás, harmonikusok	Specifikáción kívül	Megfelelő tervezés, szoftver	Hibás algoritmus

A működési zavar lehet:

Ártalmatlan: pl. akkumulátortöltő rövid kiesése

Kellemetlen: pl. monitor vibrálás

Javítható: pl. hibás mérés, megismételhető (?)

Kárt okoz: pl. adatvesztés, kemence kihűl, csirke megfagy, tirisztor tönkremegy

Nagy kárt okoz: pl. nagy értékű berendezés meghibásodik, energiaellátás kiesik

A zavarvédelem lépései

- zavarforrás megszüntetése
- csatolás megszüntetése ill. csökkentése
- bemeneti túlfeszültségvédelem, zavarszűrés, leválasztás
- zavartűrő berendezés tervezése (hardver, szoftver)
- zavarvizsgálatok elvégzése

A fontosabb zavarforrások I.

1. Villámcsapás LEMP

A villámcsapás és az azzal járó elektromágneses jelenségek olyan elkerülhetetlen jelenségek, amelyek különböző hatásmechanizmusok útján tranziens túlfeszültségeket hoznak létre a villamos, távközlési és egyéb (pl. információtechnológiai, kábeltévé és jelvezetékek) vezetékes rendszerekben.

Várható értékek:

	U_p kV	dU/dt kV/us	I_p kA	dI/dt kA/us	Q As
Közvetlen	<1000		150-250	<100	50
Közvetett	10-100	10-100	1-10		<1

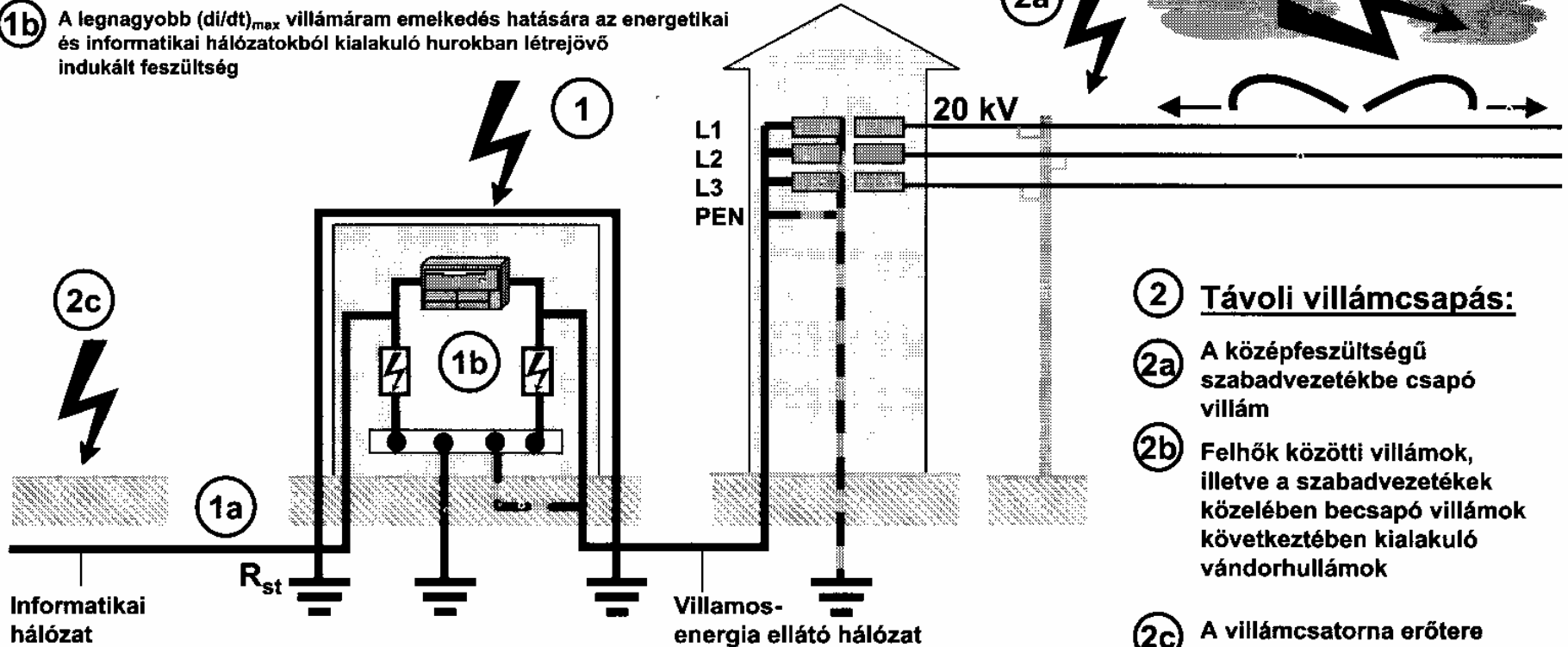
A villámcsapás mechanizmusának szemléltetése

1 Közvetlen vagy közeli villámcsapás:

A külső villámvédelmi berendezést, az ipari berendezések fémszerkezetét, kábelt, stb., érő villámcsapás

1a) Az R_s földelési ellenálláson a lököáram következtében létrejövő feszültségesés

1b) A legnagyobb $(di/dt)_{max}$ villámáram emelkedés hatására az energetikai és informatikai hálózatokból kialakuló hurokban létrejövő indukált feszültség



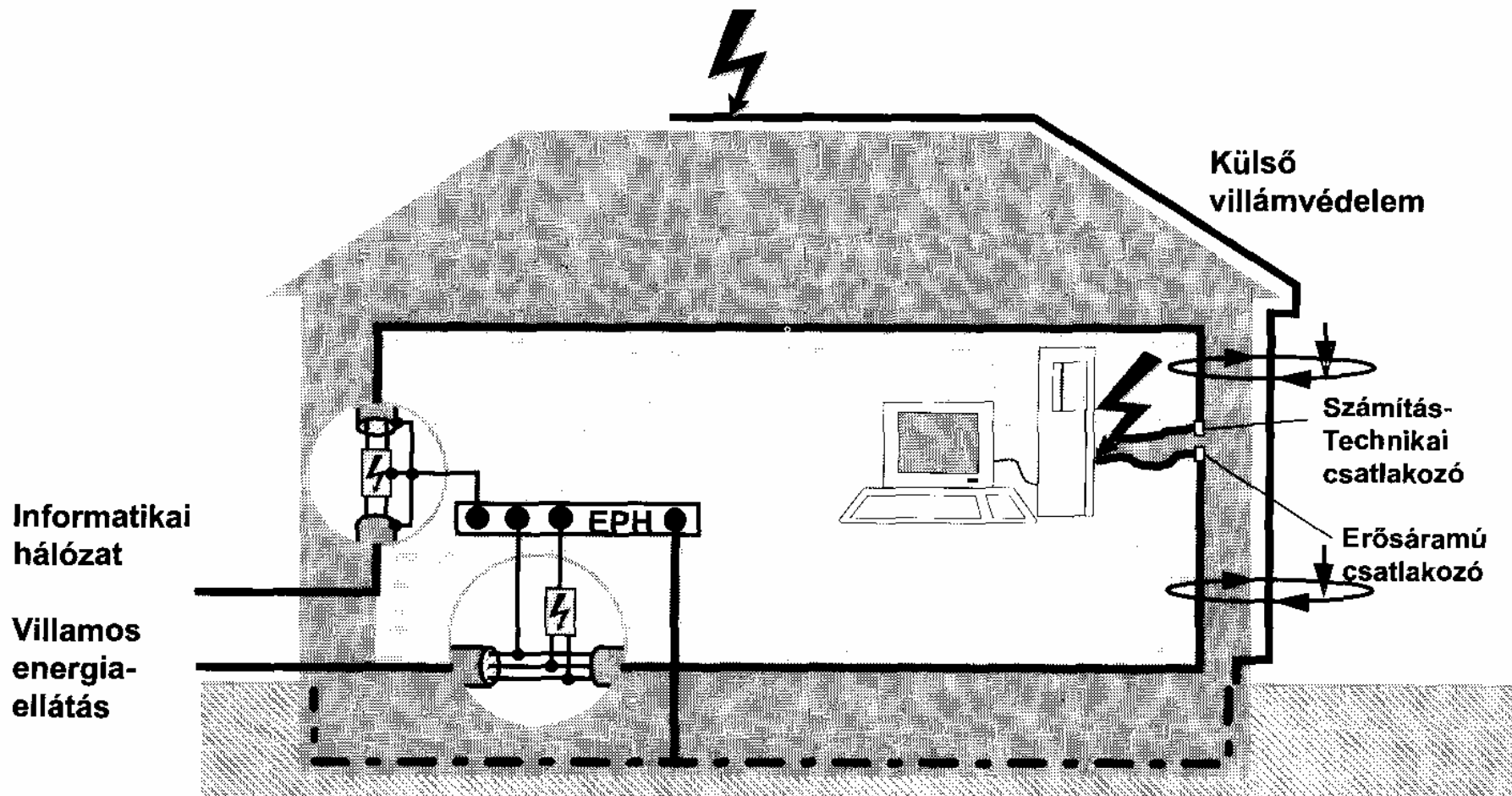
2 Távoli villámcsapás:

2a) A középfeszültségű szabadvezetékbe csapó villám

2b) Felhők közötti villámok, illetve a szabadvezetékek közelében becsapó villámok következtében kialakuló vándorhullámok

2c) A villámcsatorna erőtere

Az induktív és konduktív (az EPH-n keresztül) csatolások szemléltetése a villámfelfogókhoz vagy az épület egyéb szerkezeteihez történő közvetlen villámcsapás esetére



A fontosabb zavarforrások II.

2. Nukleáris elektromágneses impulzus (NEMP)*

várható értékek:

	LEMP 100m	NEMP talajszinten
E_{max} kV/m	40	100
H_{max} A/m	160	700
tf nsec	10..1000	5..8

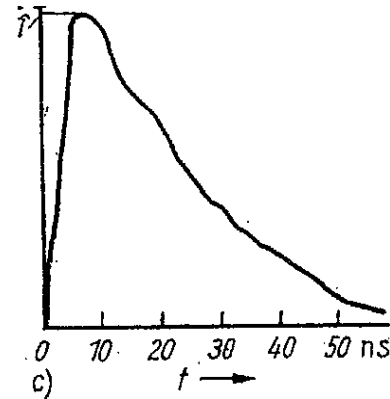
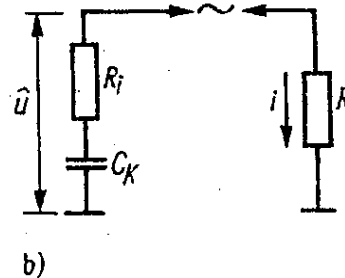
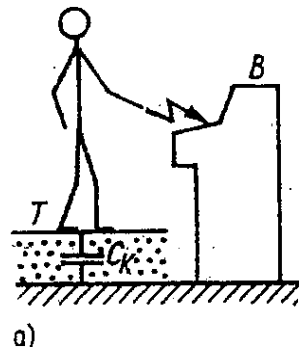
pl. 1 m² vezetékurokban indukált feszültség 120 kV

* Nem képezi az IEC 1000-5 sorozat tárgyát

A fontosabb zavarforrások III.

3. Elektrosztatikus kisülés ESD

elektrosztatikusan feltöltött kezelő hatására



jellemző értékek:

$$U=5..15 \text{ kV}; C=100..250\text{pF};$$

$$R=100..500 \text{ Ohm}$$

$$dU/dt=1\text{kV/ns}$$

A fontosabb zavarforrások IV.

4. Kapcsolási tranziensek:

transzformátorok, induktivitások bekapcsolása

áramirányító kommutáció

induktivitások, relék kikapcsolása

kiolvadó biztosító működések

pl. egyenáramú relék kikapcsolásakor elvileg fellépő
túlfeszültségek láthatók az alábbi táblázatban

Relé típus	U [V]	I [mA]	Rr [kOhm]	Lr [mH]	Cr [pF]	f0 [kHz]	Umax [kV]
Schrack RP010024	24	20	1.2	2.7	75	28	3.8
RELOG 2RH 01	220	12	19	217	116	3	15.8

A fellépő túlfeszültségek átütik a kapcsoló kontaktusát ennek hatására nagyfrekvenciás impulzus sorozat (burst) keletkezik.

A fontosabb zavarforrások V.

5. Hálózati feszültség zavarai

Feszültségtolerancia 10%

negatív sorrend 2%

Feszültségletörés 10..99%

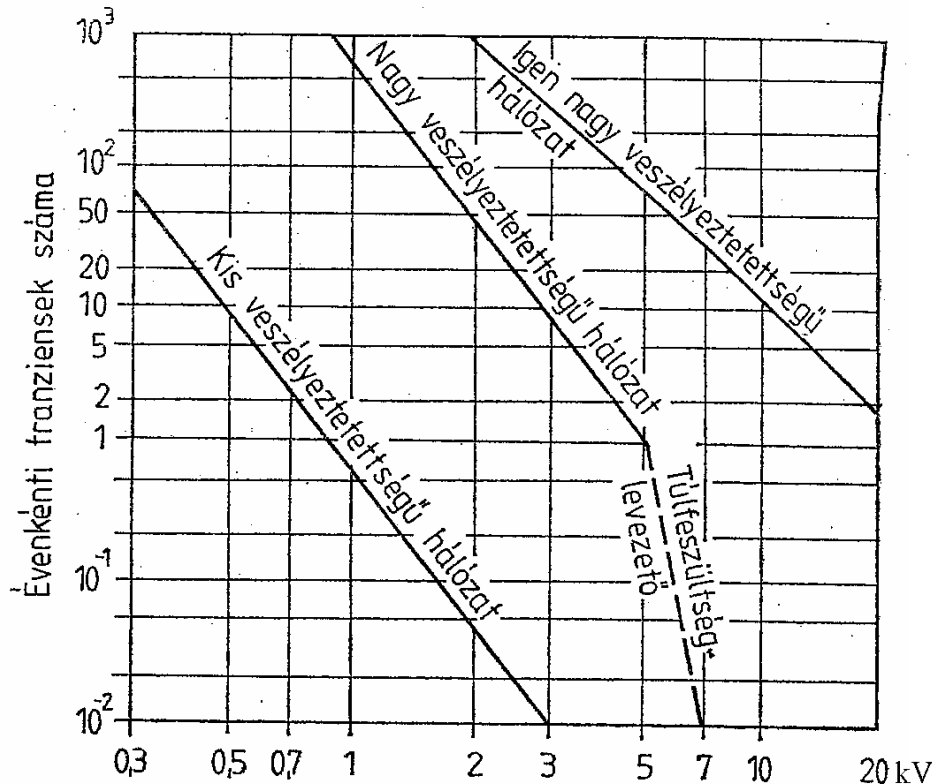
feszültség kimaradás 10msec-től

feszültség harmonikusok 5..10%

áram harmonikusok 5..30%

feszültség emelkedés 110..130%

tranziens túlfeszültség 200..2000V

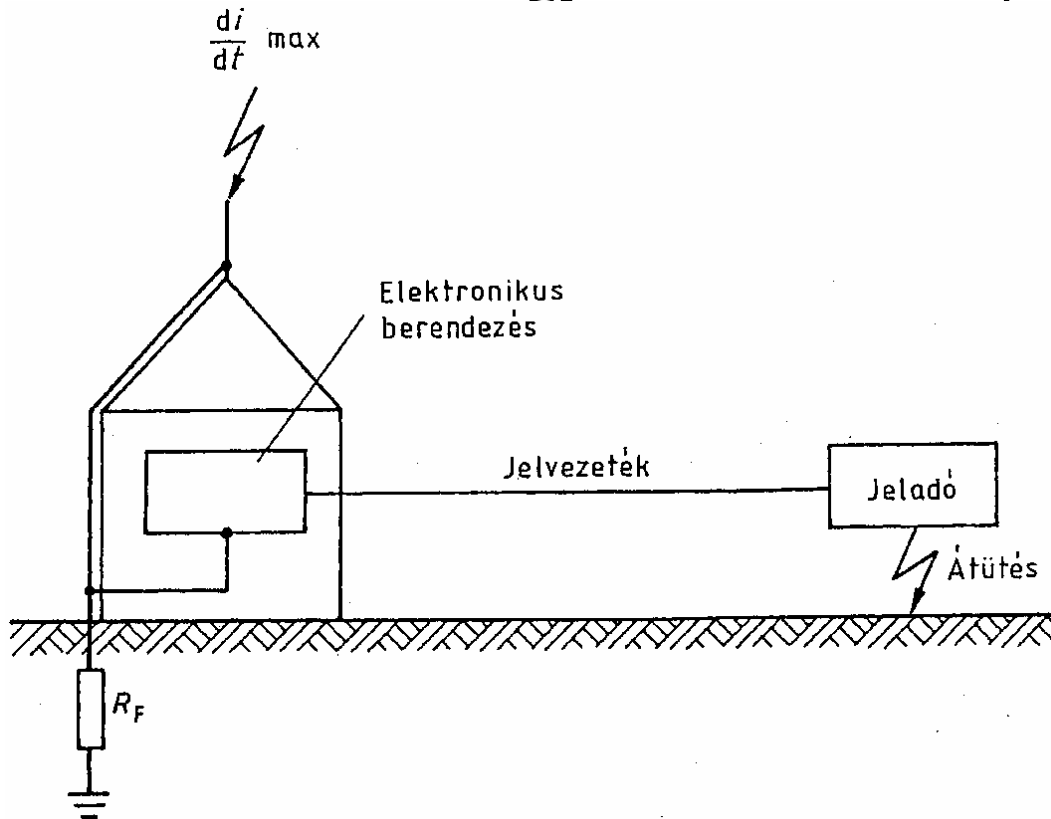


Villamosenergia-elosztó hálózatok feszültségtranziens-gyakoriságának eloszlási görbéje

A zavarok csatolásának módjai I.

1. Galvanikus csatolás

közös vezetéken vagy földelésen fellépő feszültség



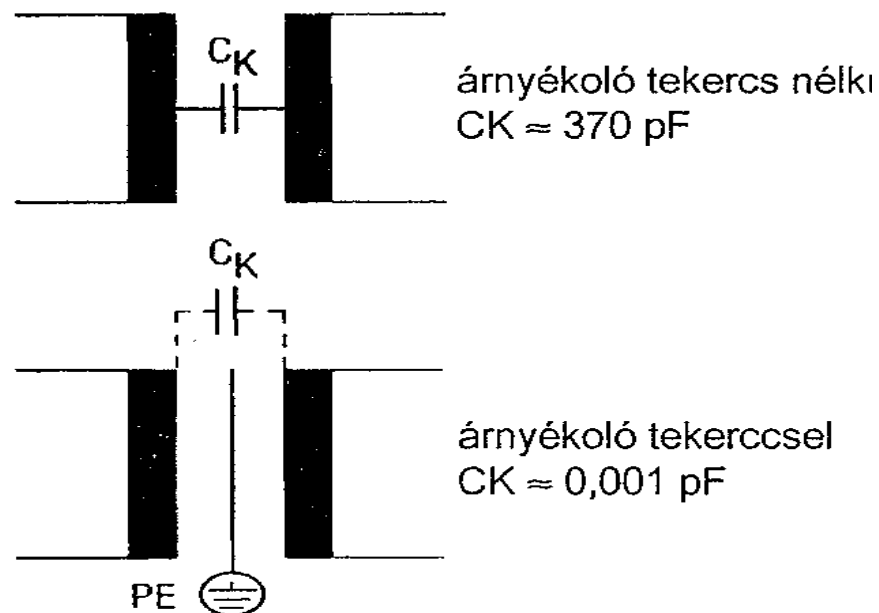
védekezés: külön földvezetékek, galvanikus leválasztás

A zavarok csatolásának módjai II.

2. Kapacitív csatolás

szórt kapacitások, pl. transzformátor tekercsei között

Transzformátor csatolókapacitása

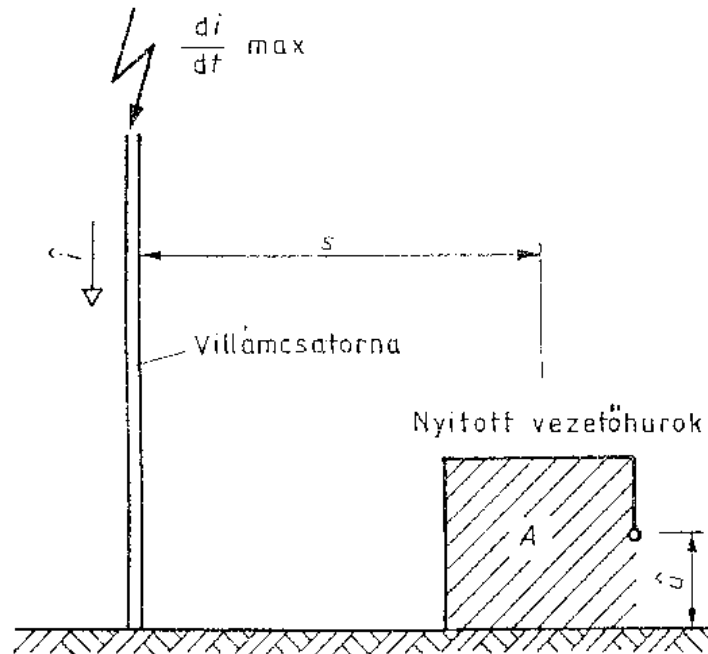


védekezés árnyékolással lehetséges

A zavarok csatolásának módjai III.

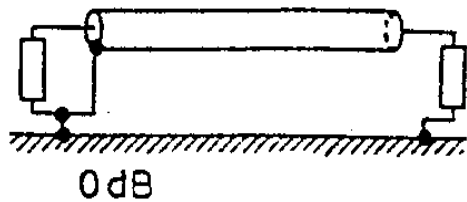
3. Induktív csatolás

vezetékgyűrűk közötti kölcsönös induktivitás

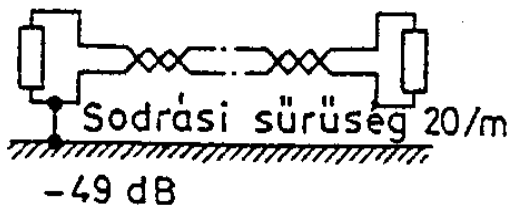


védekezés: csatolások csökkentése, mágneses árnyékolás

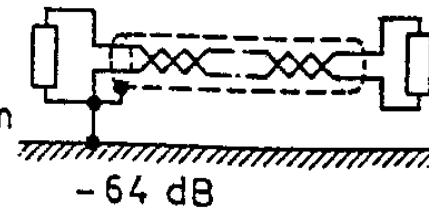
Kábelek csillapítása külső zavarjelekre



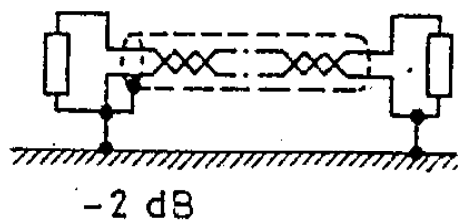
a)



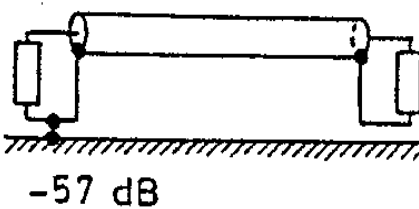
d)



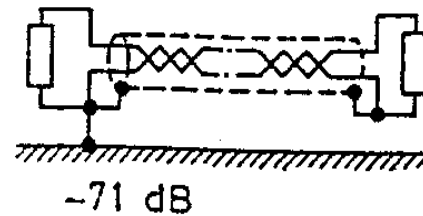
g)



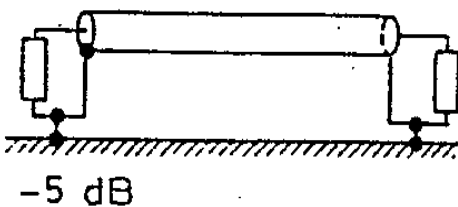
b)



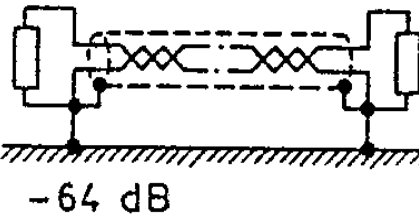
e)



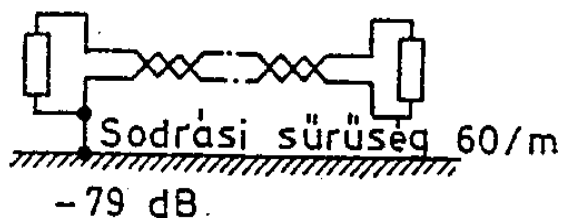
h)



c)



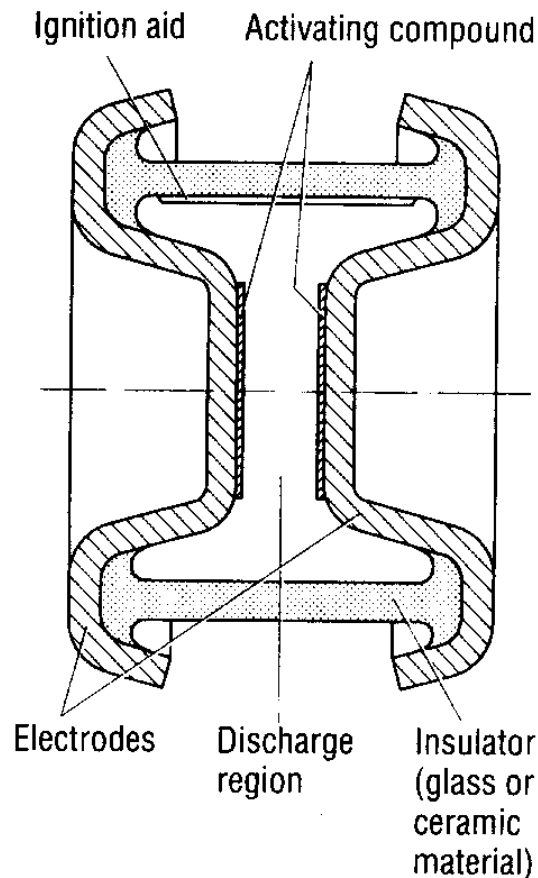
f)



i)

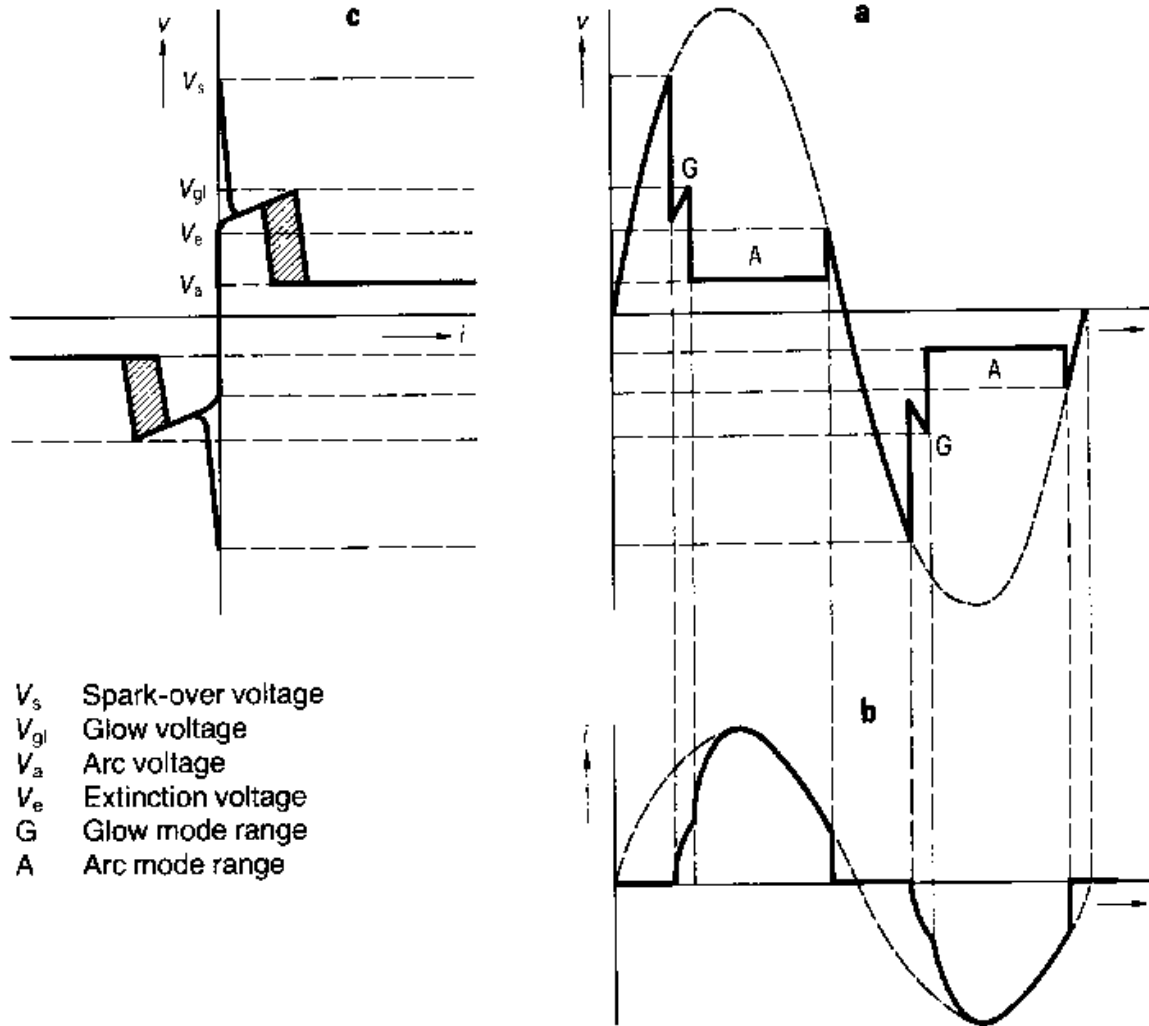
Túlfeszültség védelmi eszközök

1. Szikraköz

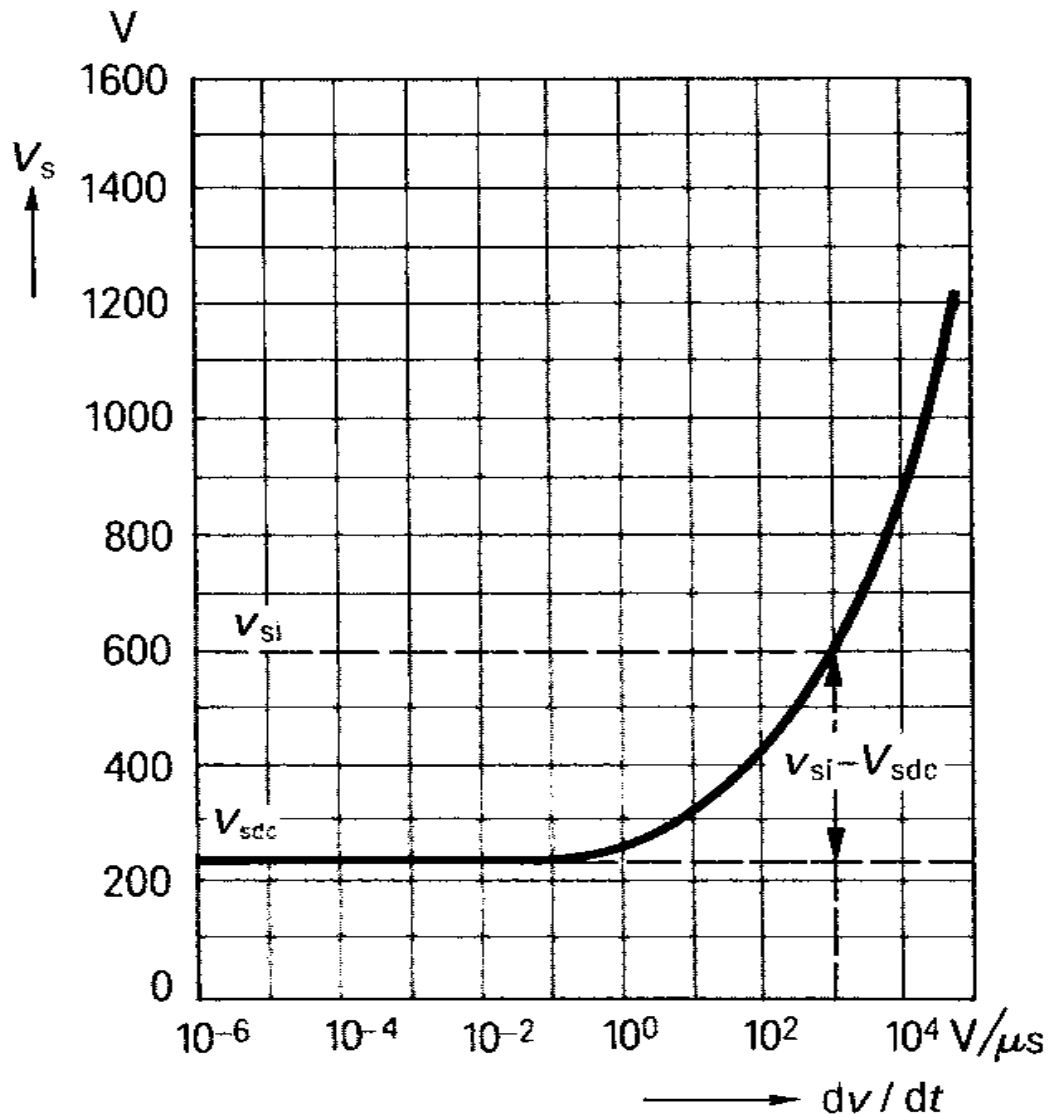


Nemesgázzal töltött, üveg vagy kerámia szigetelésű szikraköz, amely túlfeszültség hatására begyújt.

Szikraköz működési karakterisztikája

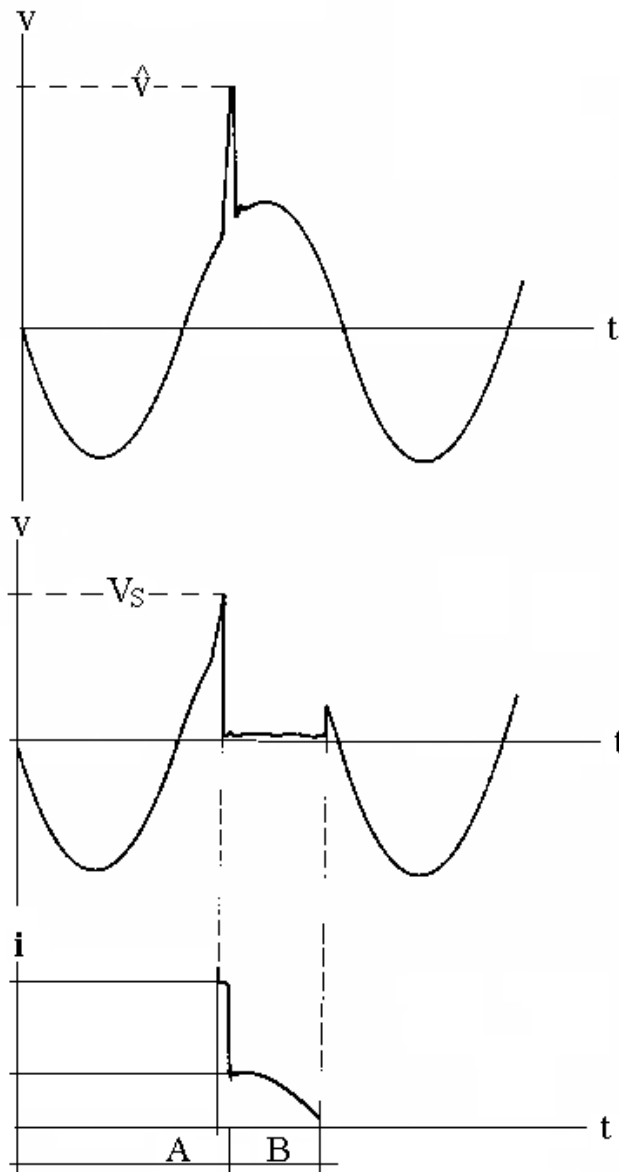


Szikraköz megszólalási feszültsége a feszültség meredekség függvényében

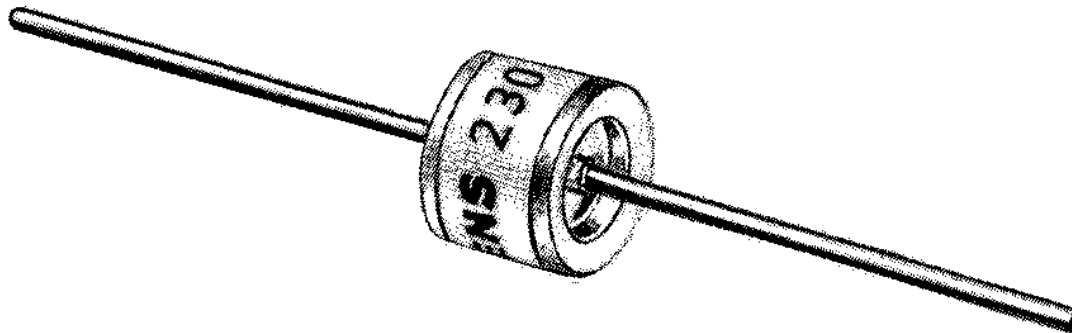
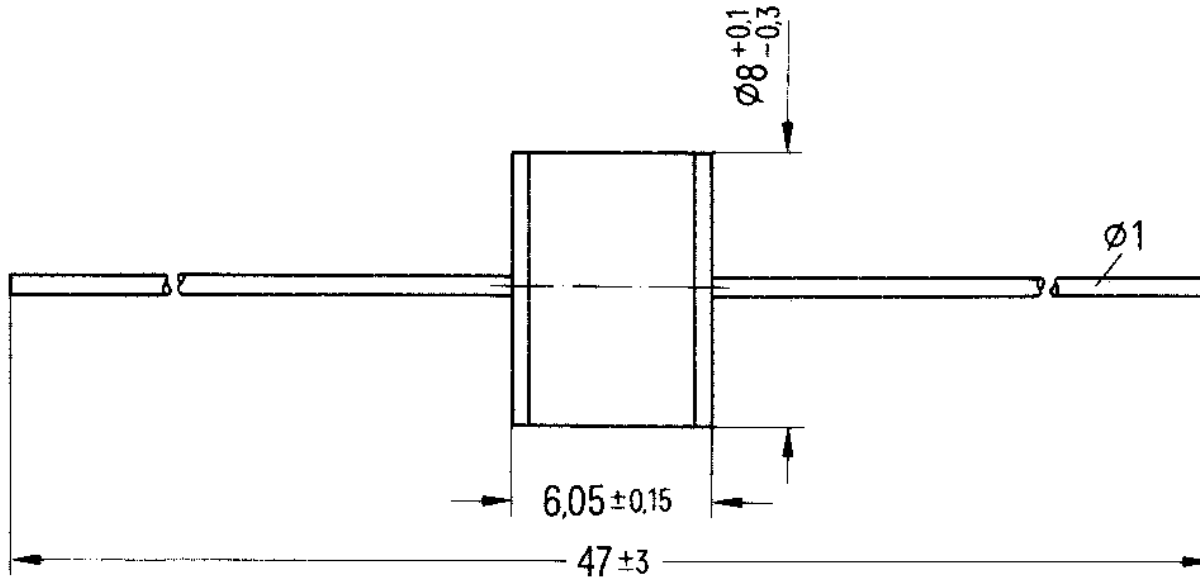


Szikraköz működése túlfeszültség hatására.

A szikraköz az áram nulla átmenetekor alszik ki.



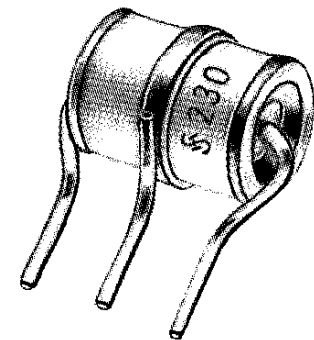
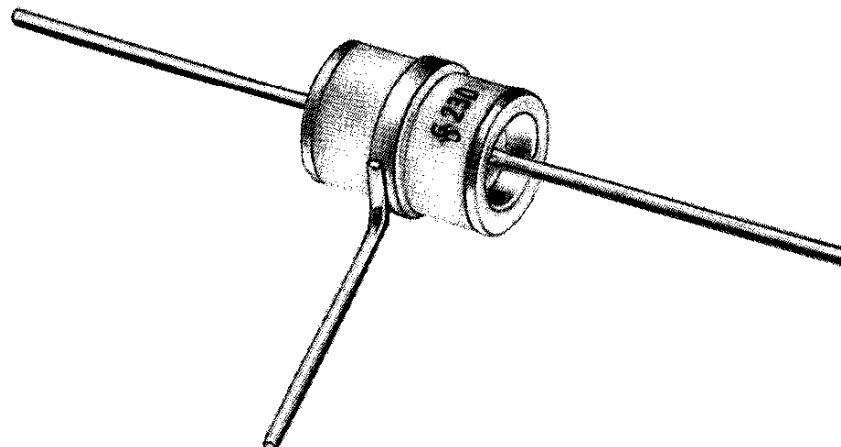
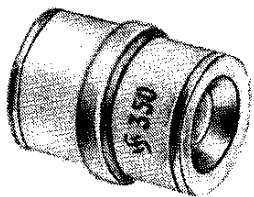
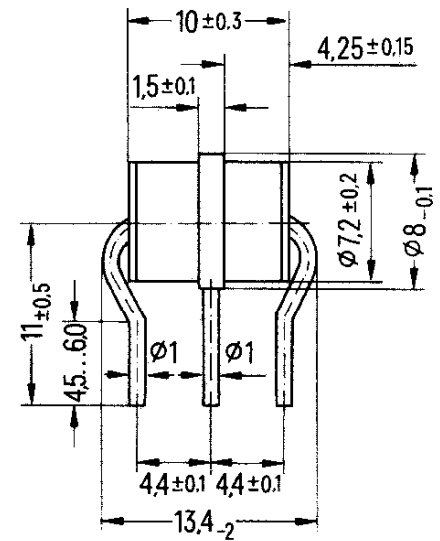
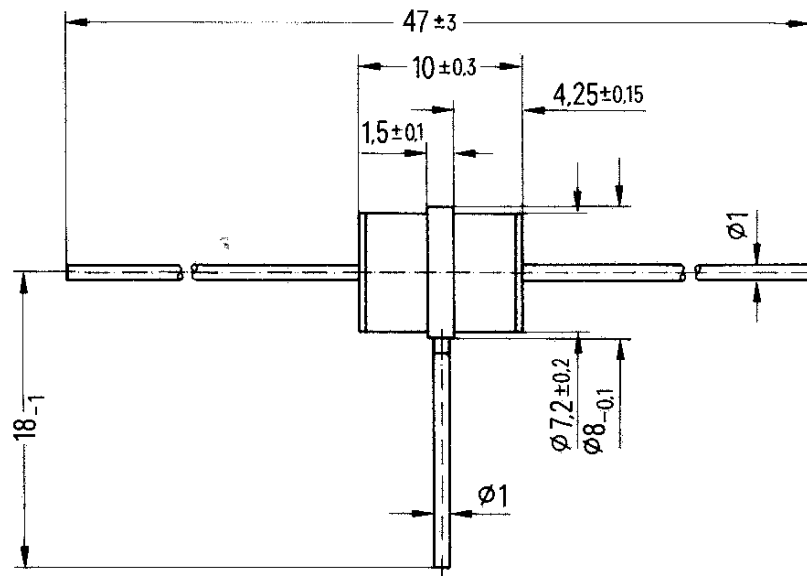
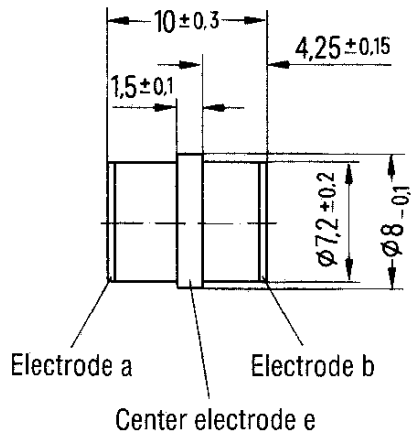
Siemens gyártású levezető méretei



Túlfeszültség levezető jellemző adatai

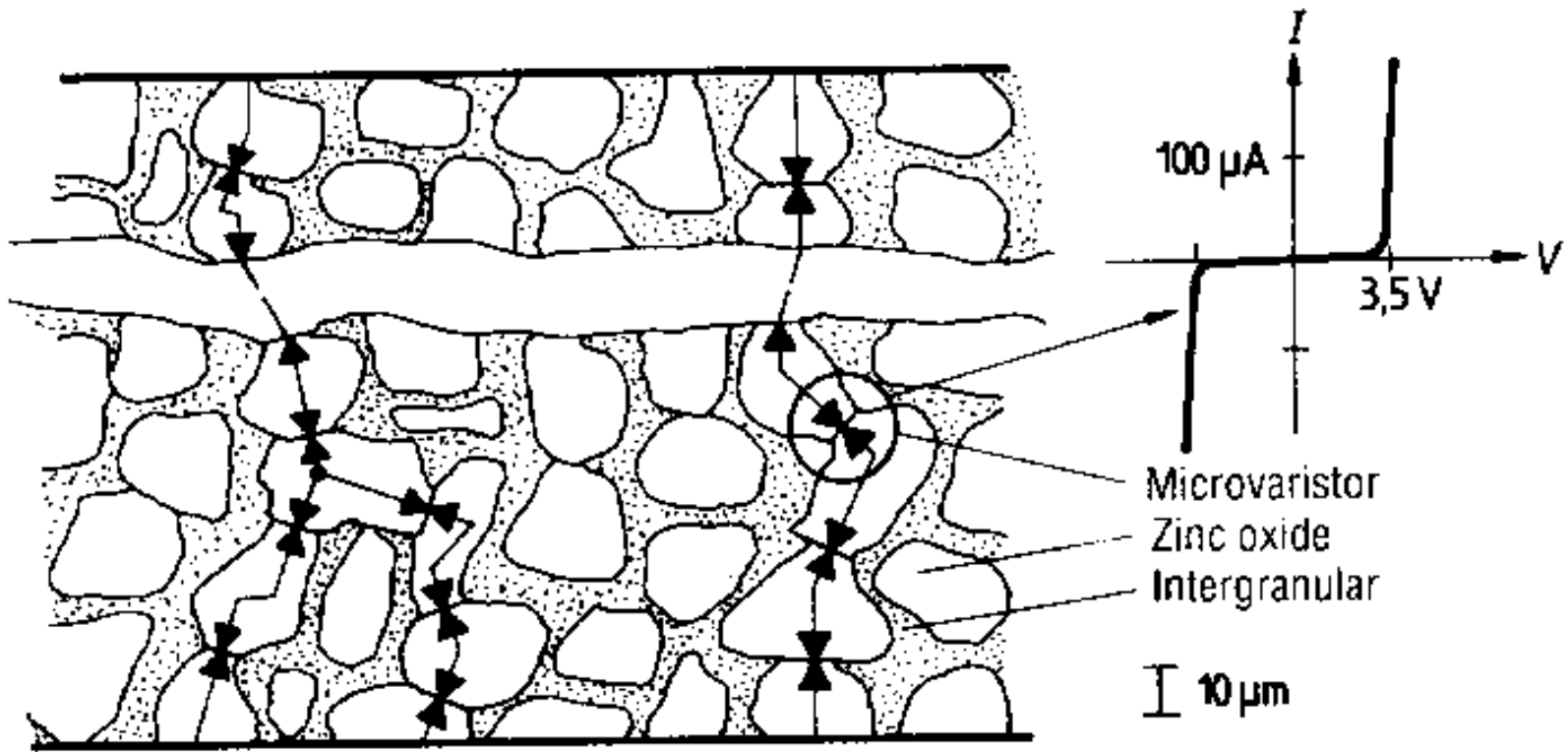
	Type	A81-C90X	A81-A230X	A81-A350X
Nominal dc spark-over voltage V_{sdCN}	V	90	230	350
DC spark-over voltage referred to V_{sdCN} in	%	± 20	± 20	± 20
Impulse spark-over voltage v_{si} (at 1 kV/ μ s)	V	< 650	< 650	< 700
Nominal impulse discharge current i_{diN} (8/20 μ s wave)	kA	10	10	10
Nominal alternating discharge current I_{daN} (50 Hz, 1 s)	A	20	20	20
Insulation resistance R_{is} (at 100 V dc) (for $V_{\text{sdCN}}=90$ V at 50 V dc)	Ω	$\geq 10^{10}$	$\geq 10^{10}$	$\geq 10^{10}$
Capacitance C	pF	≈ 1.5	≈ 1.5	≈ 1.5

Két egybeépített levezető

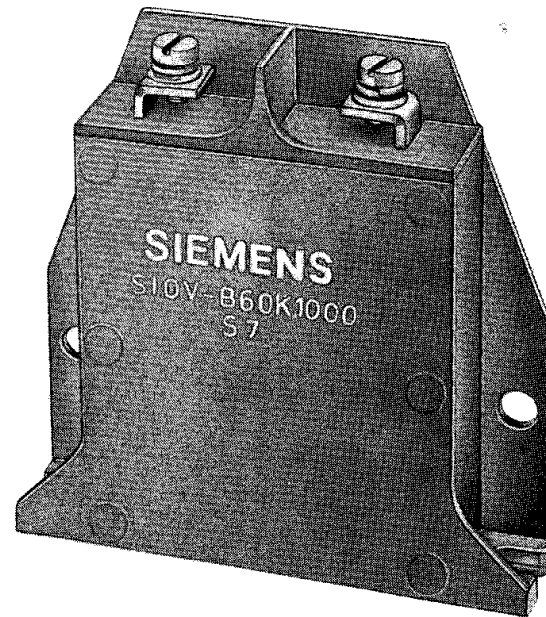
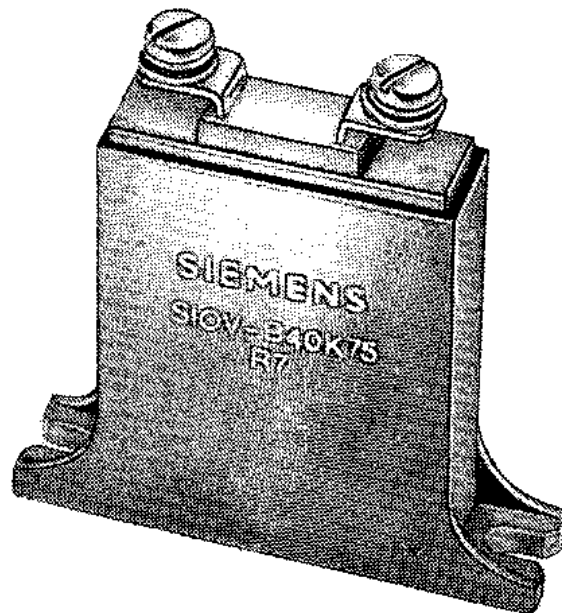
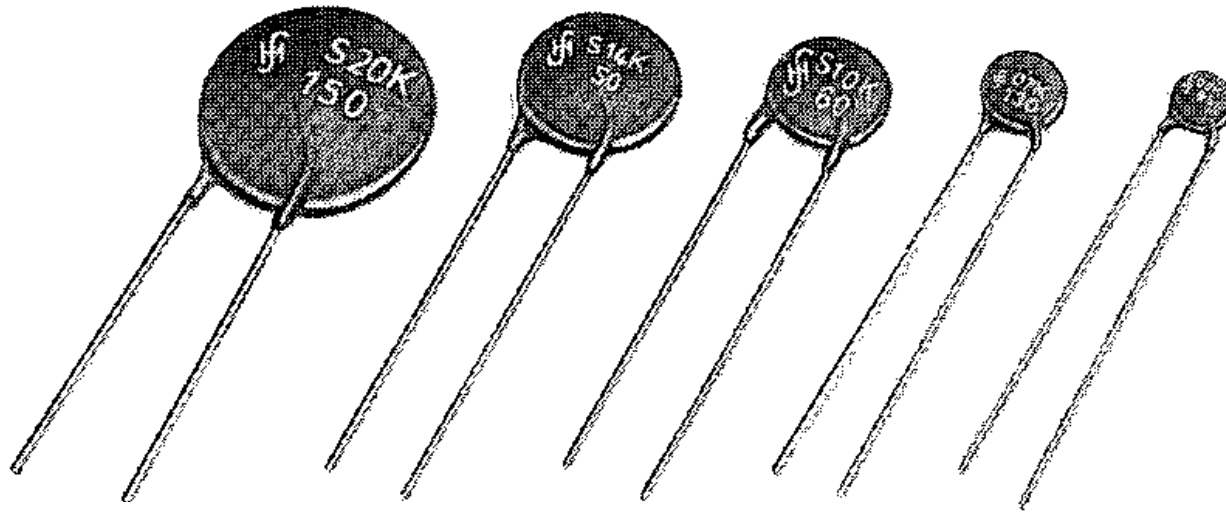


Varisztor

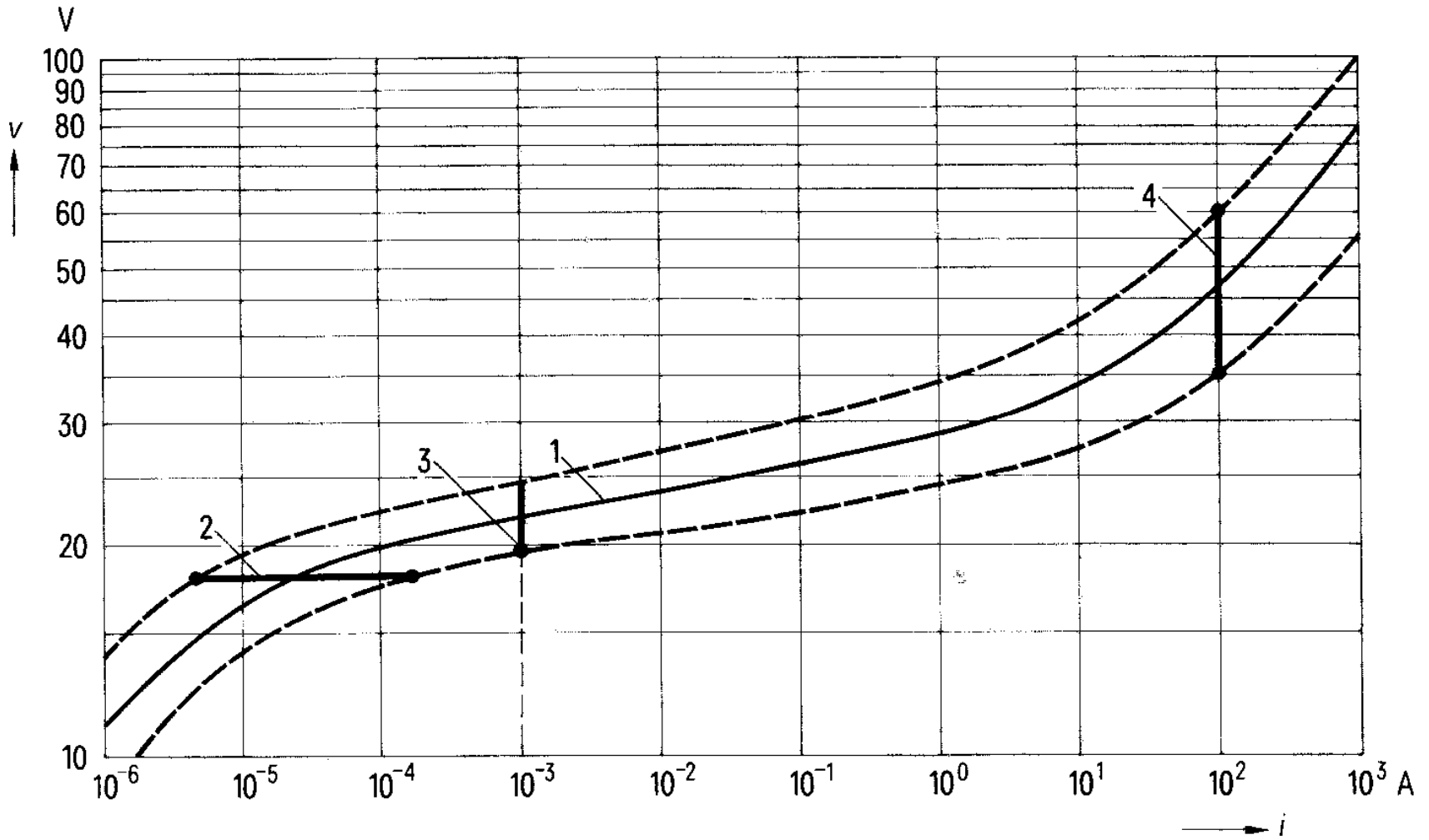
Fémoxid, jellemzően cinkoxid alapú kerámia jellegű anyagból égetett eszköz, amelynek nemlineáris feszültség-áram karakterisztikája van



Különböző típusú és teljesítményű varisztorok



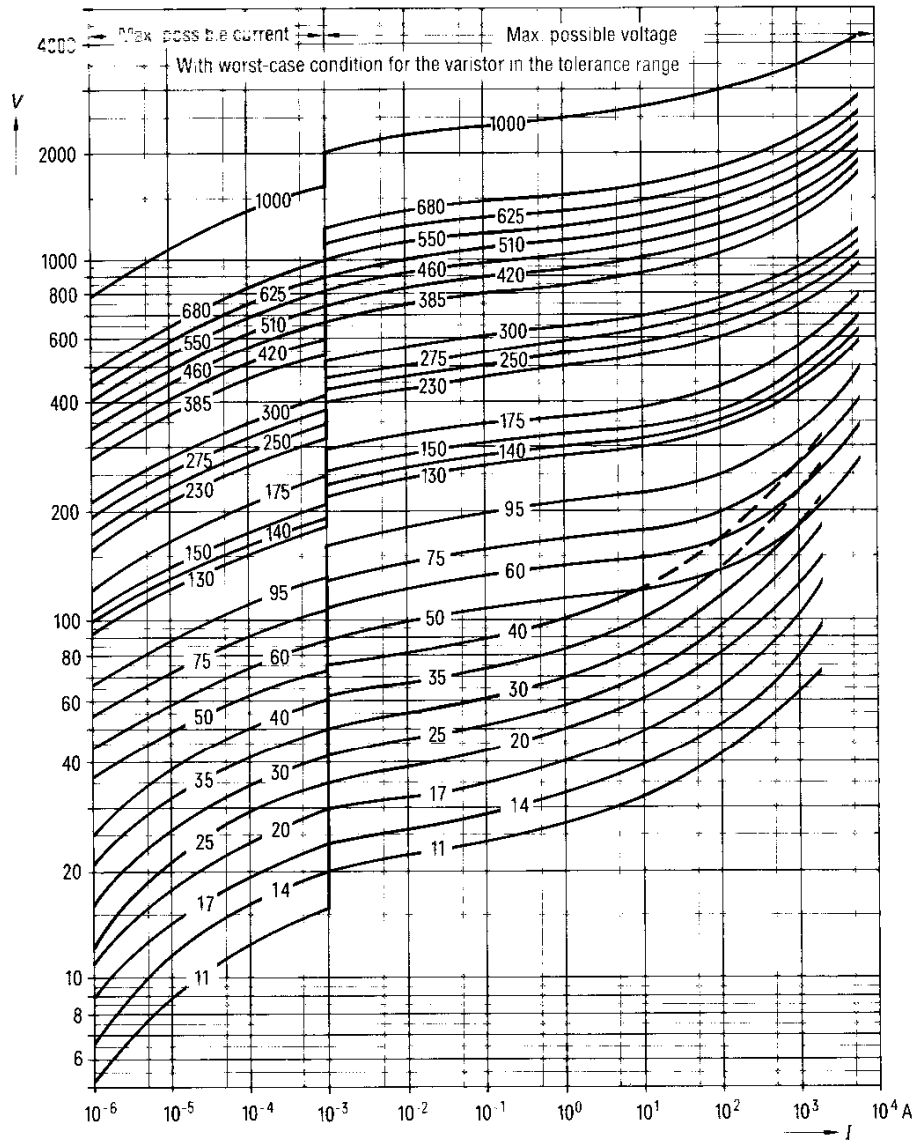
Varisztor karakterisztika I.



Varisztor karakterisztika II.

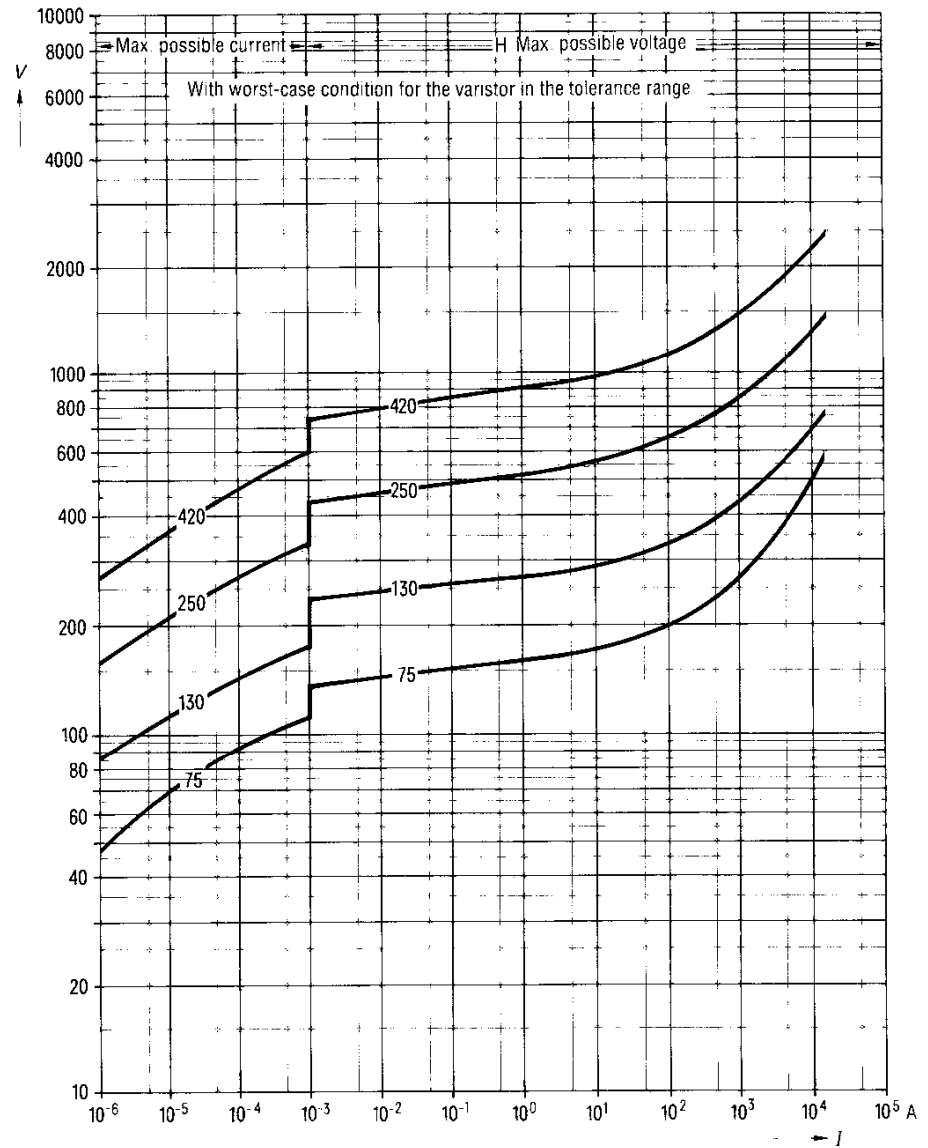
v 2.22.5

SIOV-S20K11...1000



v 2.22.6

SIOV-B25K75...420



Varisztorok jellemző adatai

névleges feszültség (1 mA) 15..1500 V

max áram (8/20 μ sec)

S05K230	1x	400 A	850 V
	10^6 x	10 A	650 V
S20K230	1x	6500 A	950 V
	10^6 x	50 A	550 V

max tartós disszipáció

S05K230	0.1 W
S20K230	1 W

kapacitás

S05K230	110 pF
S20K230	1000 pF

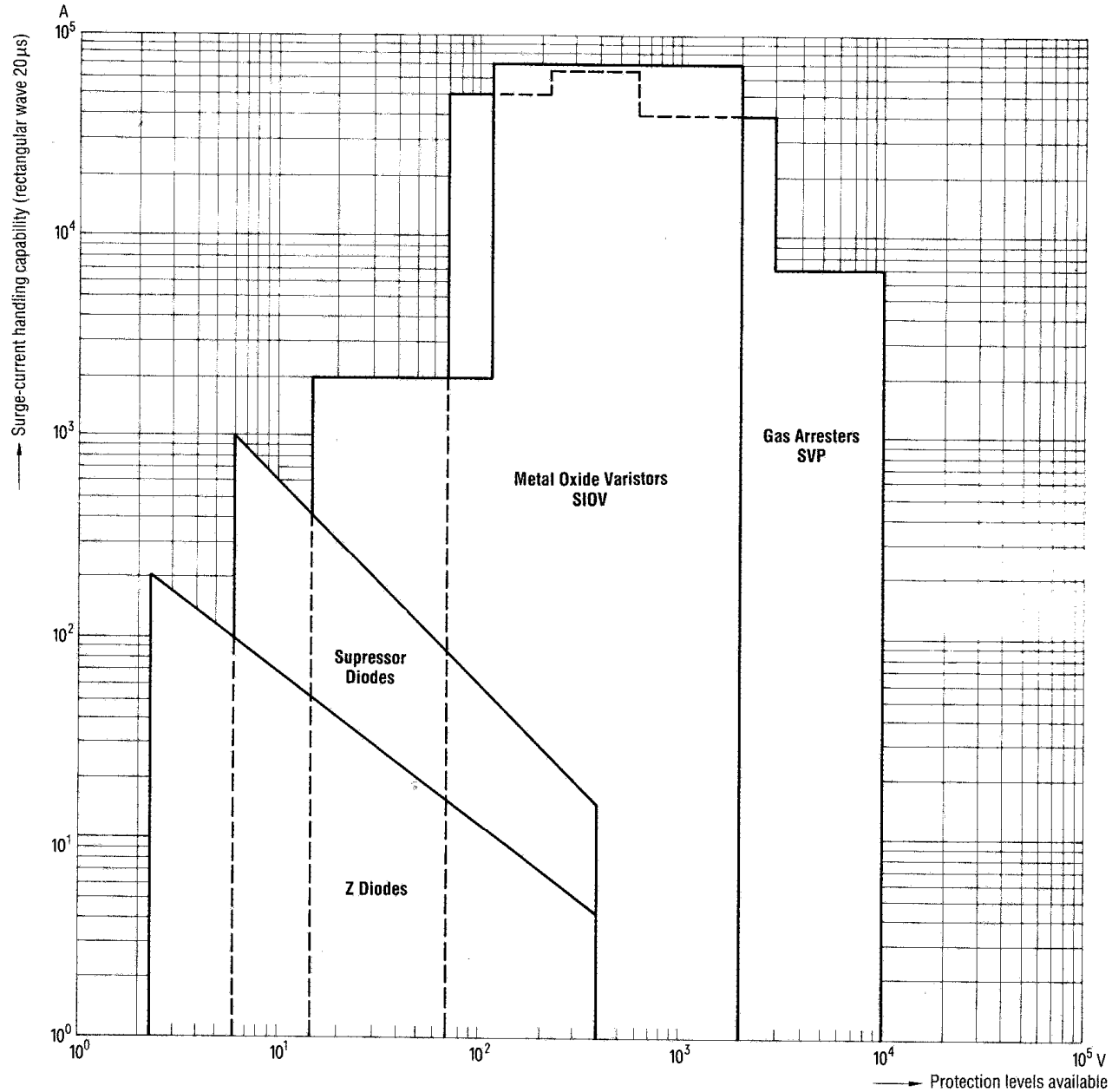
Szupresszor dióda

Rövid idejű, nagy áramlökések levezetésére alkalmas speciális zéner dióda

Jellemző adatok:

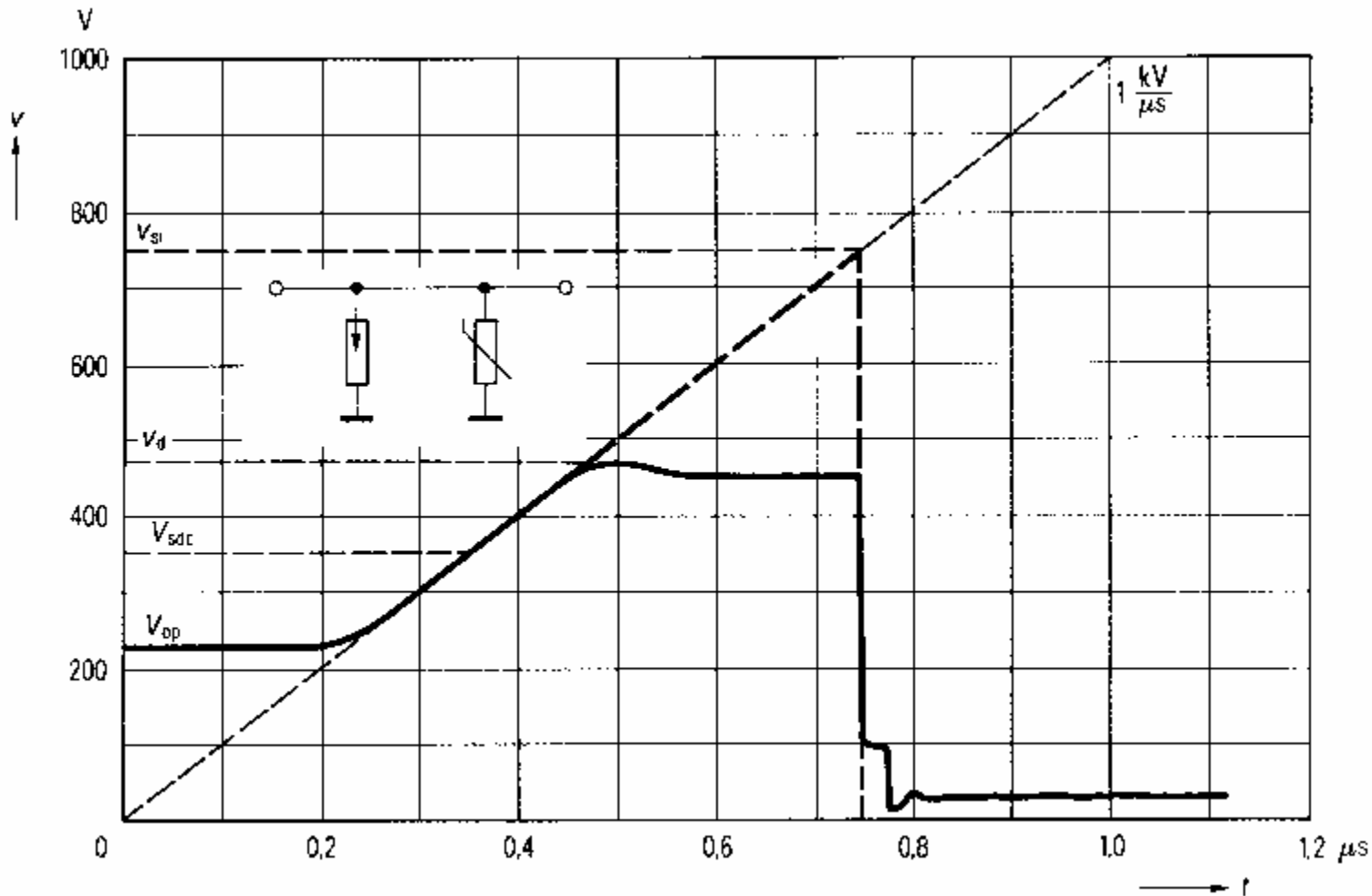
névleges feszültség	5..200 V
kapcsolási idő	1..10 psec
polaritás	mono- vagy bipoláris

Túlfeszültség levezető eszközök összehasonlítása



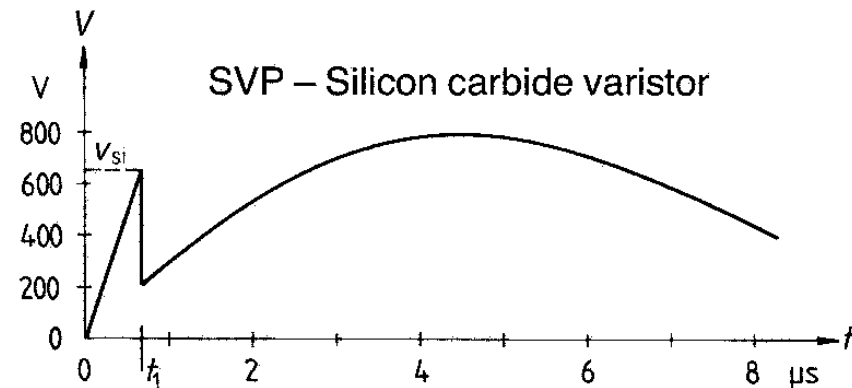
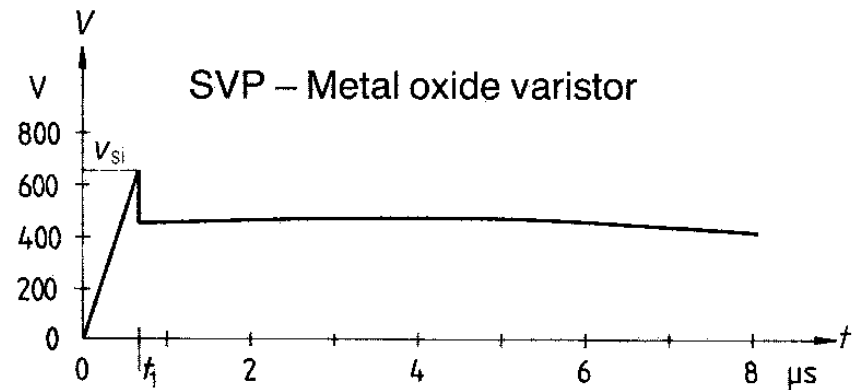
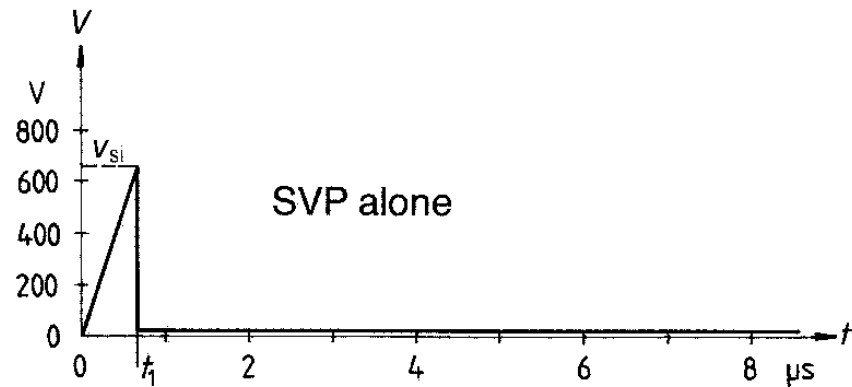
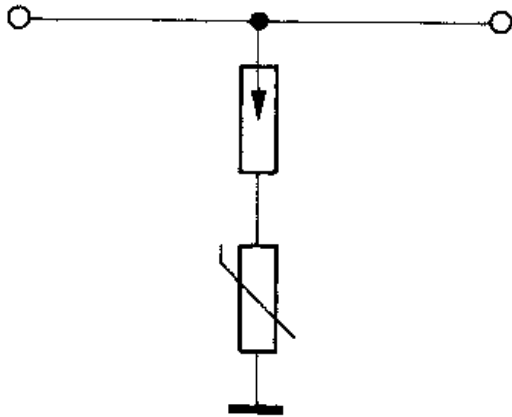
Túlfeszültség védelmi eszközök kombinált alkalmazása I.

Szikraköz és varisztor párhuzamos kapcsolása

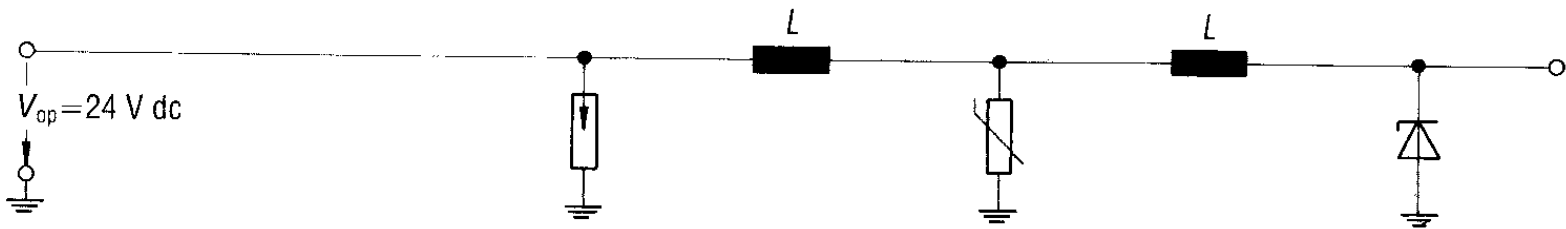
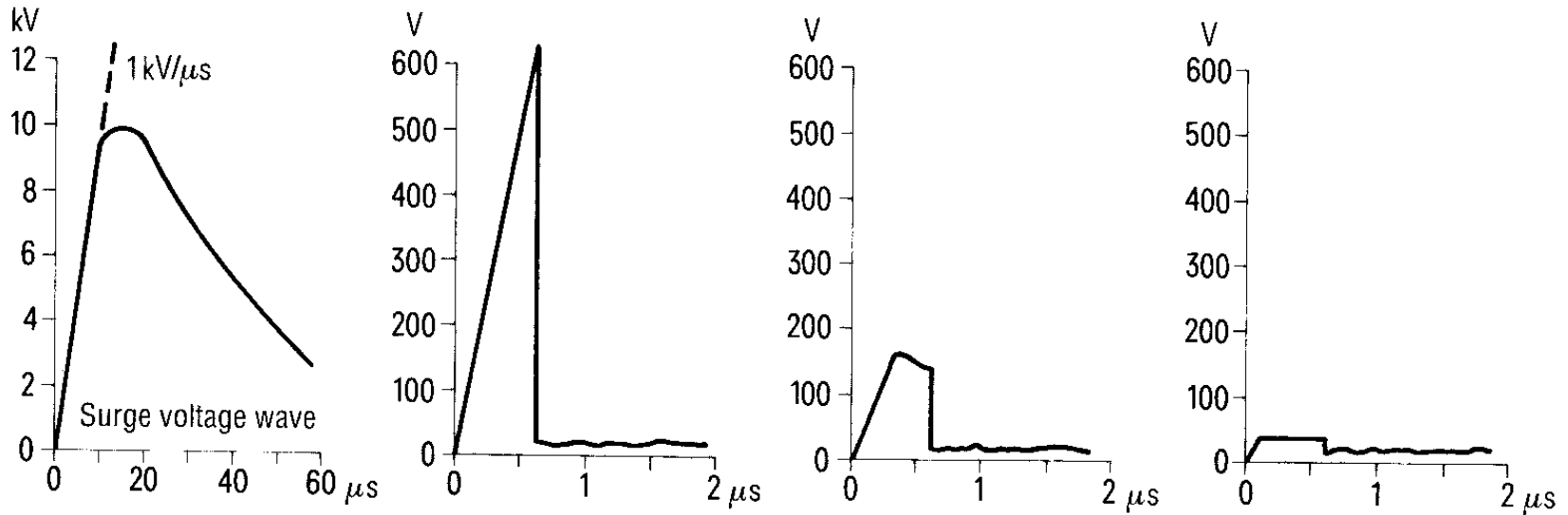


Túlfeszültség védelmi eszközök kombinált alkalmazása II.

Szikraköz és varisztor soros kapcsolása



Töblépcsős túlfeszültség védelmi elrendezés



Type	S1-C150	SIOV-S20K25	1N5645A
Surge current	up to 50 kA	up to 2 kA	up to 0.2 kA
Response time	500 ns	<25 ns	<10 ps

Zavarszűrés

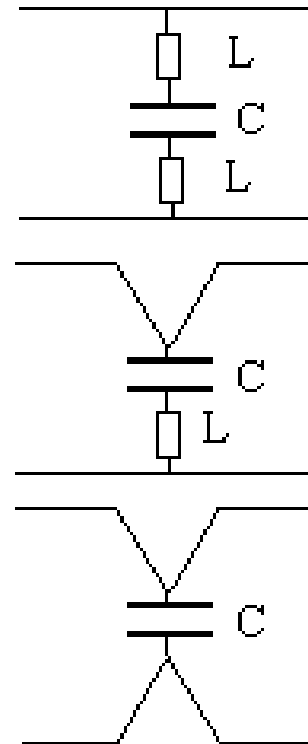
Mi a zavarszűrés? Magasabb frekvenciájú jelek eltávolítása.

Eszközök:

1. Kondenzátorok, nagy határfrekvenciájú X vagy Y típusú

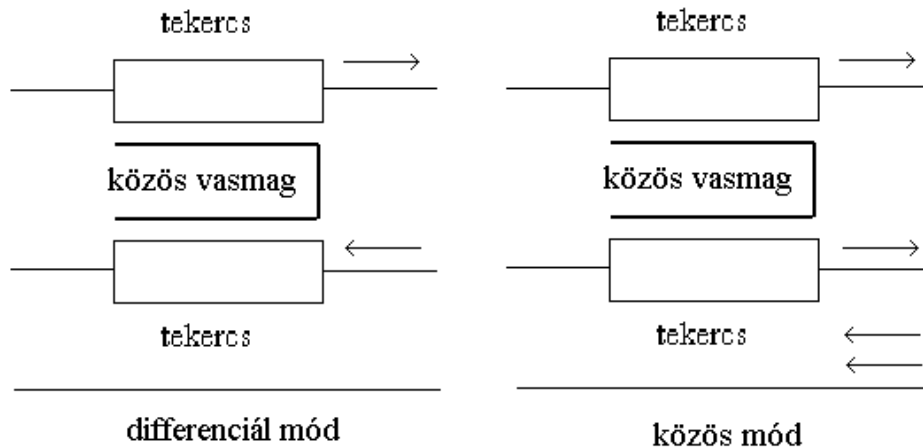
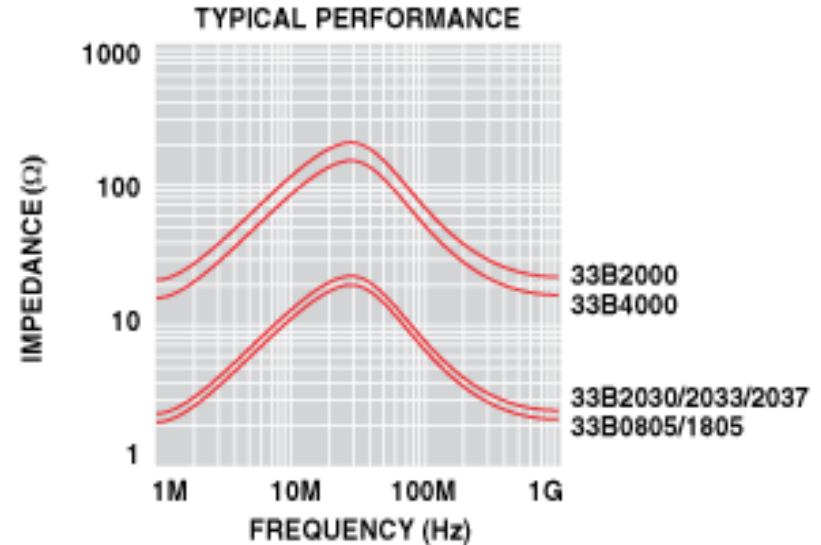
A tekercsek és hozzávezetések induktivitása következtében a kondenzátor egy adott frekvencia felett induktivitásként viselkedik.

Az Y és X kialakítás megnöveli a kondenzátor határfrekvenciáját.

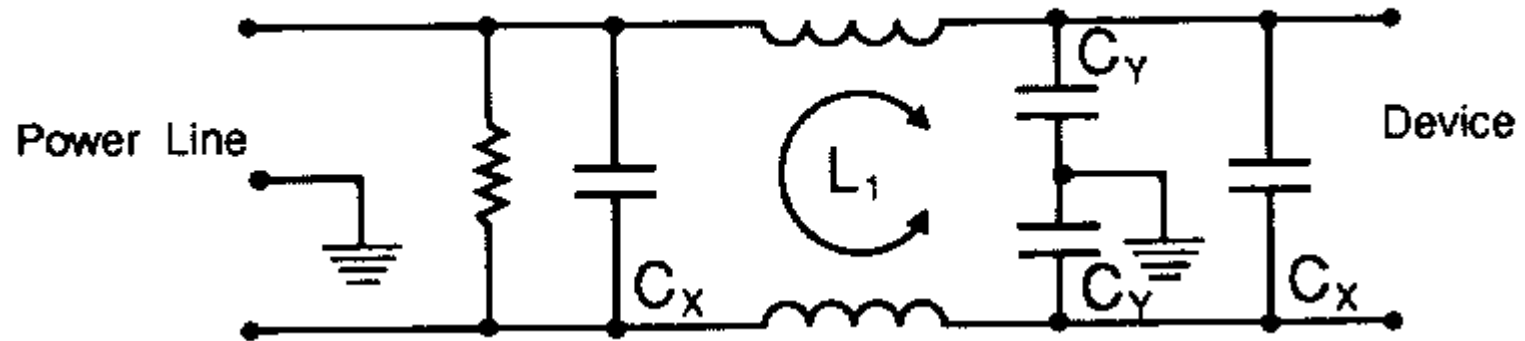


Induktivitások

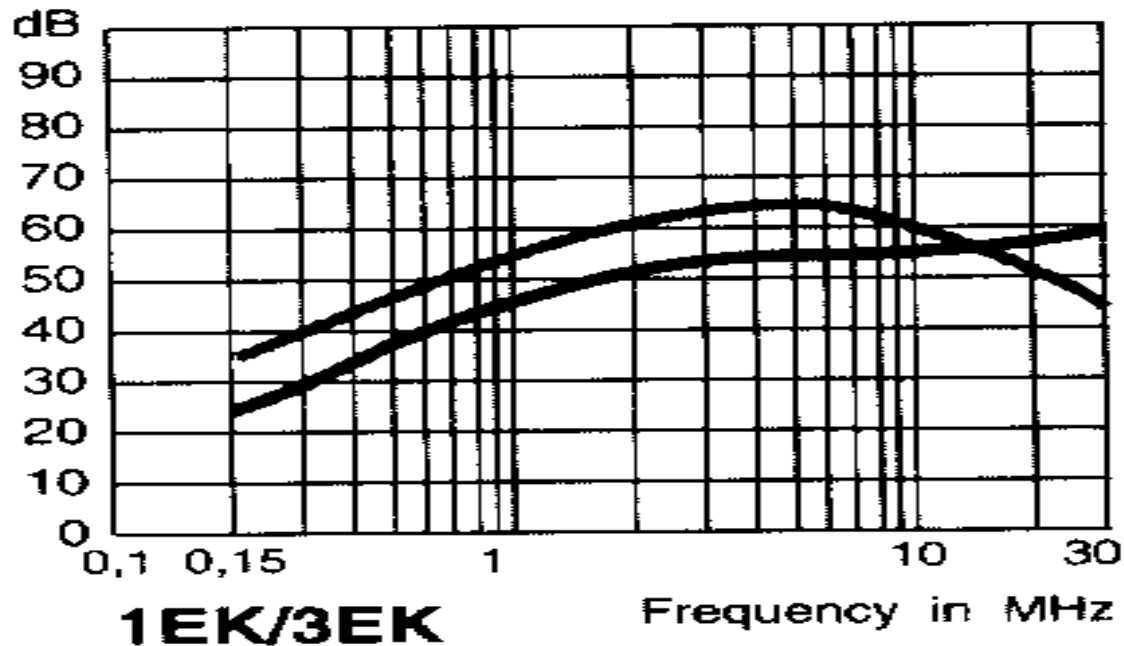
- A. Ferritgyűrű. Vezetékre húzva nagyfrekvencián csillapít.
- B. Gyári induktivitás beiktatása
- C. Tekert induktivitás ferritmagon hálózati bemeneten.



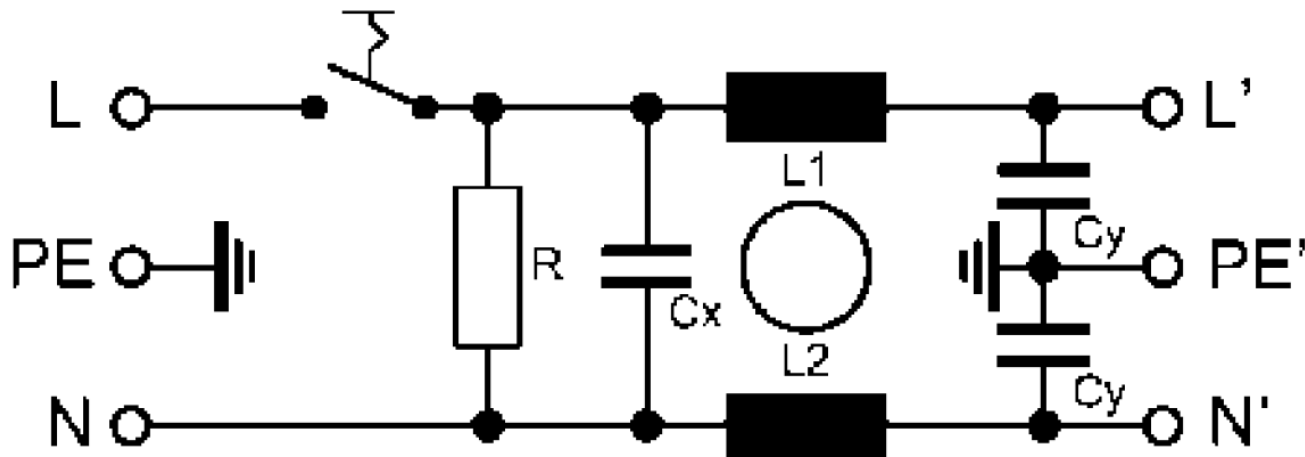
Hálózati zavaroszűrő kapcsolása



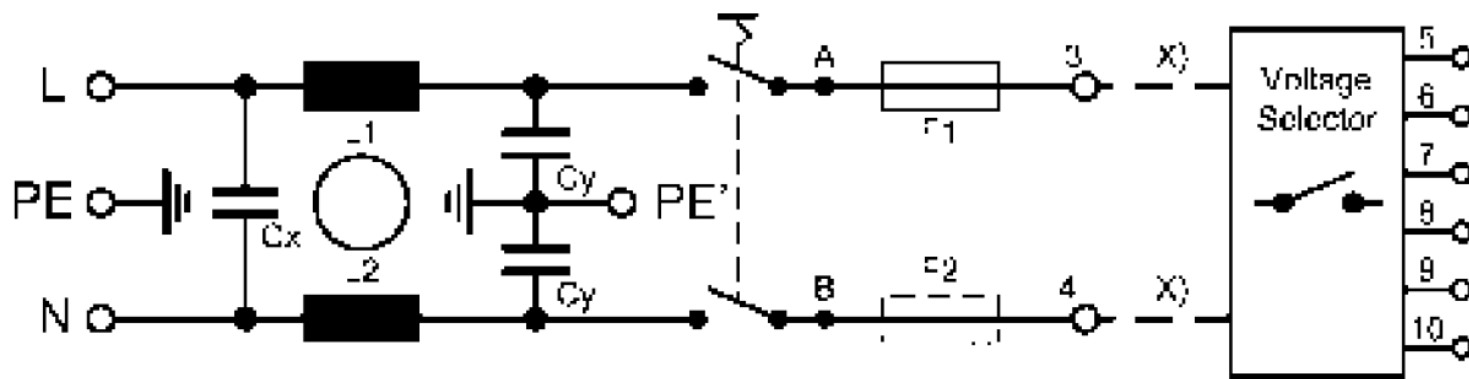
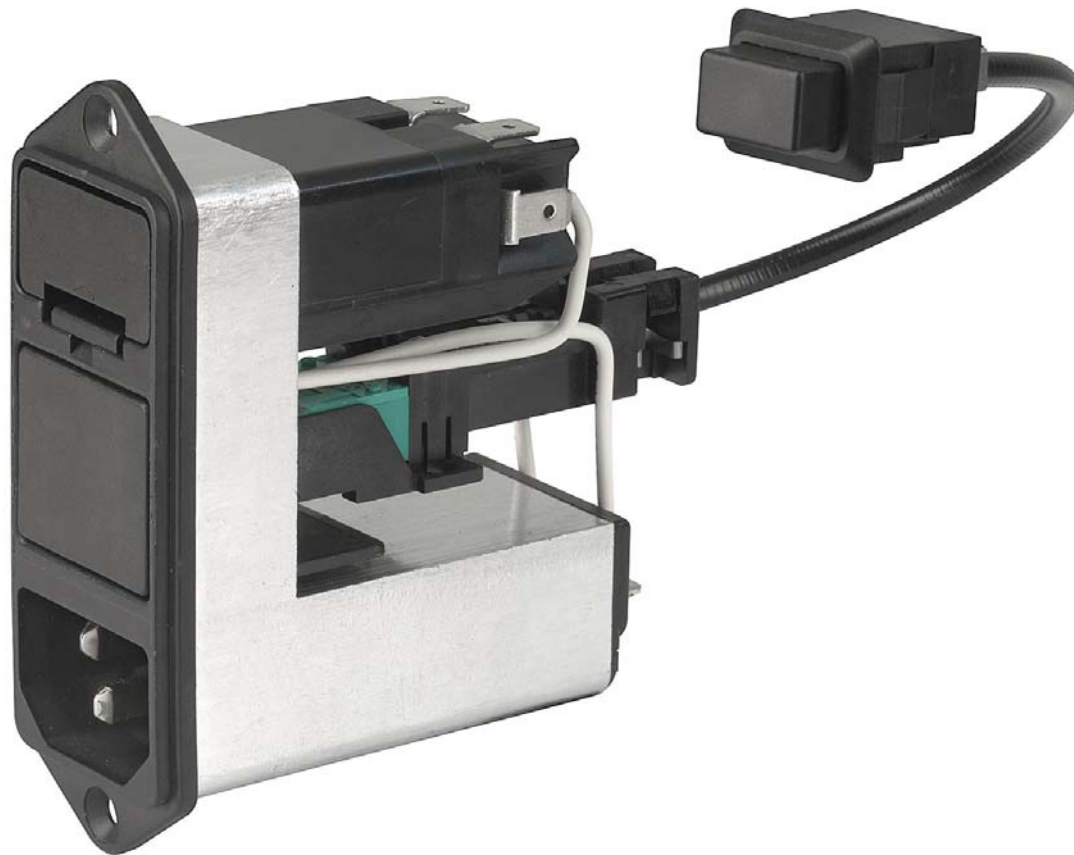
A hálózati szűrő csillapítása a frekvencia függvényében:



Hálózat csatlakozó beépített zavarszűrő vel



**Bowdenes
kapcsolóval
egybeépített
hálózati
csatlakozó
beépített
zavarszűrővel**



Megbízható, zavartűrő berendezések készítése

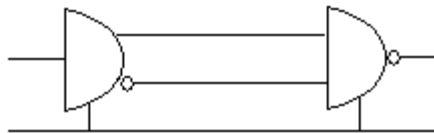
1. Jelbemenetek

Túlfeszültség védelem, zavaroszűrés

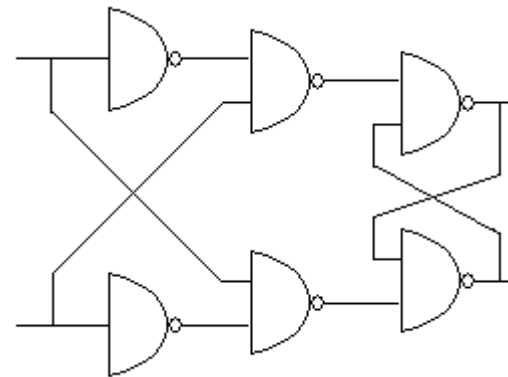
Feszültség és teljesítmény szint növelése

Galvanikus leválasztás, optikai szál alkalmazása

Differenciál módusú adó-vevő



Vonalmeghajtó áramkör



antivalencia áramkör

Áramkörök kialakítása

Zavartűrő alkatrészek alkalmazása

Csak a szükséges sebességű alkatrészek használata

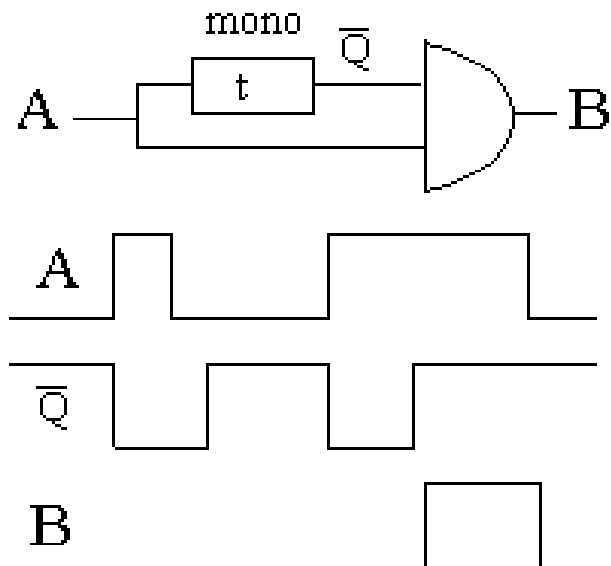
Hiszterézises komparátorok

Föld és táp vezetékek gondos kialakítása

Digitális rendszerek esetén rövid vezetékek használata

Jelkésleltetés, sávszélesség korlátozása

Redundáns rendszerek alkalmazása



Mikroprocesszoros rendszerek

Órajel vezeték gondos kialakítása

Tápfeszültség bekapcsolása és a program indulása

Tápfeszültség csökkenése

A programok felkészítése nem várt esetekre

Checksum ellenőrzés

Védelem program lefagyása ellen (watch dog áramkörök)

Ellenőrző mérések I.

Az elkészült berendezést zavarvédelmi szempontból ellenőrizni kell.

A méréseket az EN-MSZ 61000-4 szabvány írja elő.

Megnevezés	Okozója	Hatás	Vizsgálati módszer és szintek
Harmonics	Nem linearitások	hibás működés, melegedés	6..1.5% ; $0.2+0.5/n$ ($n>25$) THD = 8%
Interharmonics	konverter, ívkemence	hibás működés, melegedés	
Signal voltages	a. audio frekv. 110..2000Hz b. közép frekv. 3..20 kHz c. radio frekv. 20..150 (500) kHz		a. mint a harmonikusok b.c. kidolgozás alatt
Voltage fluctuation	terhelés változás, transzformátor szabályozás		$\pm 8\%$ vagy $\pm 12\%$ 2..3 sec / 5..10 sec ismétlődés
Voltage dips interrupts	zárlatok	kommutációs hiba, adatvesztés	30%, 60% 0.5..50 periódus 100% 0.5..50 periódus
Voltage unbalance	aszimmetrikus terhelés	forgó gépek túlterhelése	$\pm 2\%$
Power frequency variations	nagy teljesítmény változás, szigetüzem	mérési hiba, szinkronhiba	$\pm 2\%$ vagy $+4\% \dots -6\%$

Ellenőrző mérések II.

Megnevezés	Okozója	Hatás	Vizsgálati módszer és szintek
100/1300 us voltage surge	nagy teljesítményű biztosító kiolvadás	nagy energiatartalom, meghibásodás	1.3 Un; 0.1/1.3 msec
1.2/50 - 8/20 voltage-current surge	villám, zárlat, kapacitás bekapcsolás	meghibásodás	hibrid generátor Up/Ip 0.5..4 kV / 0.25..2 kA 1. - 0.5 2. 0.5 1 3. 1 2 4. 2 4
Fast transients	relék kapcsolása, SF6 és vákuum megszakítók	zavarok, hibás működés	50 nsec impulzus, 2.5 vagy 5 kHz ismétlődés, 15 msec burst, 300 msec periódus 0.5..4 kV

Ellen- őrző méré- sek III.

Megnevezés	Okozója	Hatás	Vizsgálati módszer és szintek
Ring wave	kapcsolási folyamatok	zavarok, hibás működés	100 kHz csillapodó lengés 0.25.. 4 kVp; 6/min 1. - 0.5 2. 0.5 1 3. 1 2 4. 2 4
Damped oscillatory waves	kapcsolási folyamatok nagy és közép feszültségű állomásokban	zavarok, hibás működés	0.1 és 1 MHz csillapodó lengés 40 vagy 400 / sec 1. - 0.5 2. 0.5 1 3. 1 2
High frequency induced voltages	kapcsolási folyamatok és zárlatok nagy és közép feszültségű állomásokban	zavarok	sweep 1kHz és 1 MHz között 20 periodus / 20 msec 1. 10 Vp 2. 20 Vp 3. 50 Vp 4. 100 Vp
Conducted radio frequency disturbances	ki dolgozás alatt		
10/700 us voltage surge	távközlő vezetékhez csatlakozó áramkörök	zavarok, meghibásodás	7/700 usec lökö hullám 0.5 .. 4 kVp
electrostatic discharges	sztatikusan feltöltött kezelő	zavarok, meghibásodás	feltöltött 150 pF kondenzátor kontakt levegő 1. 2 2 2. 4 4 3. 6 8 4. 8 15

Ellenőrző mérések IV.

Megnevezés	Okozója	Hatás	Vizsgálati módszer és szintek
power frequency magnetic field	erősáramú vezetékek mágneses tere	zavarok, hibás működés	folyamatos rövid 1..3 sec [A/m] 1. 1 - 2. 3 - 3. 10 - 4. 30 300 5. 100 1000
pulse magnetic field	villám áram	zavarok, hibás működés	8/20 usec [A/m] 1. - 2. - 3. 100 4. 300 5. 1000
damped oscillatory magnetic field	kapcsolási folyamatok nagyfeszültségű állomásokban	zavarok, hibás működés	<u>0.1 és 1 MHz</u> csillapodó lengés 1. - 2. - 3. 10 4. 30 5. 100
radiated electromagnetic field	rádió, TV adók, ipari berendezések	zavarok, hibás működés	modulált nagyfrekvenciás jel 25 .. 1000 MHz között [V/m] 1. 1 2. 3 3. 10