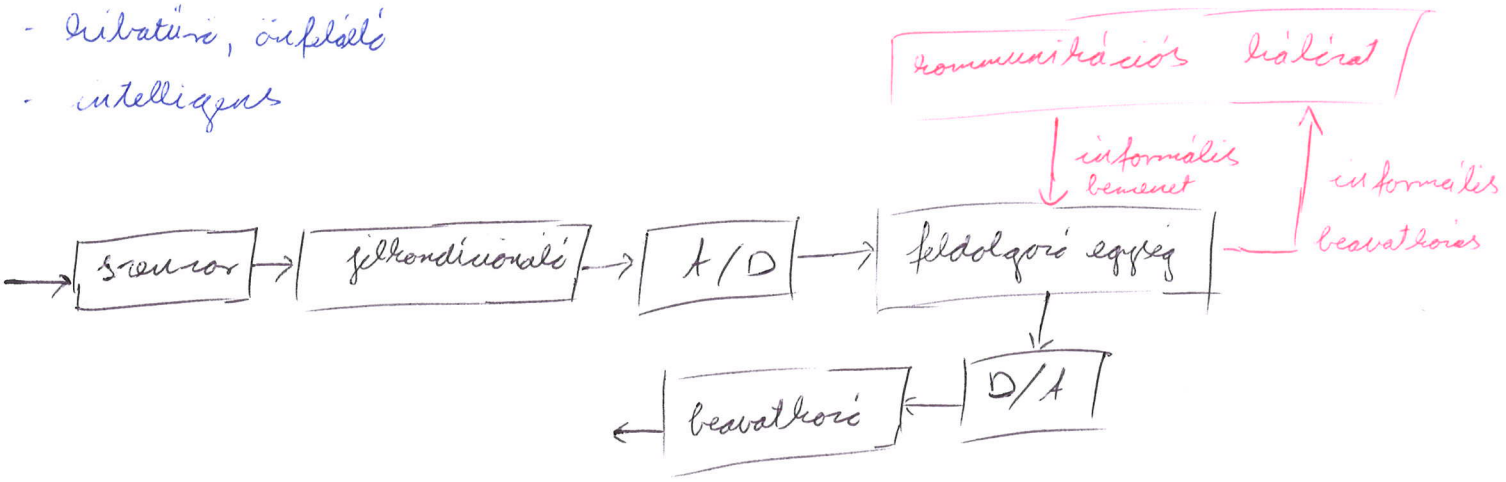
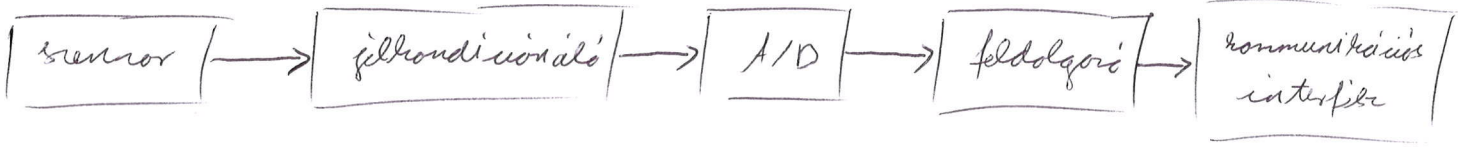
SZÉCHENYI MIT
1. EA
2011. 02. 08.

Beágyazott rendszer:

- teljesen autonóm
- intenzív kapcsolat a külvilággal (ez lehet fizikai vagy információs)
- reaktív, önfeledő
- intelligens



Intelligens szenzor:



MEMS: micro electromechanical system

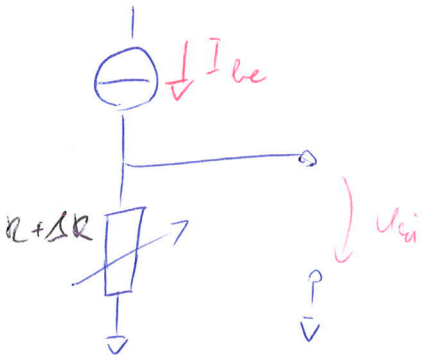
Mm integration: micromachined integration (mérjük alkatrészt a Si-n)

Jelkondicionálás

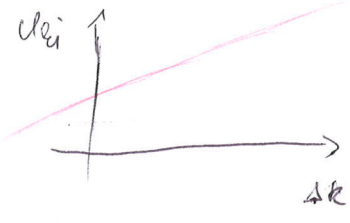
- erősítés
- AGC (automatic gain controll)
- zavarszűrés
- galvanikus leválasztás
- impedancia illesztés
- linearizálás

BEÁGY MIT
2.EA
2011. 02. 10.

Ellenállás érzékelők

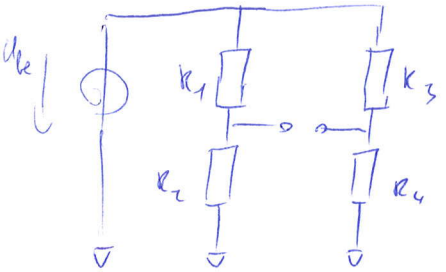


$$U_{ki} = I_{be}(R+\Delta R) = I_{be}\cdot R + I_{be}\cdot \Delta R$$



Csak differenciálisan lineáris.
Ha simplán A/D-re vezetnénk, sok bitet buknánk a nagy offset miatt

Hídkapcsolás



$$U_{ki} = \left(\frac{R_2}{R_1+R_2} - \frac{R_4}{R_3+R_4}\right)U_{be}$$

$$U_{ki} = \frac{R_2R_3 - R_1R_4}{(R_1+R_2)(R_3+R_4)}$$

$\emptyset$, ha $\frac{R_1}{R_2} = \frac{R_3}{R_4}$

$\Rightarrow R_x$ megfigyelhető!

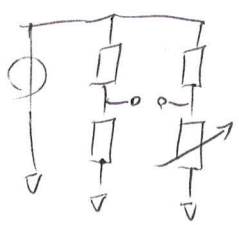
6,

Nem kell feltételeznünk kiegyenlítésre, kidimenzióból is próbálhatjuk megfigyelni

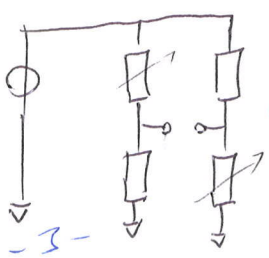
De egy változó ellenállás mellett maga a kód sem lineáris.

Rádásul a gerjesztés hibája is átmásolna. De ezt kompenzálhatjuk, ha az A/D referencia feszültsége is a kód hajtófeszültsége.

Az érzékenység ($\frac{\text{kimenet}}{\text{bemenet}}$) is túl kicsi

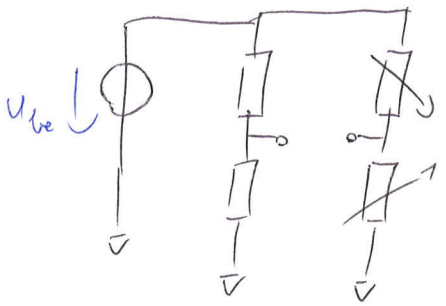


$$\Rightarrow \frac{U_{be}}{4}\,\frac{\Delta R}{R+\frac{\Delta R}{2}}$$

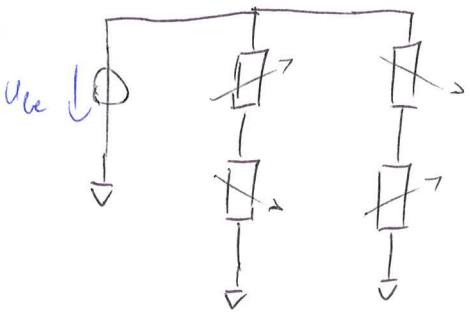


$$\Rightarrow \frac{U_{be}}{2}\,\frac{\Delta R}{R+\frac{\Delta R}{2}}$$

Ilyet nem mindig tudunk csinálni (érintkező kell, ami mindkét változást tudja), de ha megy, hiszen

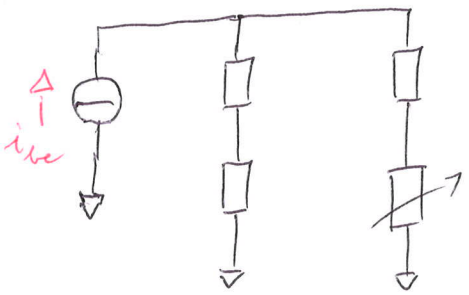


$$\Rightarrow -\frac{U_{be}}{2}\cdot\frac{\Delta R}{R}$$



$$= -U_{be}\frac{\Delta R}{R}$$

} De ettől még az átalakító elrontlatja a linearitást



$$U_{ki} = I_{be}\left(R\cdot\frac{2R+\Delta R}{4R+\Delta R} - \frac{2R}{4R+\Delta R}(R+\Delta R)\right)$$

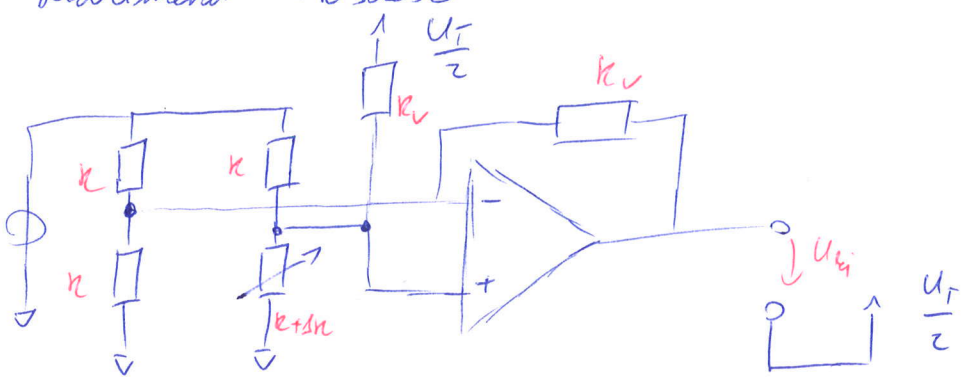
$$= -\frac{I_{be}R}{4}\,\frac{\Delta R}{R+\frac{\Delta R}{4}}$$

→ nem lineáris, de kevésbé, mint az első eset.

Áramgenerátoros kapcsolás: nem tűnik el a vezetéken, mint a fesz (ami keresik)

BEÁGY MIT
2. EA
2014. 02. 10.

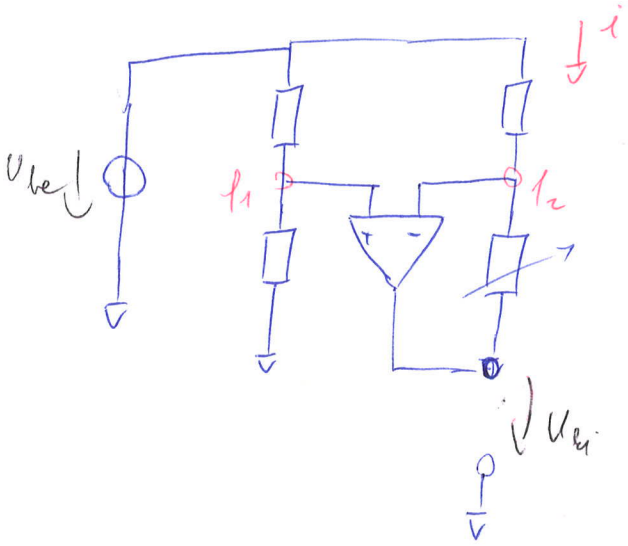
Kiáltiment erősítése



$$U_{ki} \approx -\frac{R_V}{R/2} \cdot U_{ki,\,kiid}$$

↑ igazából kis nemlinearitást visz be

Ha műszererősítőt használunk, már csak a kiid ad nemlinearitást



$U_{ki} = ?$

$$I_1 = \frac{U_{be}}{2} \qquad I_1 = I_2 \quad \text{(erősítő miatt)}$$

$$i = \frac{U_{be} - I_2}{R} = \frac{I_2 - U_{ki}}{R + \Delta R}$$

$$i = \frac{0{,}5\,U_{be}}{R} = \frac{0{,}5\,U_{be} - U_{ki}}{R + \Delta R}$$

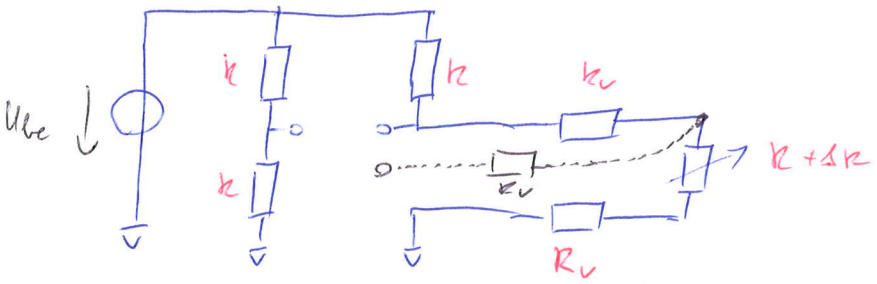
$$U_{ki} = -\frac{U_{be}}{2} \cdot \frac{\Delta R}{R}$$

Egy erősítővel ki lehet küszöbölni a nemlinearitást.

Van, hogy ez az olcsóbb. De olyan is, hogy nem fér el két szenzorral az érzékenység nő

Ha a szenzor is és a kiid is keményen nemlineáris, akkor nem sorbakötnék ezt, hanem digitálisan linearizálnám.

Mi van, ha messze van a szenzor?



$R = 350\,\Omega$

$R_v = 10{,}5\,\Omega$

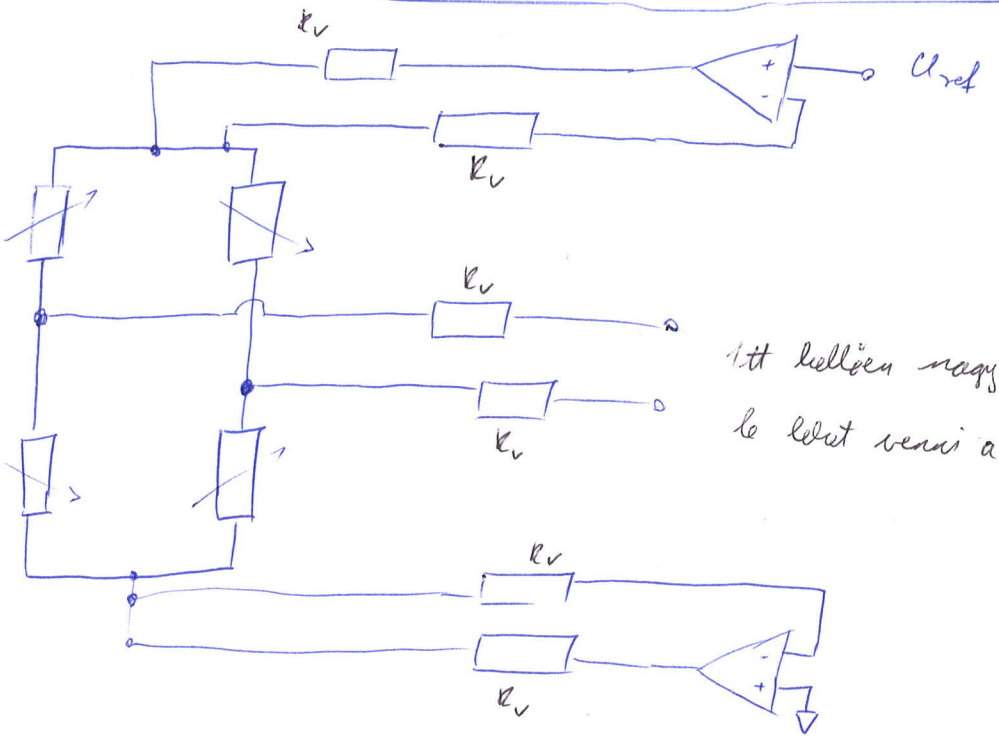
$\Delta R = 0{,}01\,R$

$U_{be} = 10\,V$

Ez elég durva offsetet okoz, ráadásul R_v hőfüggő, így nem is nagyon kompenzálható

A fenti megoldás viszont offset mentes és az erősítési hibája is elég kicsi. A bevételeket együtt vetve a mérnöki oldali hatás is kicsi

HÍ BE 'IGY
3.EA
2011. 02. 15.



6 vezetékes mérés

Itt kellően nagy bemeneti impedanciával le kell venni a jelet továbbítás nélkül

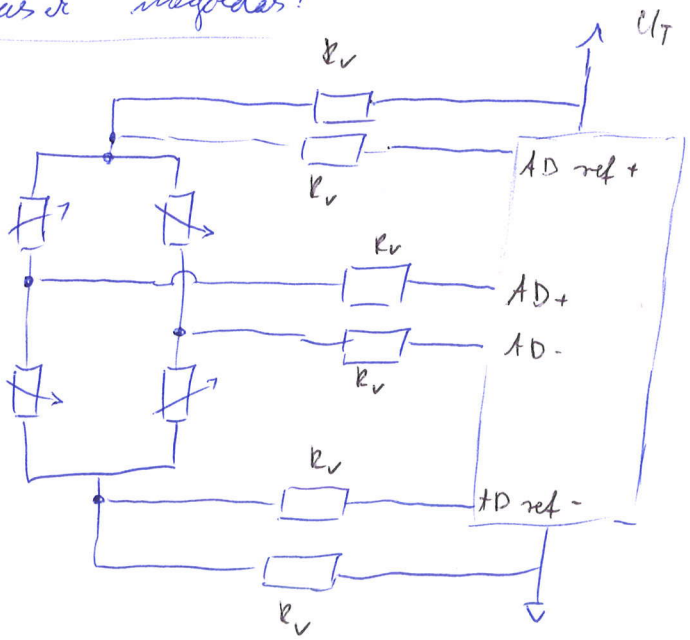
A műveleti erősítő húzza fel a feszt, így a kiad bemenetén U_{ref} lesz.

Egy erősítő a tápfeszt csak kb. 3 V-al tudja megkerülteni,

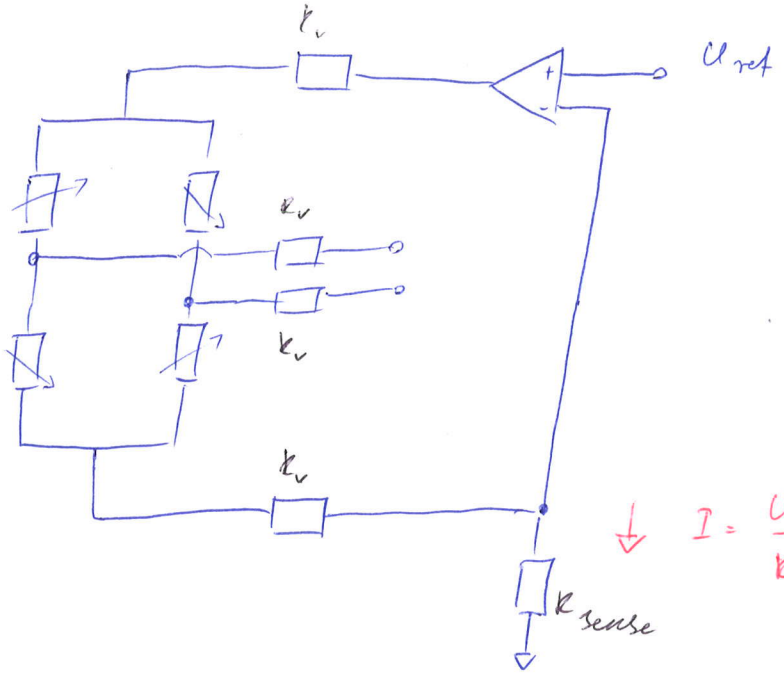
Kivéve: rail to rail erősítő.

A alsó erősítő táplálása negatív feszt igényel, hogy a kiadat földre tudja húzni

Más jó megoldás:



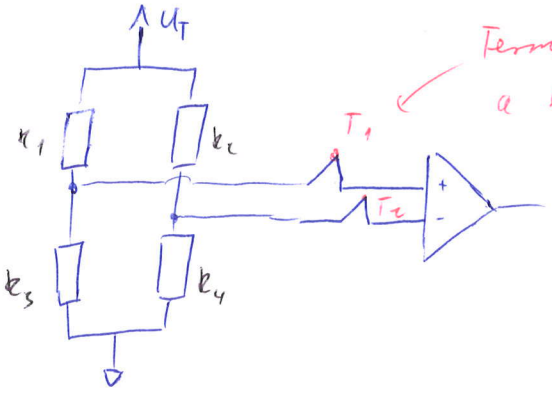
Ezek egy az egyben vannak megfelelő IC-k



Ez egy áramgenerátoros megoldás => kiesik R_v

$$\downarrow \quad I = \frac{U_{ref}}{R_{sense}}$$

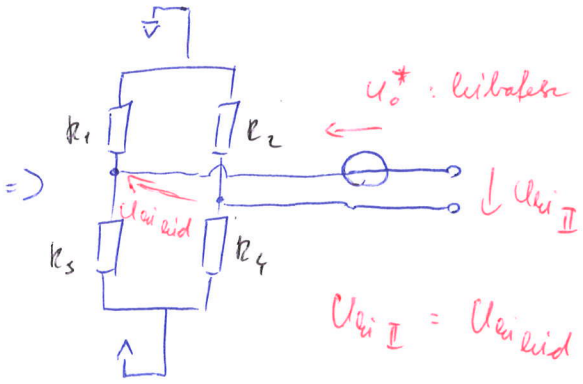
Kiolvasási hibák.



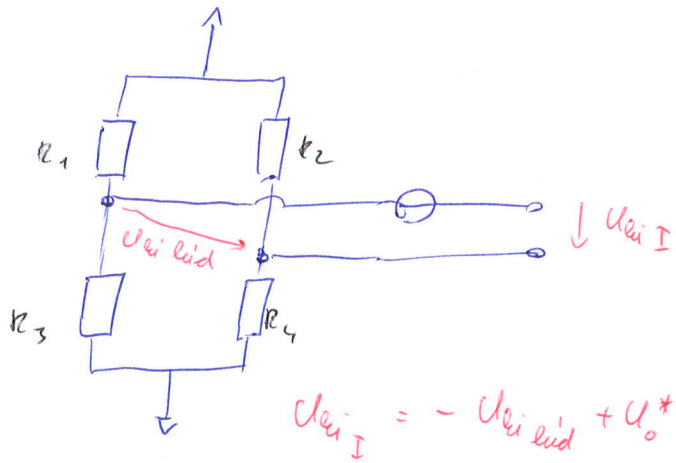
Termo elem: különböző fémek találkozásánál a kontaktusoknál feszültség alakul ki $T > 0$ K-en.

$$U \approx (T_1 - T_2) \cdot 35 \frac{\mu V}{^\circ C}$$

Ehhez jön még a bias áram és az offset feszültség



$$U_{ki\,II} = U_{ki\,kid} + U_o^*$$



$$U_{ki\,I} = -U_{ki\,kid} + U_o^*$$

Váltott polaritású gerjesztéssel és a két eredmény kivonásával kiejthetjük az összes offsetet.

BE1G4 MIT
3.E1
2011. 02. 15.

Aktív zajelnyomó rendszer szenzorhálózattal

Zajelnyomás:

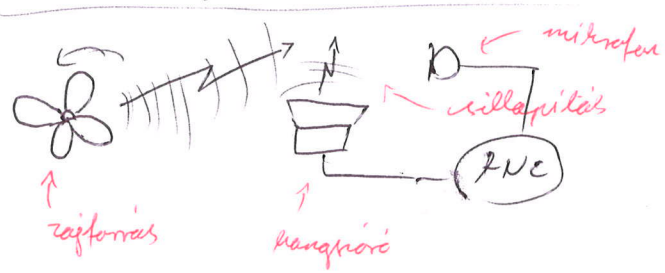
- passzív (pl hangszigetelés): költségesebb
- aktív elnyomás: ellenfázisú zajjal

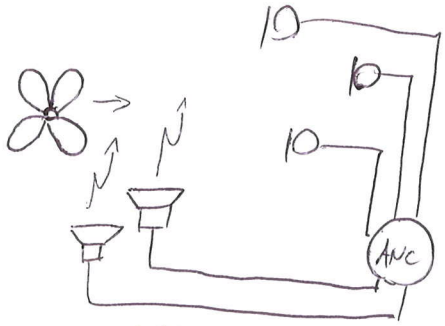
-> zajforrások:

- traktor, lengő ülés -> ellentétes mozgatás
- turbopropx hajtómű járási zaja -> aktív zajelnyomás
- légkondi zaja

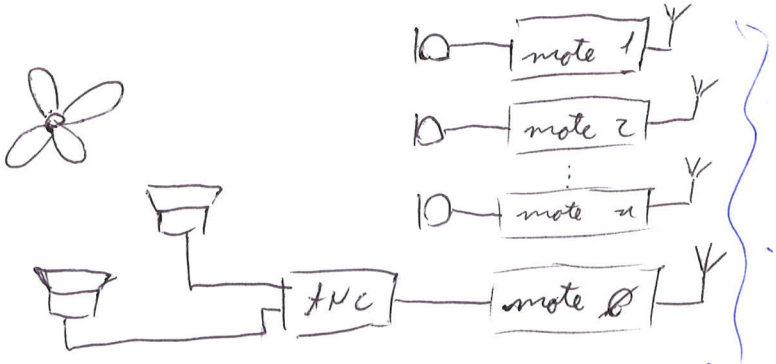
- Periodikus zaj ellen vagyunk jók, de csak kis térrészben és ha elrontjuk, jelentősen nő a zaj.
- ellenfázisú plusz zajt (energia) viszünk a rendszerbe
  - L> figyelni kell a rendszer terhelhetőségére
- több mikrofon, több hangszóró -> nagyobb térrészben tudunk elnyomni

ANC: active noise controll



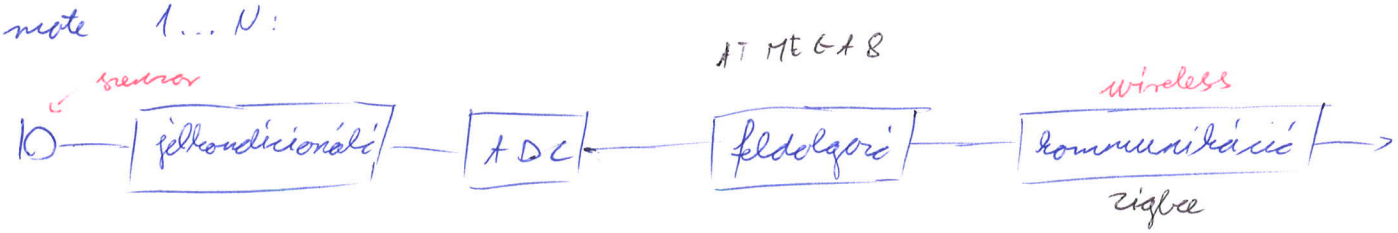


Jobb, de több vezeték és statikus

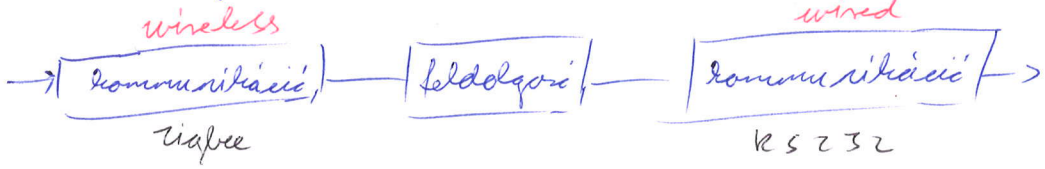


- szenzorhálózat, wireless kommunikáció
- intelligens beágyazott egységek (=> mote)
- dinamikusan alakítható rendszer

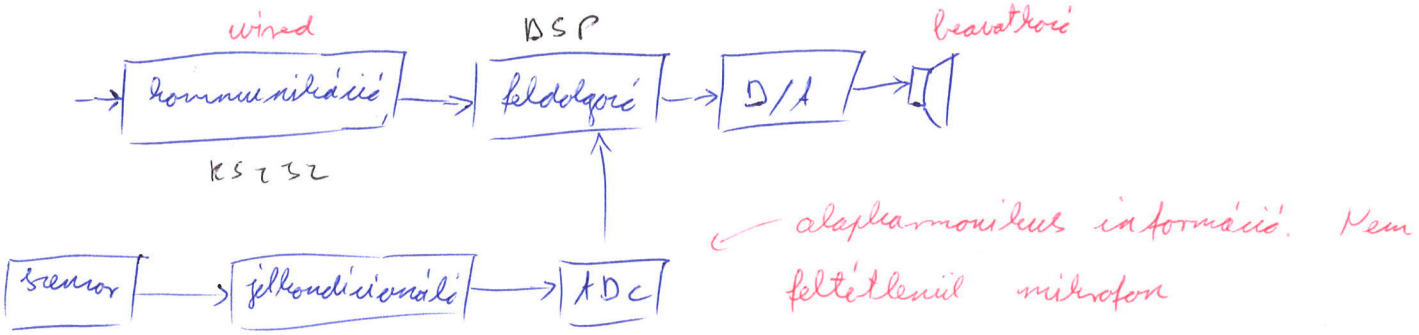
mote 1...N:

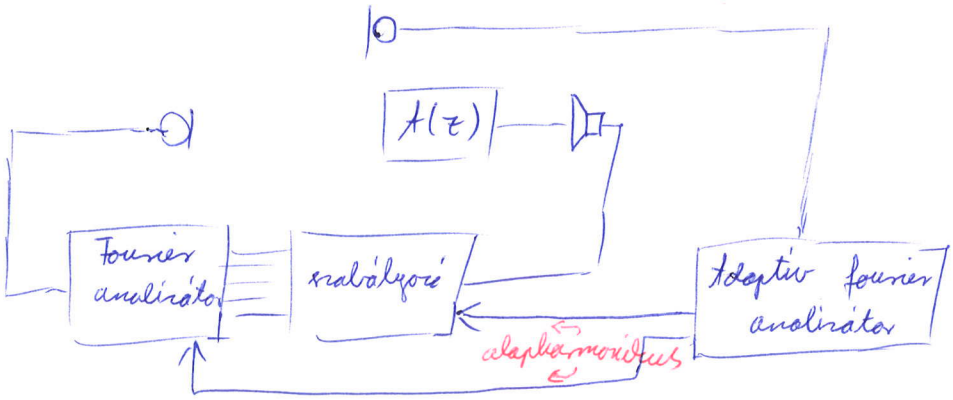


mote ∅: gateway, egyik rádiótól a másikra viszi az adatot



ANC:





BEÁGY MIT
5.EA
2016. 02. 22.

KOVÁCSHÁZY TAMÁS

Beágyazott rendszerben interruptból menekülünk

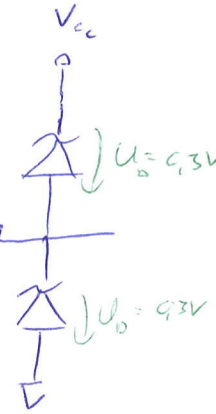
Az interrupt C-ben függvény

_BV(x) : x-val eltolást egy egyest

Hogy milyen procira fordítunk, az a make-file-ban van meg.

volatile : kulcsszó, ami utasítja a fordítót, hogy felejtse le, amit a változóról tud. Használat esetén mindig olvassa be a változót

A I/O pinek diódával fesvédettek, de áramkorlátozó ellenállás nincs => érdemes R ~1-2k vagy R használni



Van hardveresen újraindítható timer => input capture

Output capture : adott ideig számol a számláló, majd sikít

A/D átalakító : kell neki egy adott idő, amíg felveszi az értéket -> tartani kell az időzítést

BE GGY MIT
6.EA
2011.02.24.

7.EA

Microchip PIC 16F87x

- Harvard architektúra
- 8 bites adatbusz
- 14 bites címbusz (14 bites programmemória)
- RISC, 35 utasítás, egy utasítás 4 órajel
- csak belső mem, soros programozás
- 1 szintű interrupt

Periféria leírás

- Power-on reset: amíg nem stabil a tápfesz, reseteli
- Power up timer, watchdog timer
- Oszcillátor start-up timer
- 8 és 16 bites timer
- 10 bites ADC
- I2C, SPI, USART
- Brown-out reset: ha elmegy a táp, reset
- 8 szintű stack

Memória szervezés

0000H -> RESET 0004H -> INTERRUPT

- lapozás, memóriába ágyazott periféria címzés (-> volatile!!!)
- általános célú regisztered

BME AUT MIT
7.E.1
2011.03.01.

Portok felépítése

- TTL vagy Schmitt-triggeres bemenet
- weak pull-up (földre kényszeríthető)

Harvard architektúra: itt is van önmódosító program, csak nehezebb csinálni

Perifériák

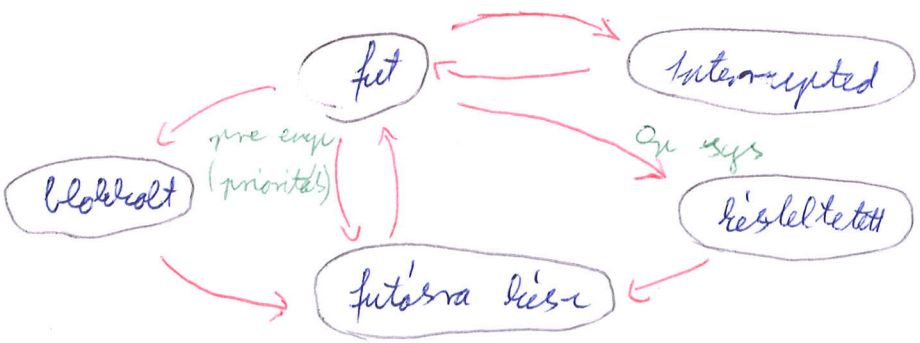
- 8 bites:
  - CLK vagy külső léptetés, előosztó
- 16 bites timer -> capture mód: kívülről olvasható
  -> output compare
- ADC
- időzítés

IT rendszer:

Watchdog timer -> clear watchdog timer
- túl kell méretezni a leállítgatás miatt

Szemaforok

Task állapot (önálló memóriával rendelkező folyamat)



- Futó utasítás adhat ki blokkoló -> és futásra készre váltó parancsot

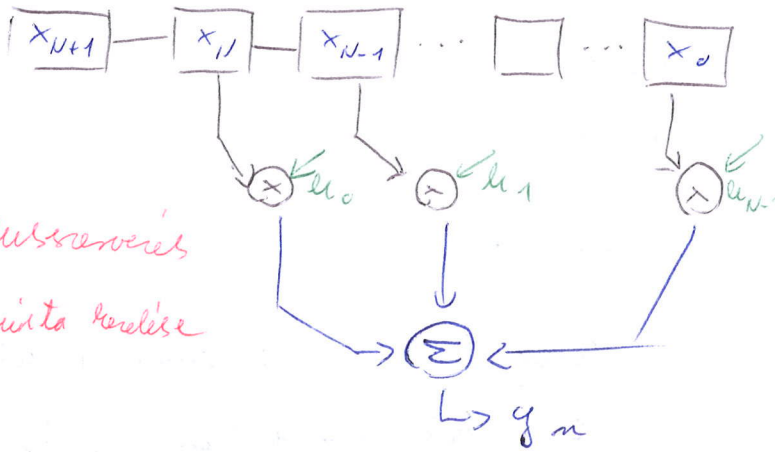
DSP

Motivációk:

- nagy számítási teljesítményt, adatfeldolgozást igénylő alkalmazások
  -> kép, hangfeldolgozás
- szűrés
- decimálás, interpolálás
- transzformációk (Fourier, Hilbert)
  ↳ gyorsabb, hatékonyabb feladatvégzés
  ↳ tömörítés, titkosítás
- moduláció / demoduláció

Konvolúció:

$$y_n = \sum_{n=0}^{N-1} x_n \, h_{N-n}$$



Cél:

- nagy sebesség
- sorozatok akkumulálása
- gyors címkiosztás
- utolsó N minta tárolása

Megoldások

- MAC: multiply and accumulate
- 1 utasítás alatt több memóriahozzáférés
- HW támogatott címkiosztás és ciklikus bufferelés
  => fixpontos címaritmetika

BME VIK MIT
8. EA
2011. 03. 03.

DSP-k folytatása:

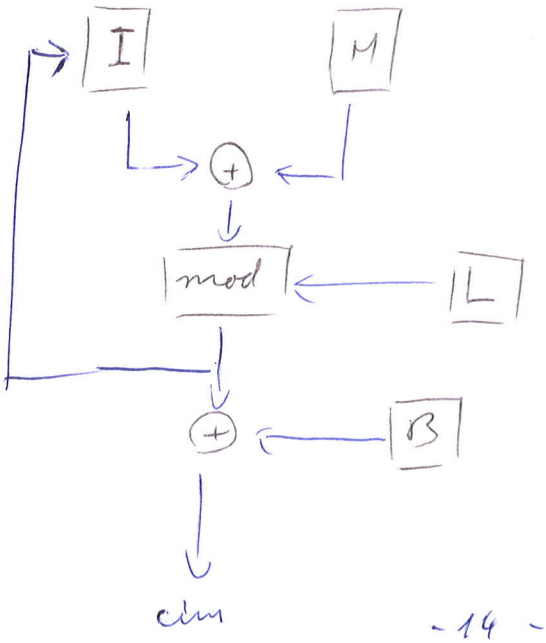
- elő kellett segíteni a konvolúciót
- aritmetikával párhuzamosan adatmozgatás
  (tömbös adatlekérés, az indexelést önmüködően végzi)
- van benne barrel shifter (tetszőleges számmal shiftel)
- ezen felül még ÚK DMA is nehet a ciklusok belül

Független címaritmetika (cirkuláris puffer)

- kijelölünk egy báziscímet (B)
- index regiszter címet bázisrelatívan
- ~~módosít~~ modify: hányasával végrálunk
- length: cirkuláris puffer hossza

$$cím = B + (I + M) \,\%\, L$$

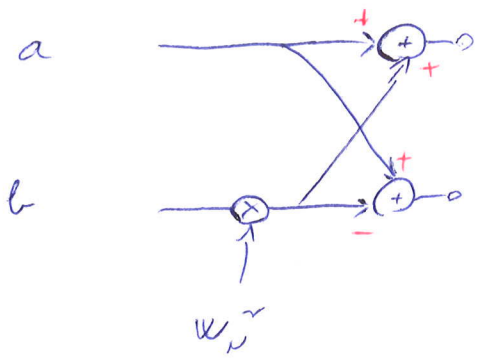
% = modulo



Gyors DFT: FFT

$$DFT: \quad X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}k\cdot n}$$

Pillangó: 2-es DFT, az FFT alapművelete



$$W_N^r = e^{-j\frac{2\pi}{N}r}$$

(komplex aritmetikát algebrai alakban (cos + j sin) kezelünk

DSP támogatja a pillangót

Kell még bit reverse címzés:

0 0	→	0 0
0 1	→	1 0
1 0	→	0 1
1 1	→	1 1

Ilyet is tud

Barrel shifter: osztás/szorzás 2 hatványaival

eredmények tologatása (skálázása) → mindig az értékes jegyeket tárolják

Hagyományos modell: az volt eddig

- külön adat és program busz (cím és adat állással) ← programadat

BÉÁGY MIT
8.E1
2011.03.03.

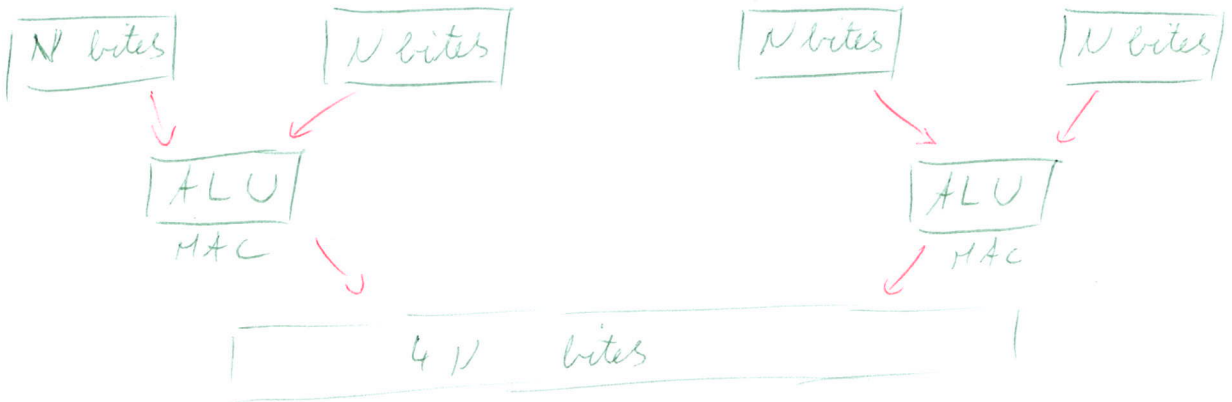
Analog Devices, SHARC:

- konfigurálható fix / lebegőpontos architektúra
- dual port memóriák

További sebességnövelési lehetőségek

- órajel → nagy fogyasztás, melegedés
- több processzor, párhuzamos struktúrák (transputer) — teljes csövesítés — több PC irányból
  → azt is megismételjük többször, ami nem kell ⇒ drága
- SIMD: Single Instruction, Multiple Data — Ugyanazt a műveletet kell sok adaton elvégezni. konvolúció, FFT
  Csak a számító blokkot ismételjük. (több magos CPU, egy dekóder)
- Pipeline

A SIMD volt a befutó → javított hagyományos DSP



VLIW: Very long Instruction Word (egyfajta SIMD)

- megjelenik, mely kis részutasításokat végezhetők el egyszerre és ezeket összefordítjuk egy műveletbe (compiler csinálja)

Akár 128/256 bites

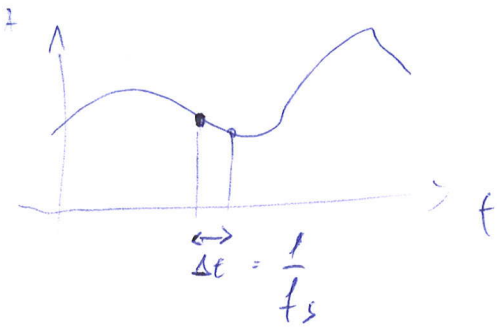
Szuperskalár architektúra:

- sok műveletvégző egység
- futási időben azonosítjuk a párhuzamosan elvégezhető műveleteket
- nem megfelelően előre a futási idő

=> DSP-ben ez ciklus

Megjelennek a specializált DSP-k.

Mintavételezés:

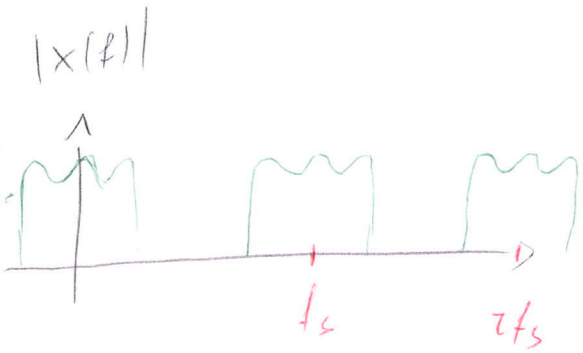


- ekvidisztáns mintavételezés: Δt konst
- ha az órajel ugrál => jitter
  => additív zaj

Csak számra egyenlő

Shanon tétel: B sávkorlátozott jelre a visszaállításhoz szükséges $f_s \geq 2B$

Többszörös mintavételezés: jelleg és hatás jelsor



← kalmerati spektrum

Egy jel nem lehet egyszerre sáv és időkorlátos !

Kvantálás:

csonkolás

kerekítés => kisebb hiba

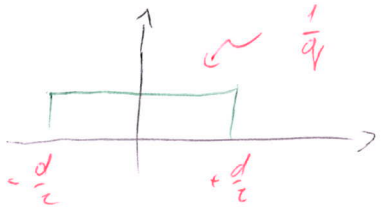
BME VIK MIT
Q.E.I
2011. 03. 08.

- kerekítéses kvantálás: az eredeti és a kvantált jel várható értéke egyezik
- A kvantálás nem invertálható => nem egyértelmű

kvantálási modell: kvantáló lineáris zajmodell

- matematikailag jól kezelhető
- x és n_q korrélálatlan
- kvantálási zaj egyenletes eloszlású

- n_q legyen fehér zaj (kis kvantum lépcsőkre teljesül)



=> $E(n_q) = 0$

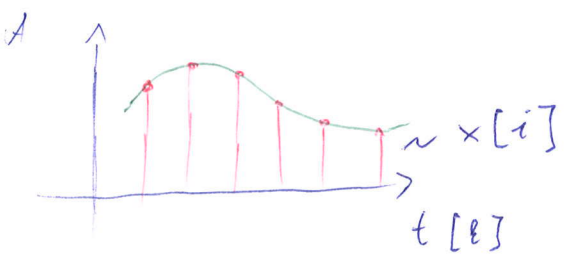
kvantálási tétel: Ha a jel valószínűség-sűrűség függvénye korlátos att, ahol $\frac{1}{\tau_B} >$ kvantum lépcső, akkor a jel statisztikus tulajdonságai helyreállíthatók

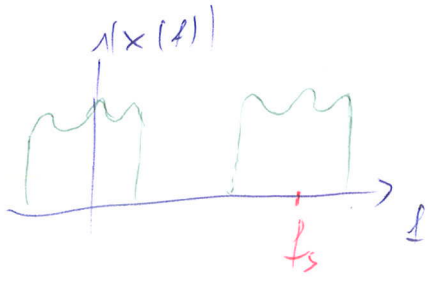
$\frac{1}{\tau_B} > q$

Decimálás:

- Mintavételi frekvencia csökkentése

=> $\downarrow M$: f_s csökkentése M-es faktorral





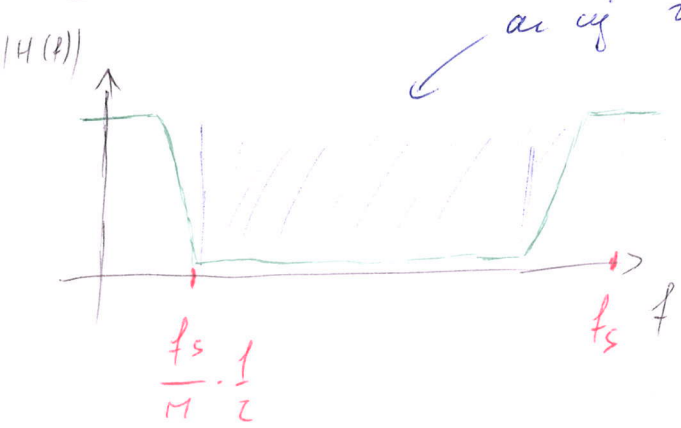
Decimálás: csak minden $x[k \cdot M]$ marad meg

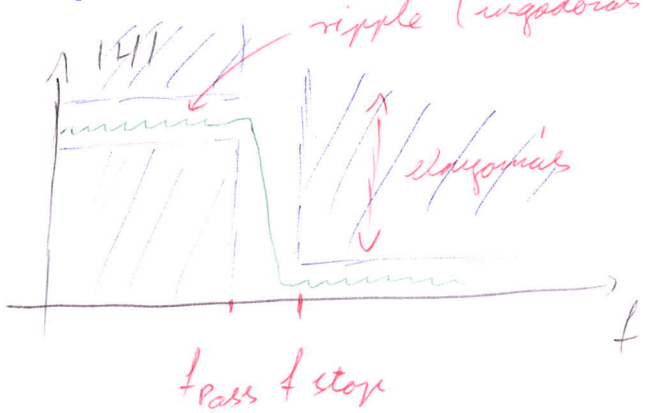
$$\Delta t' = M \cdot \Delta t = M \cdot \frac{1}{f_s} \qquad f_s' = \frac{f_s}{M}$$

egyed: sűrűbb átlapolódás (az eredeti leolvasási spektrum beszűrődik)

Tipikusan: mintát eldobása + aluláteresztő szűrő

Átlapolás gátló szűrő:

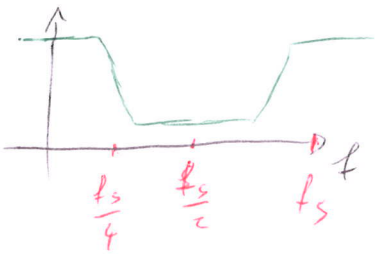




Ezeknek kell: $f_{pass} = \frac{f_s}{M} \cdot \frac{1}{2}$

Ez az egész decimálás mintavételezés után van!

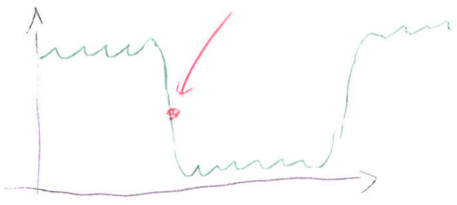
Az átmeneti tartomány specifikációi kérdése



$\downarrow 2$ => féleli szűrő

Féleli szűrő: half-band filter

tulajdonság:



ennél a pontnál antiszimmetrikus

=> minden 2. szűrőegyüttható 0.

kisebb számítási igény, de tovább az átmeneti tartomány

BÉLAY MIT
10. EA
2011. 03. 10.

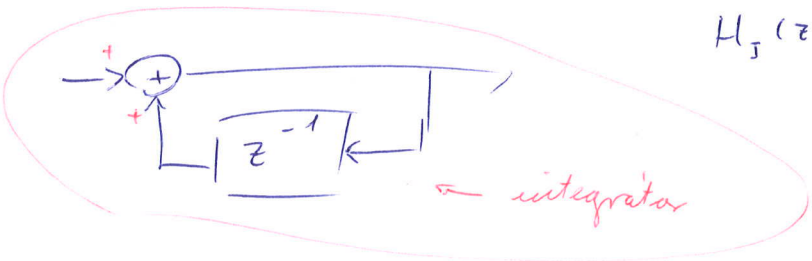
Cascaded Integration Comb (CIC)

(kaszkád integrátor fésűs szűrő)

Mozgóátlag átlagolás:

$$\sum_{i=k}^{k+M-1} x(i) = \sum_{-\infty}^{k+M-1} x(i) - \sum_{-\infty}^{k-1} x(i)$$

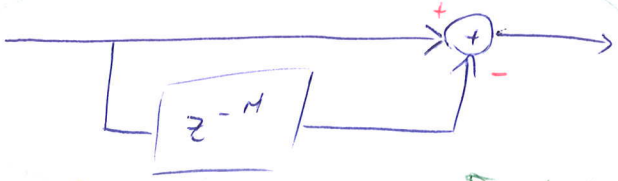
I:



← integrátor

$$H_I(z) = \frac{1}{1-z^{-1}}$$

D:



← derivátor

$$H_D(z) = 1 - z^{-M}$$

$$H(z) = H_I(z) \cdot H_D(z) = \frac{1-z^{-M}}{1-z^{-1}}$$

← a késleltetett integrált vonom ki

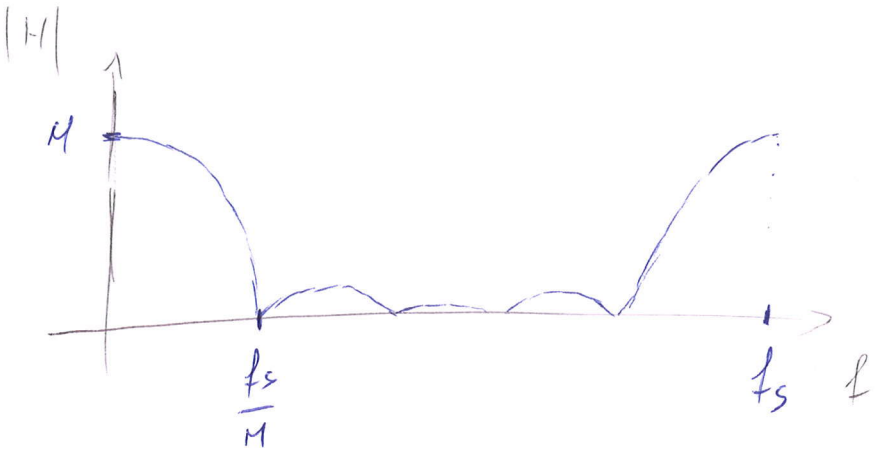
$$\sum_{i=0}^{M-1} z^{-i} = \frac{1-z^{-M}}{1-z^{-1}}$$

$$z = e^{j2\pi f \Delta t} = e^{j2\pi \frac{f}{f_s}}$$

$$\frac{1-z^{-M}}{1-z^{-1}} = \frac{e^{-j2\pi f M \frac{\Delta t}{2}}}{e^{-j2\pi f \frac{\Delta t}{2}}} \cdot \frac{e^{+j2\pi f M \frac{\Delta t}{2}} - e^{-j2\pi f M \frac{\Delta t}{2}}}{e^{+j2\pi f \frac{\Delta t}{2}} - e^{-j2\pi f \frac{\Delta t}{2}}}$$

$$= e^{-j\pi f (M-1)\Delta t} \cdot M \cdot \frac{\sin(\pi f M \Delta t)}{M \cdot \sin(\pi f \Delta t)}$$

← diszkrét sinc
A folytonos, f_s-el eltolgatva és halmozva



Szűrőnek vagyok, de a lesűrési pontom ($\frac{f_s}{M}$) bármi is a csillapítása => adott savat (pl. 50Hz) kiszűr

n darab sima szűrő egymás után => sima^n szűrő

Szélesíthetjük a lesűrési tartományt

sima szűrő — decimáló

A sorrend felcserélhető, mert lineáris műveletek

Sima 3:

az előző pld példára $K = M^3$

modulé aritmetika kell mellé

növekvő szélesség

csökkenő szélesség

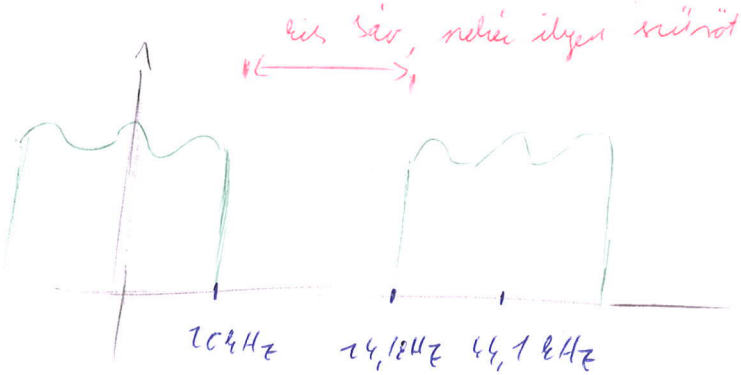
decimálás = : átlagolás?

BME-AUT MIT
10. EA
2011. 03. 10.

Interpolációs: a mintavételi f növelése

D/A átalakításra került jel ismétlődik

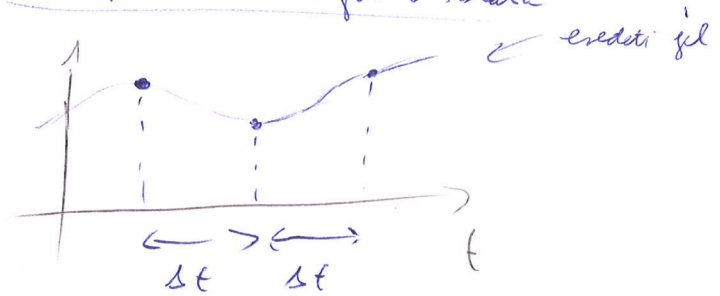
CD: $f_s = 44{,}1$ kHz

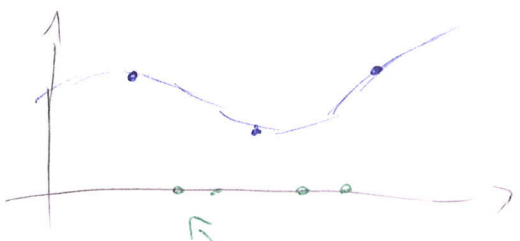


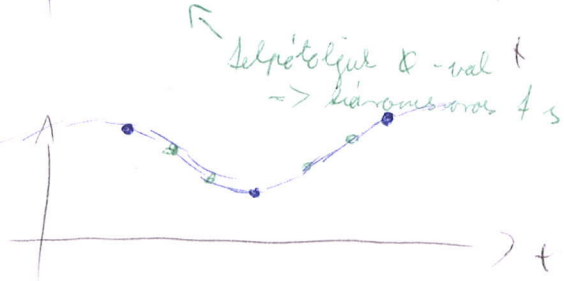
=> Túlmintavételezünk, és így raknak ki

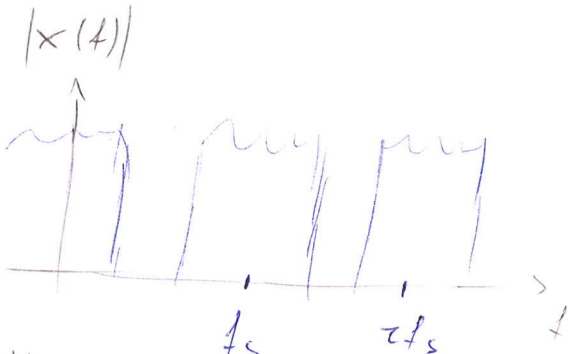
=> $n \cdot f_s$-nél lesz a spektrum körüli hely, amit jóval könnyebb lenyelni

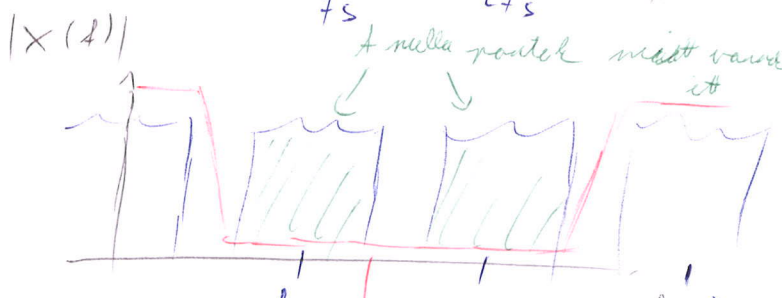
Interpolálás megvalósítása

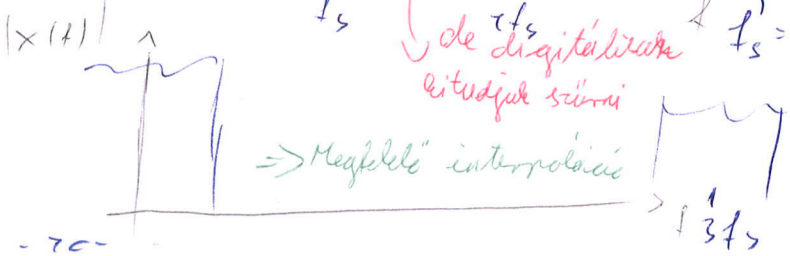












eredeti jel

ez a sűrűség ?
=> ezek tudják kiszámolni az interpolált y-okat

$h_0\ h_1\ h_2\ h_3\ h_4$

$h_0\ h_1\ h_2\ h_3\ h_4\ h_5$

$h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6$

sűrű szűrő együtthatói

$\{h_0, h_3, h_6, \dots\} * x_i \to y_i$

$\{h_1, h_4, h_7, \dots\} * x_i \to y_{i+1}$

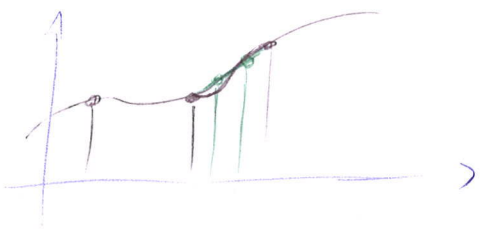
$\{h_2, h_5, h_8, \dots\} * x_i \to y_{i+2}$

Polifázisú szűrő

így kapjuk meg a megfelelően interpolált jelet

(kiszámolhatjuk a bemenetbe redundanciát, hogy ne szorozzunk nullával felesleges.

Lineáris interpolációt



frekvencia: diszkrét sinc

diszkrét sinc

=>

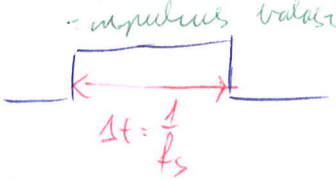
diszkrét sinc^2, ami a lineáris interpolációt adja => nem jó

=> Ne használjunk lineáris interpolációt

DAC:

nulladrendű tartással: Δt ideig tartja az értéket

=> impulzus válasz



$\xrightarrow{\mathcal{F}}$ sinc (folytonos)

=> jó ismétlődés elvonás, de kis frekvenciát torzít. Ezt ellenkompenzálhatjuk

BE'A-Y MIT
11.EA
2011. 05. 17.

FPGA

Felhasználói specifikáció -> előrefele rendszertervező ->
-> funkcionális rendszertervező -> logikai tervezés

Egyszerű PLD eszköz (Programmable Logic Device):

- PAL: programmable array logic
- PLA: programmable logic array
- GAL: programozható architektúra

A programozható logikák egyre általánosabb elemekből épültek fel.
És a hatékonyságot rontja, de növeli a használhatóságot.

Másik programozási irányból MPGA: elő FPGA

Felhasználói programozhatóság:

OTP: csak egyszer
- kis méretű, jó kapcsolók, átégetéssel programozható

FPGA: minden indításkor

SRAM FPGA:
- minden indító memóriában van
- olcsó, bármikor átírható, jellemzően CMOS technológia
- de módosulhat (pl. kozmikus sugárzás) -> kritikus alkalmazásnál figyelni kell

Hogy hogyan bootol az FPGA, azt a mode választók dönti el.

FPGA:

I/O blokkok, logikai blokkok és programozható összeköttetések

Logikai cella granularitás szerint:

finom durva

1T NAND2 MUX4 LUT4 1/2 CAL

↑ 1 tranzisztor szintű programozás. Már elhalt

Mindig a feladathoz megfelelő komplexitást kell választani.

Egy mai logikai cella:

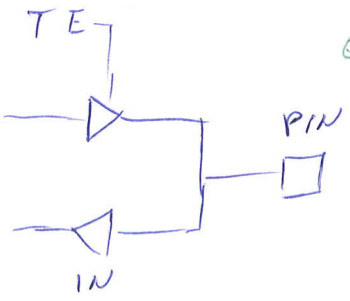
- átviteli logika, soros (tömbsorosítást is lehet), memória

I/O

Ma elég sokféle jel/tápfeszültség van, ebből egy esetben jellemzően többet is használ.

=> ezt az FPGA-nak mind tolerálni kell

-> egyben több funkciós, konfigurálható lábak



alap I/O, a maiak ennél jóval bonyolultabbak, kb. 15-20 regisztert tartanak egy lábhoz

BE IGY MIT
11. E 1
1911. 03. 17.

Jellemzők:

- többnyire hierarchikus
- okosan kell vele tervezni (a programnak)
- nem foglhat el.
- a CHIP felületének 80% lenul és kapcsolá
- erősen belekerülhet a sebességbe

ASIC: célspecifikus IC, egyedi tervezés és gyártás.
Csak nagyon nagy tételben éri meg ...

BME IIT MIT
12. Cl
2011. 03. 22.

A μ elektronika technológiai korlátai

Digitális rendszerek tervezése FPGA áramkörökkel
SRAM alapú FPGA architektúrák

- Alapvető erőforrások:

  - logikai blokk; I/O blokk; összekötés

- Felhasználói szinten csak az első 2 jelenik meg
- a összekötés nem direkt erőforrás, de bizonyos esetekben az erőforrás értékének igen fontos eleme

Belső felépítés:

- SRAM FPGA minden LUT alapú (kivétel MCT)
- a lut egy 1 bites memória, mélysége (címtartománya) változó
  4 vagy több bemenetes szokott lenni

T Vannak igények, amit általános erőforrásokkal is meg tudnak valósítani, de még jó, ha van célerőforrás.
Vannak erőforrás, amiket nem célszerű illeszteni a sebesség miatt,

BRAM -> blokk RAM - belső FPGA RAM

MULT -> szorzó

CDCM -> órajel menager, globális órajelgenerátorok kezelője

Spartan 3

CLB felépítése: - megosztott elrendezés
- független belső blokkok
- hasznos extra memória funkciók

VEGY MII
17.E1
2011.03.22.

1. A mikroelektronika technológiai korlátai

CLB erőforrásai:

1 LUT 4 + 1 FF

- az eredeti koncepció nem változott → alap erőforrás: logic cell

LUT 4 van benne:

- négy változós fgv, de csak egy változóra korlátmentes
- működési idő bement

FF: élvezérelt, ↑ és ↓ is lehet

Gyors átvitelképzés:

- az aritmetikai műveletek hatékony alapeleme, gyors carry logika

előtte: CLA: array look-ahead adder
CS: array select adder

Soros logikával:

- a soros erőforrásigényes
- összegzés FA

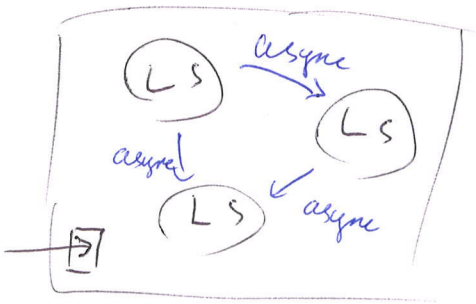
Tömbszorzó

aritmetikai MUL művelet → belső multiplexerek

Memória alapú CLB

BRAM

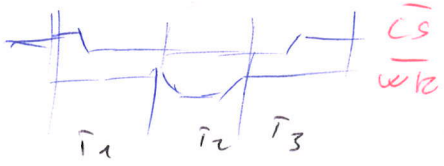
- GALS: globálisan aszinkron, lokálisan szinkron
- szinkron: teljesen
- teljesen aszinkront nem használunk



- szigorúan kvázilokális
- a szinkron része szigorúan szinkron

SRAM írás: $\overline{CS}$ és $\overline{WR}$ ellenőrzi, egy órajellel

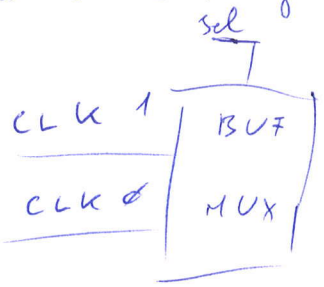
elvégeztatva:



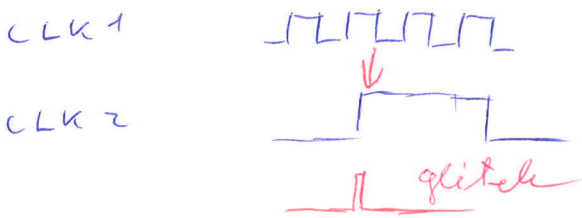
- vannak dual port SRAM-ok
- SKL 16: 16 bites shiftregiszter

4 fajta időzítés

- a rendszer gond nélküli működésének biztosításához



a gitek nélkül kapcsol át:



hibásan miatt van

=> késleltet. De nagyobb sebességnél ez már nem meg

- frekvencia szintézer

Folytonos Fourier.

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi ft}\, dt$$

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{j2\pi ft}\, df$$

$$X(k \cdot \Delta f) = \sum_{i=0}^{N-1} x(i\Delta t)\, e^{-j2\pi k \cdot \Delta f \cdot i \cdot \Delta t}$$

$$\Delta f = \frac{f_s}{N} \qquad \Delta t = \frac{1}{f_s}$$

$$X(k) = \sum_{i=0}^{N-1} x(i)\, e^{-j2\pi \frac{k \cdot i}{N}}$$

$$x(t) = \sum_{i=0}^{N-1} X(i\Delta t)\, e^{-j2\pi \frac{k \cdot i}{N}}$$

$\frac{N}{2}$-es DFT

$$= \sum_{\ell=0}^{\frac{N}{2}-1} x(2\ell)\, e^{-j2\pi \frac{k\, 2\ell}{N}}$$

$$+ \sum_{m=0}^{\frac{N}{2}-1} x(2m+1)\, e^{-j2\pi \frac{k(2m+1)}{N}}$$

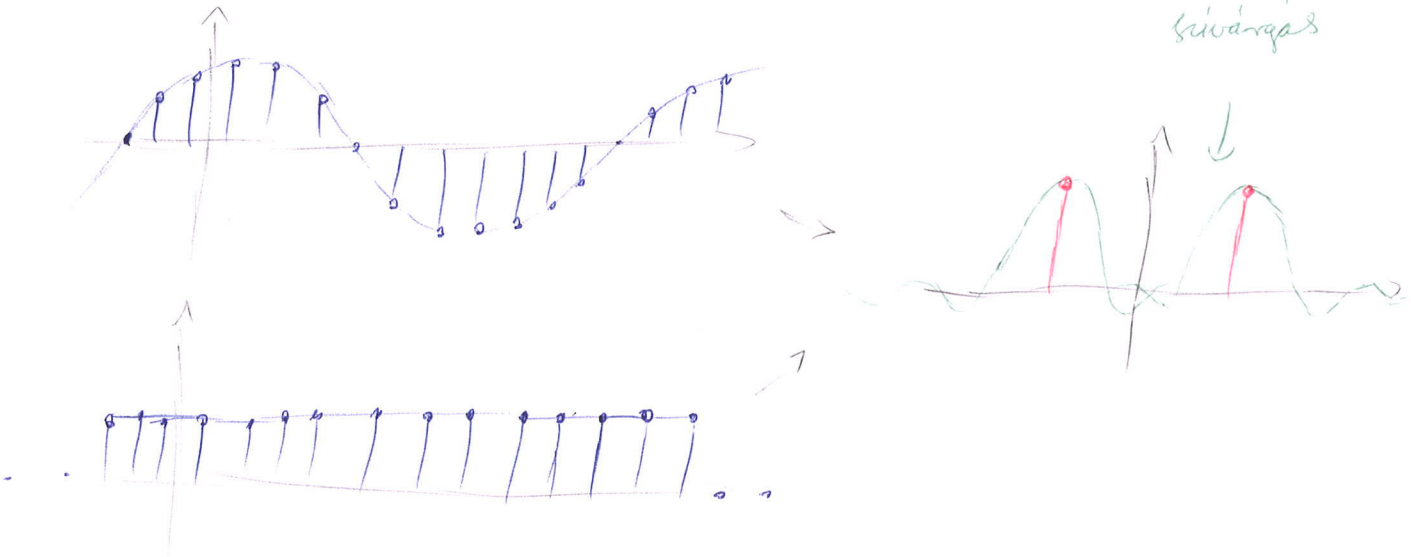
$\frac{N}{2}$-es DFT

Eltérések a folytonos fouriertől:

- spektrum átlapolódás: mintavételi frekvenciát tartani kell, levenetet szűrni kell
- szivárgás: véges mintaregisztrátum miatt lép fel, ami gyakorlatilag a jel egy részének kiablakolását jelenti
  => az ablak spektruma konvolválódik a jelspektrummal

→ Fourier spektruma sinc

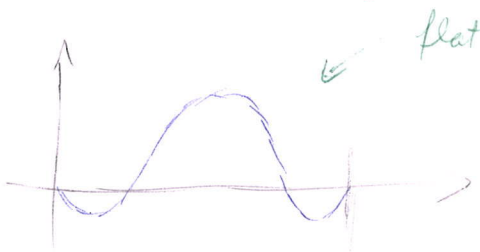
négyszögjelből szinusz jelet vágva:



De ha koherens a mintavételezés, akkor a sinc lesimításai pont úgy esnek, hogy jó lesz a DFT

Másrészt ha nem koherens a mintavételezés, a csúcs sem esik DFT gridre → hiba, f-ben és t-ben is

→ használjunk jobb ablakokat



4× szélesebb főhullám, lassú oldalhullámok
→ jó amplitudó becslés, nagy szivárgás

Rideg tenne effect.

- diszkretizálás hatás: ami nem gridre esik, az ki van takarva ⇒ fontos jelek eltűnhetnek
- ↑ DFT felbontásával javul a dolog

BE SGY MIT
13.E1
2011. 03. 24.

Szűrés frekvencia tartományban

-> spektrummal való szorzás

De diszkrét tartományban, DFT-vel számolva cirkuláris lesz a ~~konv~~ konvolúció, mert a DFT periodikusnak gondolja a jelünket.

=> ki kell egészítenünk a jelünket τN hosszúra nullákkal és akkor nem ér vissza a szűrő hatása

[Magában a DFT-ben nem kell f_s, csak a visszaalakításkor.]

Alulmintavételezés:

Periodikus jelet jóval alulmintavételezve, de digyelve a Kondorcet fáziselőrejelzésre -> elvidiótálás -> lejön a jó jel, de lassabb frekvenciával fog tűnni.

Egy mérésből nem lehet tudni, hogy ez történik-e -> időalap váltogatás mellett ha szélesik a jel, akkor ez volt a helyzet

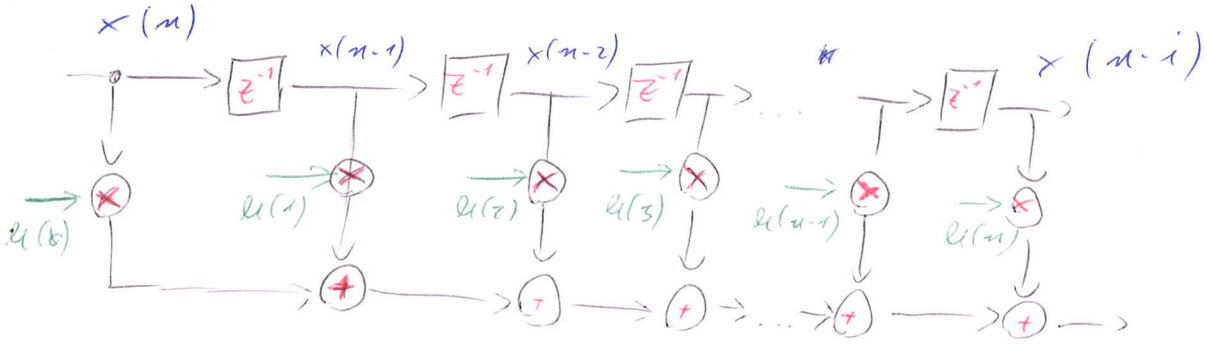
BEÁGY MIT
15.EA
2011. 03. 31.

Digitális jelfeldolgozás

lineáris sorrat <=> konvolúció

$$g(n) = \sum_{i=0}^{N-1} h(i) \cdot x(n-i)$$ ← ez konvolúciós összeg

N: tap szám



N darab késleltető, N szorzó és N-1 összeadó kell

portolás: a compilert nem szokták újraírni.
helyette a compiler kimeneti kódját fordítják tovább
=> portolás

Nem csak gyári procit lehet FPGA-ba ültetni, csinálhatunk sajátot. de lehet másodlagos procit, lehet emulálni, vagy újat csinálni

DSP-nek minden ciklusban kell:

- felvárni az operandust
- dekódolni
- műveletet végezni
- adatot kimenteni
- I/O műveletekkel foglalkozni

konvolúció végrehajtói:

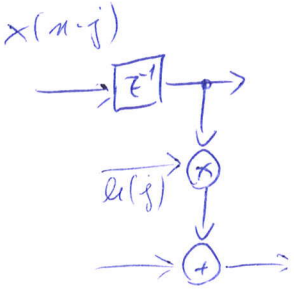
z^{-1} -> regiszter

(x) -> párhuzamos szorzó

(+) -> párhuzamos összeadó

Több adatrész és műveleti blokk segítségével növelhető fel a műveleti sebesség

Az FPGA-kban megjelentek a kapcsolható DSP blokkok

Ha minden 

blokkot példányosítunk, akkor a szűk keresztmetszet a jelterjedési sebesség lesz, mert 1 szorzású és N-1 összeadású idő kell neki (alsó számú sor)

=> nem tolható fel a f órához, helyette pipeline

Pipeline: nő a késleltetés, de jobb mintavételi f.

Ezt úgy érjük el, hogy a moduloket $\boxed{z^{-1}}$ elemekkel kapcsoljuk szét (mondjuk M darabbal). Ekkor a jelterjedésnek csak addig kell elfutnia, így

$t_{szorzó} + \frac{N}{M} t_{összeadó}$ lesz a jelterjedési sebesség.

Viszont nem lesz kész 1 órajel alatt, késni fog.
De nagy tap szám mellett nagy mintavételi f

=> lehet FPGA-t használni DSP-ként

BME MIT
15.E.A
2011.03.31.

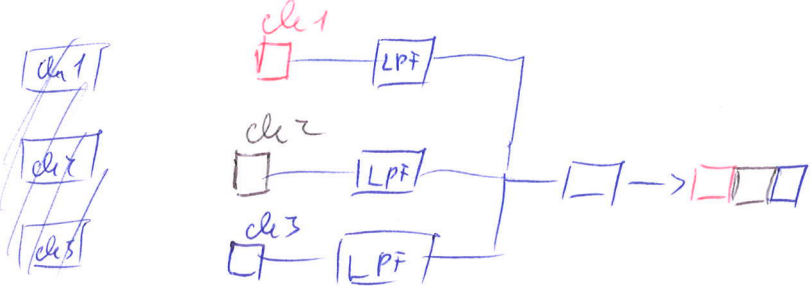
De processornál az adatot a MEM-ből kell előáratni és oda betenni -> önálló címaritmetika kell

(pl. CBF: cirkuláris buffer a szűrő együtthatóhoz)

Kelefen számol: nem kell külső plusz memória a számoláskor, egyenesen adja az eredményt, akárhonnan az adatot kapta

Multi channel:

több csatornán, több szűrő fut. Ezeket együtt valja ki a kimenetre. Itt nem gond a pipeline, ugyis öszheb kell várni egymást. A pipelineon keresztül egyszerre több szűrő megy



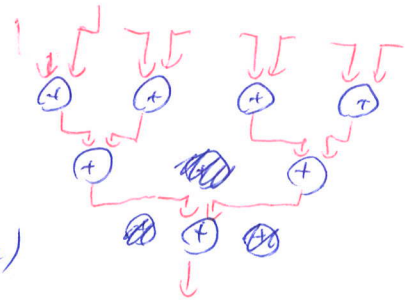
FIR szűrők

. ha valami speci kell, az 60-100-as tap szám

Összeadó fa:

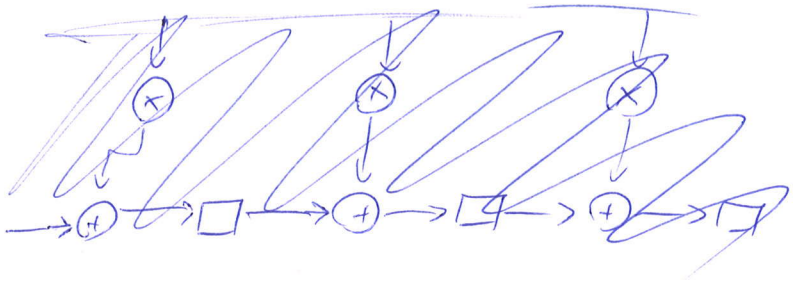
nem sorba vannak kötve, hanem

a kritikus útban így log2(összeadó lesz)



De bármi nehéz kiszámolni -> nem használják

Transzponált FIR: PPT itt is kerülendő, kiába nincs kritikus út.



Ha szimmetrikus vagy antiszimmetrikus a szűrő, akkor az azonos $|h(i)|$ értékkel rendelkező szorzandó értékeket előjelesen előösszegezzük és így feleannyi szorzó kell.

Ahol $|h(i)| = 0 \Rightarrow$ el lehet hagyni a szorzót

Többcsatornás: a kimenőjelsorozat a mintavételezési csatornaszámszorosa

IIR szűrő:
kevesebb tap, de gerjedhet → vigyázni kell a számábrázolással (kerekítés), hogy ne lépjen elő.

MATLAB tud XILINX FPGA szűrőket csinálni

MIT labor
1. GY
2011. 02. 09.

Verilog:

van egy operandusú logikai művelet: egy vektor összes bitjén végzi el

& 4'b 1101 : 0

=== : egyenlőség a don't care (x) és a high impedance (z) figyelembevételével

Öntás és modulképzés csak kettő hatványaival

Összefűzés:

{2{3'b 101}, 2'b 00} = 8'b 101 101 00

Csak <= értékadást használunk (egyszerre kiértékelő, hardverközeli)

Triggerfeltétel lehet mint- és élérzékeny, de egyszerre csak egy féle

Egy blokk legyen teljesen szinkron vagy aszinkron

always @ (*): bármely bemenet változása esetén

if (q): ha q változik

latchet lehetőleg ne használjunk (mert nem lehet miatta időzítést szimulálni)

<= : nem blokkoló értékadás = : blokkoló értékadás (sorrendérzékeny)

nem blokkoló értékadás sor: a reg változók előző értékét használja,

a <= a+1;
b <= a;
} esetén is

for ciklus: többszörös D példányosításkor

case és if else csak always blokk van.
inkább legyen reg változó és always (*) blokk

[latch: élérzékeny D tároló]

LOGSYS.MIT.BME.HU

WEBPACK

SE ÁGY MIT
16. tA
2011. 04, ~~Version~~ 05.

tDC-k:

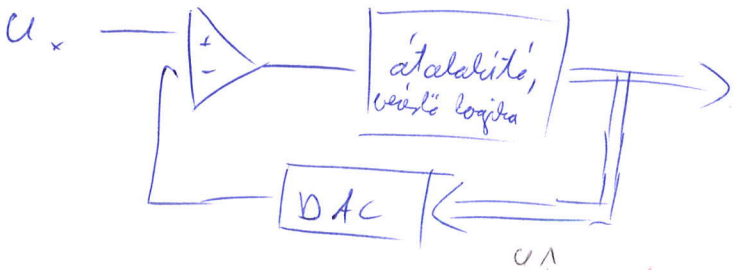
követett: feszt → { idő / frekvencia / stb } → digit kód

követlen: feszt → digit kód

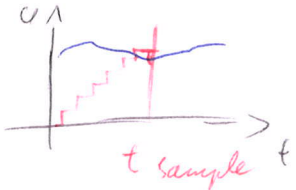
digit kód → jel definiált pillanatértéke
→ jelfüggő pillanatértéke
→ átlag

követlen módszerek:

- visszacsatolt:



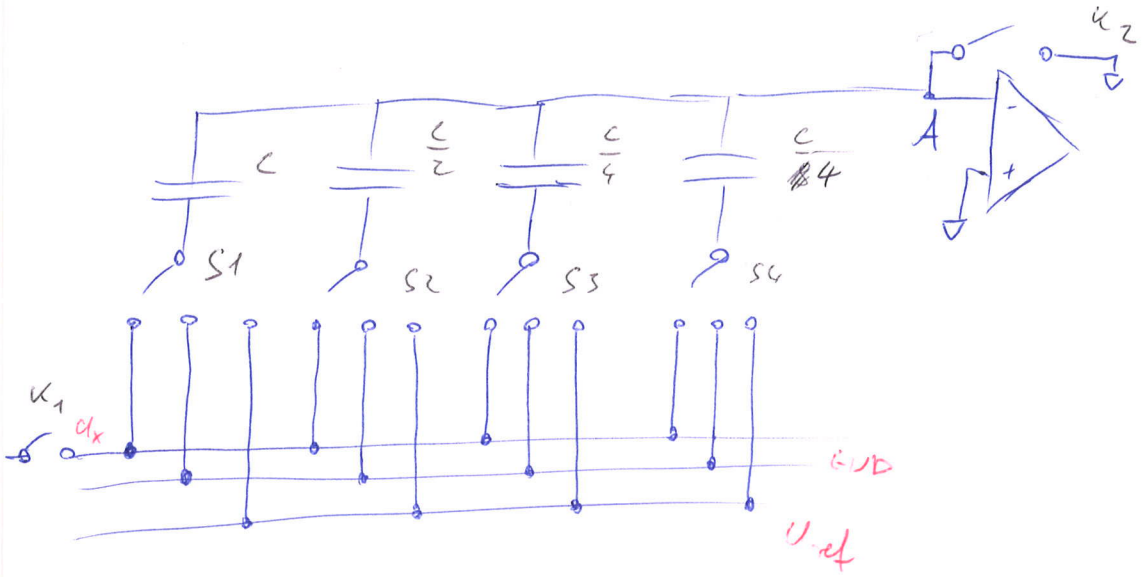
Ha a vezérlő számláló:



- egyg ciklusú, lassabb.
- változó és jelfüggő mintavételi idő

Ha követő számláló: ha a jel elég lassú.
- gyorsabb, de itt is jelfüggő mintavételi idő

- szukcesszív approximáció:
  - gyors
  - fix konverziós idő (bitszámtól függ) → jól definiálható mintavételi idő
  - 12-14 bit, ~100 kHz



Működés:

U_1; U_2 be, S-de U_x-re => Mindent U_x-re tölt

U_1 ki, többi ugyanaz => Hold üzemmód

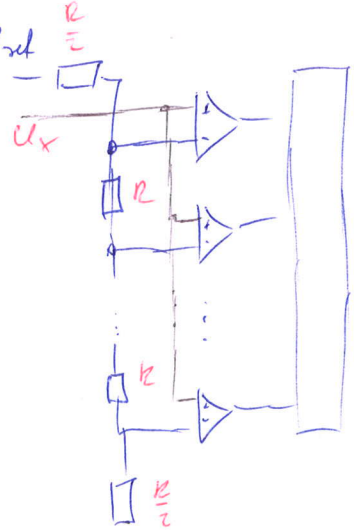
U_2 ki => C lebeg

~~S_1~~ => $S_1 \dots S_N$ felette => $U_1 = -U_x$, mert a kondi tartója a föld

$S_1 \dots S_N$ U_{ref}-re kapcsolása => U_{ref} megoszlik a kapcsolt kondi és a többiek párhuzamos eredője közt, $-U_x$ pedig szuperpozícióval rájön

=> S_1-re $\frac{U_{ref}}{2} - U_x$-et komparálunk

Vonalas konverter



gyors, de sok komparátor kell

$\frac{R}{2}$ -> kisebb hiba, 0,5 LSB-s elcsúszatás

2^n db R, $2^n - 1$ komparátor

↳ emiatt nagy felület

↳ az utólagos trimmelés miatt drága

↳ nagy bemeneti C

↳ de nagyon gyors

~ 100MHz 12 biten

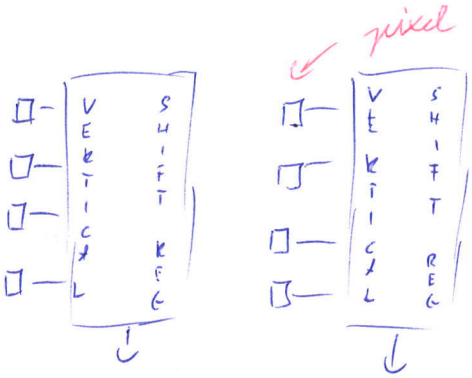
~ 1 GHz 8 biten

-> CCD-ben ez az ~~A/D~~ A/D

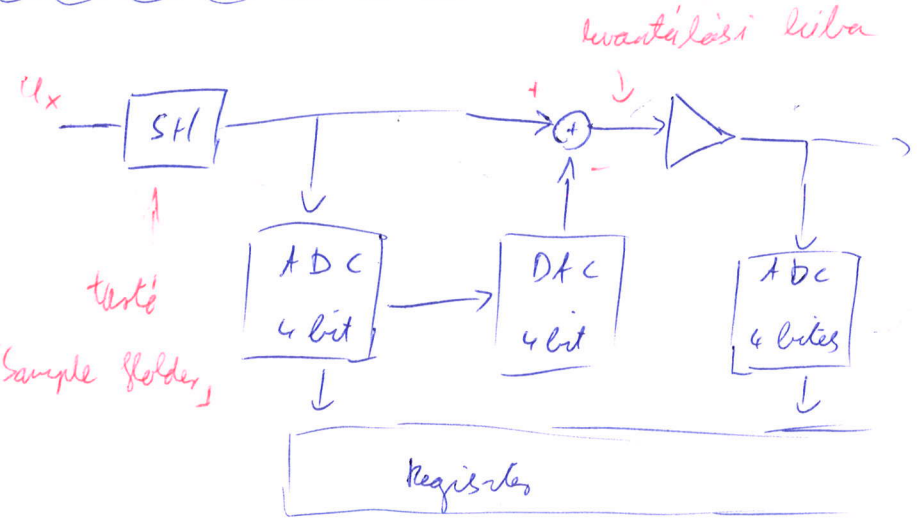
-> digitális kiépítésben is ez van

BÉLA GY MIT
16.EA
2011. 04. 05.

CCD :

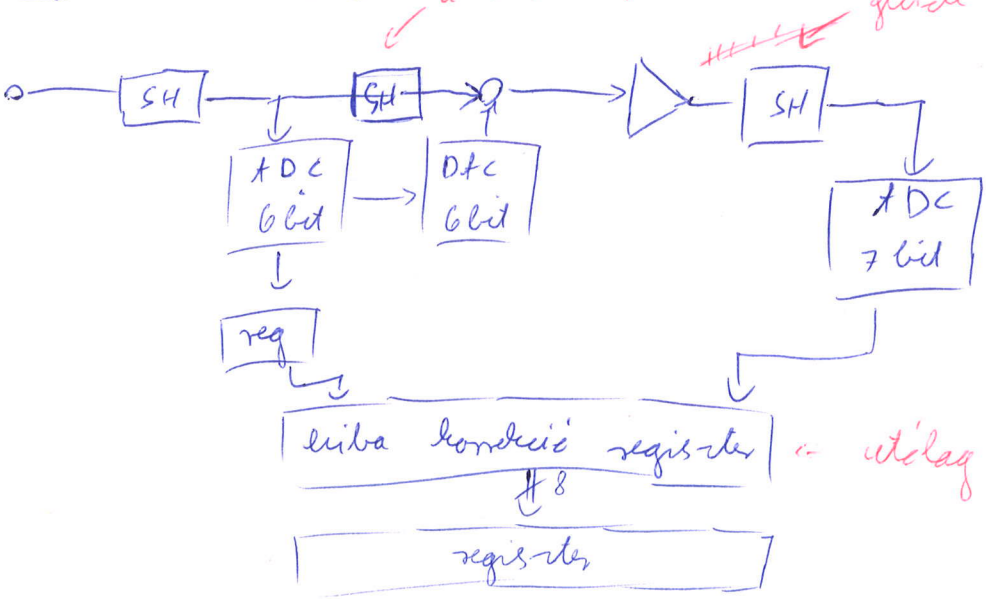


Subranging A/D átalakító



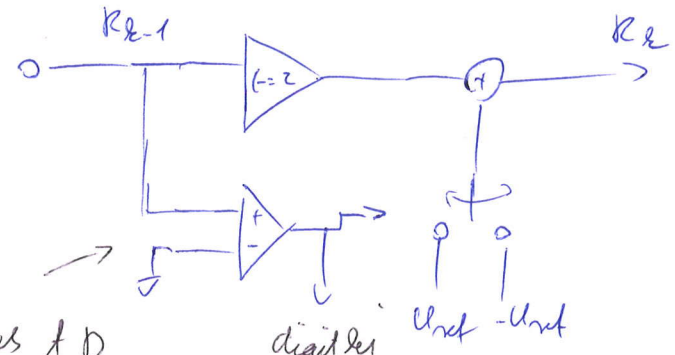
- Több fokozat (akár 4)
- Lassabb jelút
→ nagyobb késleltetés
~ 100MHz
- 8-10 bit

Pipelined subranging



← utólag kompenzál linearitási hibát

Finomfogerősítós fokozat



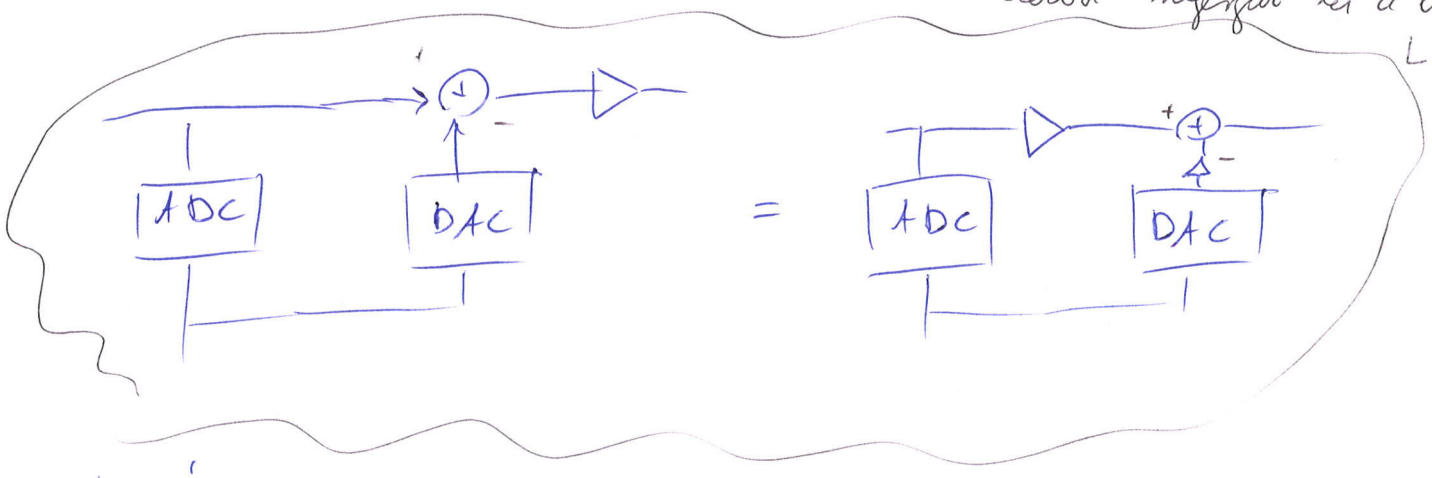
1 bites A/D, lineáris

$U_x \rightarrow 2U_x \pm U_{ref} \rightarrow$

$\rightarrow 4U_x \pm 2U_{ref} \pm U_{ref} \rightarrow$

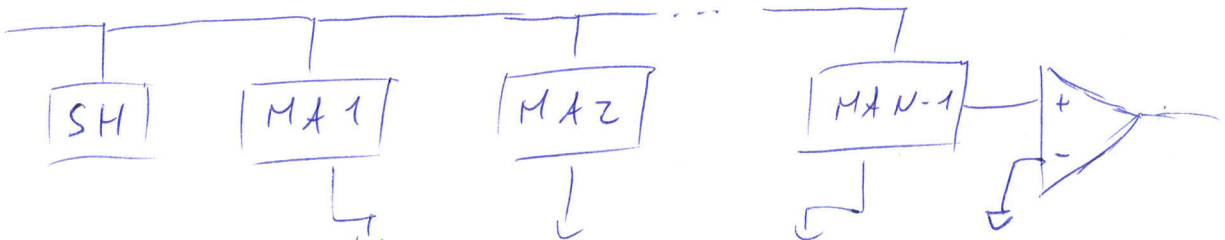
$\rightarrow 8U_x \pm 4U_{ref} \pm 2U_{ref} \pm U_{ref}$

ezekből negyedik ki a bitet LSB lineáris

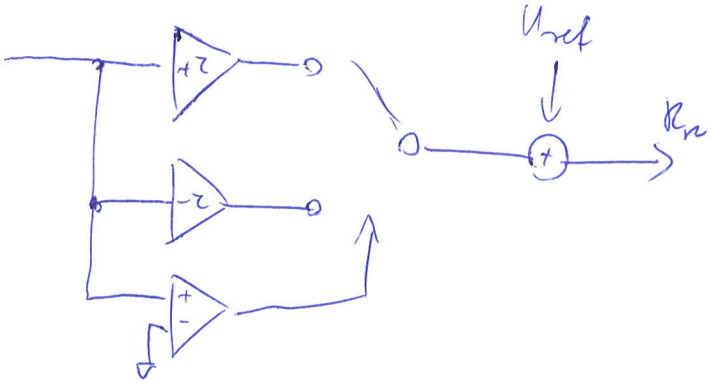


- darabszám

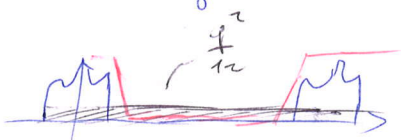
Pl.: 8 bites tmpr



regiszter ⇐ Gray to binary átkódoló



BEÁGY MIT
17.EA
2011.03.07.

Ha nagy f_s-sel mintavételezek, majd részügyük a felső rajzfél részét ()

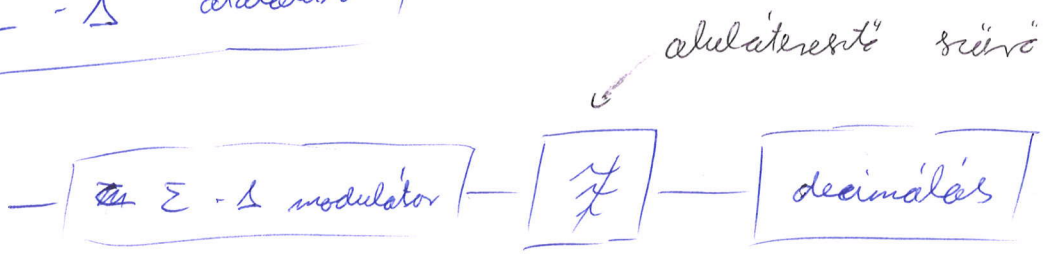
Ezt ledecimálják: $\downarrow K$

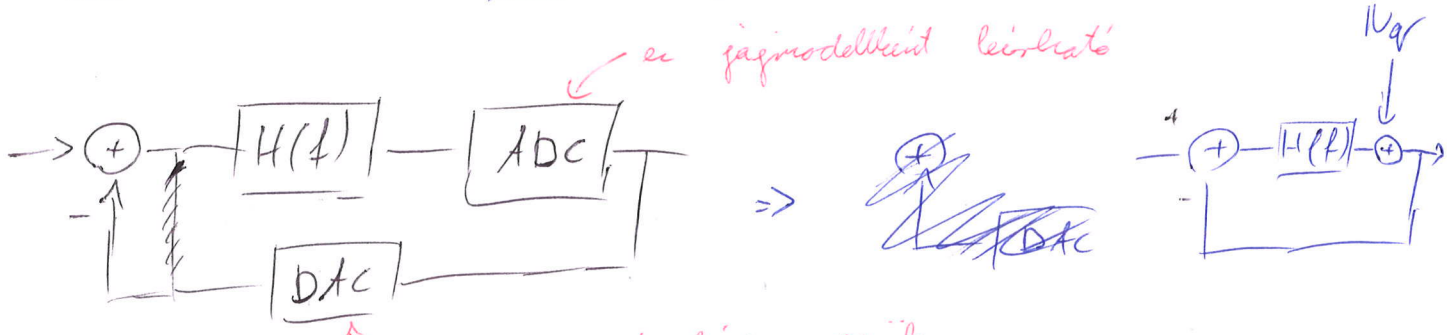
$\Rightarrow$ A maradék kvantálási zajteljesítmény $\frac{q^2}{K \cdot 12}$

$q' = \frac{q}{\sqrt{K}}$ Ha egy bitet akarnak megspórolni

$\Rightarrow \sqrt{K'} = 2 \Rightarrow K = 4$

Σ-Δ átalakító





$$Y(f) = \frac{H(f)}{1+H(f)} X(f) + \frac{1}{1+H(f)} N_q(f)$$

Pl $H(f) = H(j\omega) \cdot \frac{1}{j\omega}$

$$P(j\omega) = \frac{\frac{1}{j\omega}}{1+\frac{1}{j\omega}} X(j\omega) + \frac{1}{1+\frac{1}{j\omega}} N_q(j\omega)$$

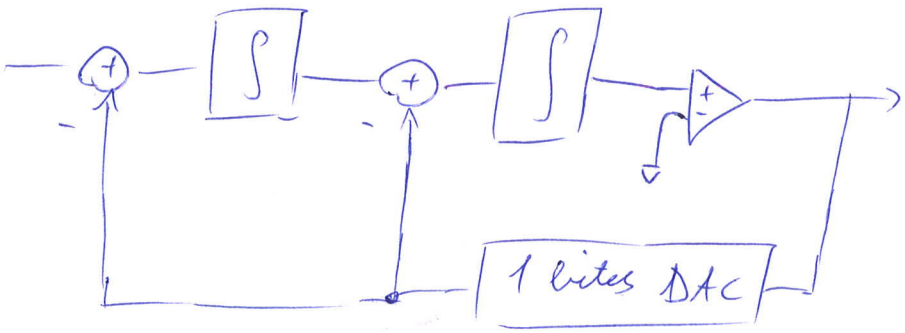
(kisfrekvenciás jelre aluláteresztő; zajra felüláteresztő)

⇒ zajformáló szűrő, az alacsony frekvenciás zajt elnyomja, a hasznos jel bentmarad.

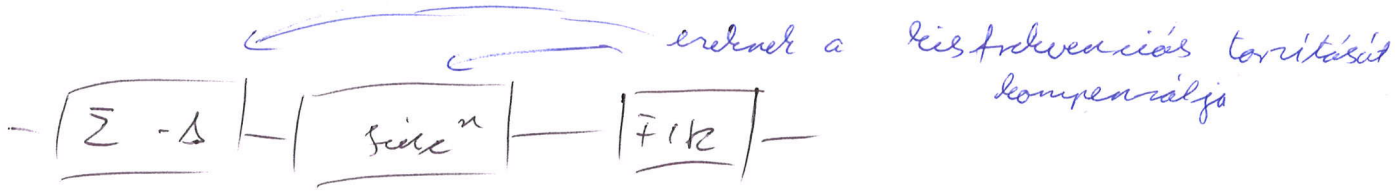
Ezután aluláteresztő szűrés és decimálás

Minél nagyobb fokszámmal szűrünk, ez a hatás annál kontrollabb ($H(f)$-ről beszélve)

Másodfokú modulátor.



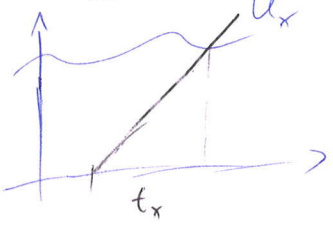
Másodrendű Σ-Δ átalakító felett a stabilitásra figyelni kell

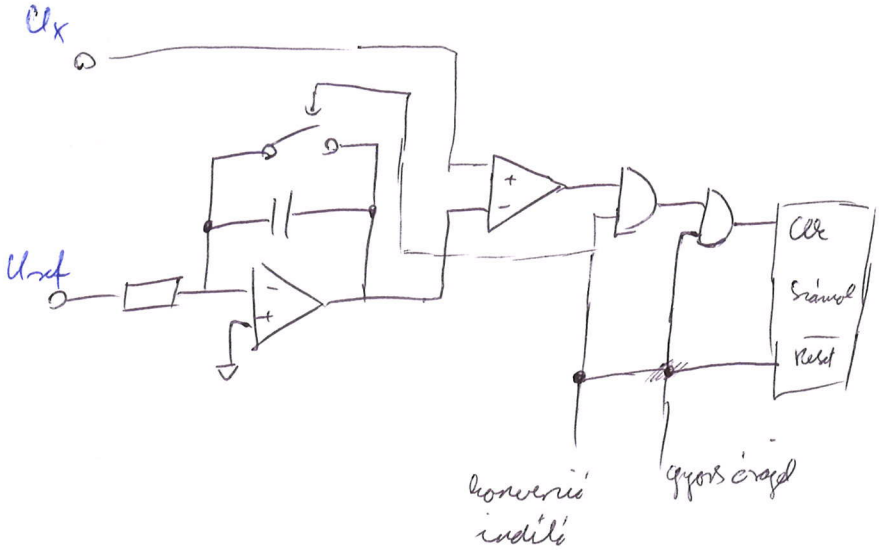


BME AGY HIT
17.EA
2011.04.07.

Közvetett módszerek

Törésrajzeráторos



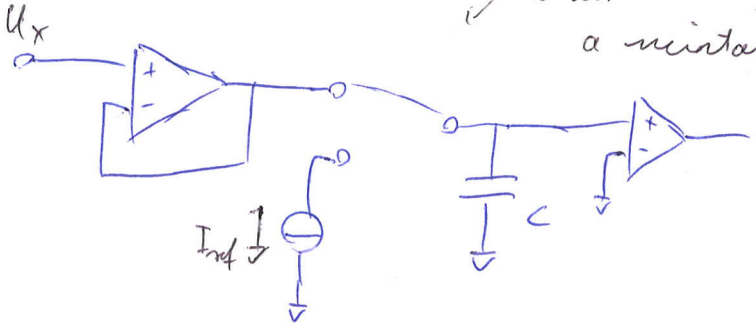


Elindítjuk, integrál, közben a gyors órajel pörgeti a számlálót, amíg az integrál el nem éri U_x-et.

=> t_{konv} változó

=> t mintavételi időpont jelfüggő

Kondenzátor kisütős módszer



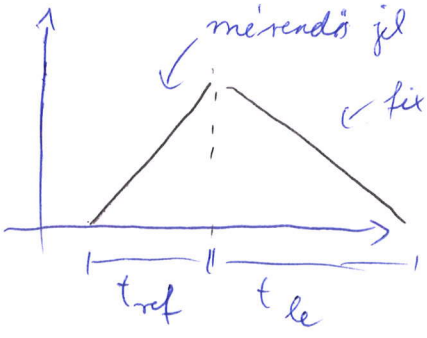
kondit töltünk, majd leintegrálgyuk => legalább a mintavételi időpont fix, de az idő változó

... ugyanaz az elvész, mint fent

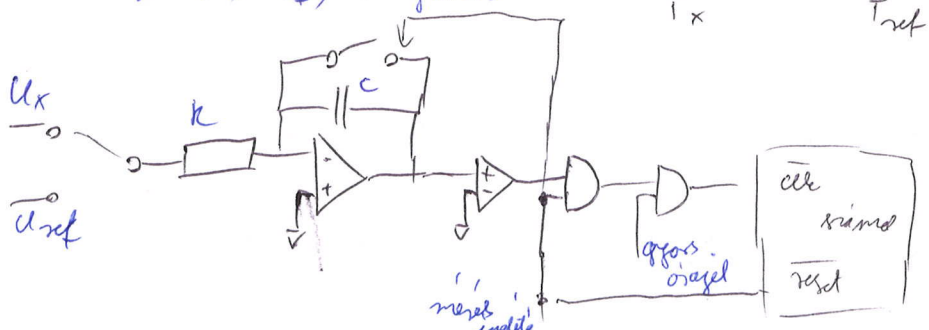
Dual slope

$$\frac{U_x}{k} T_{ref} + \frac{U_{ref}}{k} T_x = \alpha$$

$$\frac{U_x}{T_x} = -\frac{U_{ref}}{T_{ref}}$$



fix meredekségű ($\int U_{ref}$) leintegrálás



$T_x = n_x \cdot T_{óra}$ $\quad T_{ref} = n_{ref} \cdot T_{óra}$

- t mérési ideje $T_{óra}$ legyen stabil → rövid idejű stabilitás
- t_n integrálási idő növelésével nő a pontosság

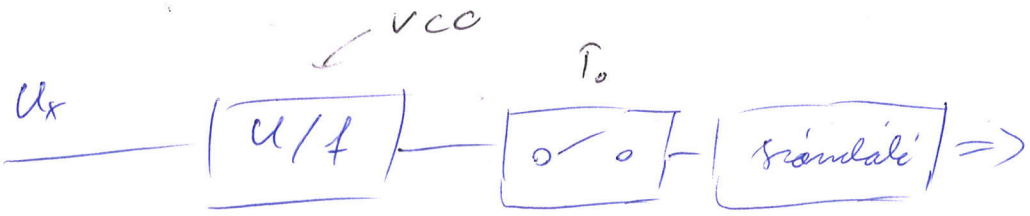
~ Digitális multiméterekben ez van

- f zavarnak az integrálja rá bele a mérésbe. Ez $\pm C$ jelre worst case fél periódusnyi.

Szinusz jelre $0{,}64 \cdot A \quad \left(= \frac{2}{\pi} \cdot A\right)$

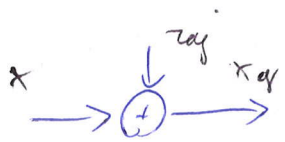
Megoldás, ha a zavarfrekvenciákra igazítjuk a mérési időt (pl. PLL-el szinkronizálni)

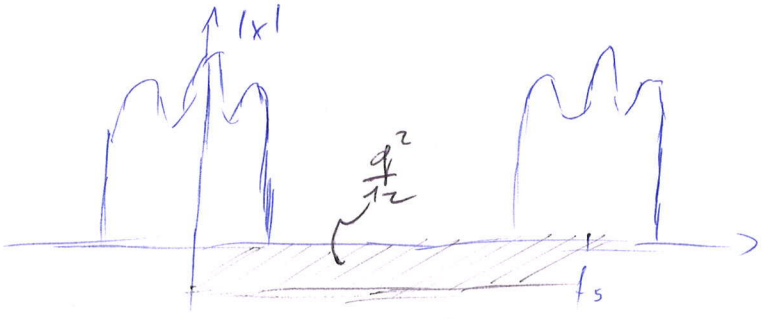
Digitális integrátor



$f = k \cdot U_x$ $\quad N = T_0 \cdot f = \underbrace{T_0 \cdot k}_{\text{konst}} \cdot U_x$

kvantálás



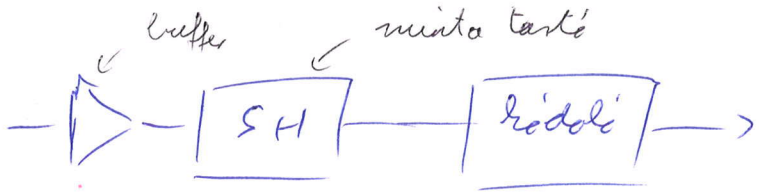


$\frac{q^2}{12}$ a fajlagos teljesítmény, ez osztódik el a spektrumon egyenletesen (mert fehér)

=> ha $f_s \uparrow$, a zajszint csökken.

BEÁLY MIT
18.51
2011. okt. 16.

AD átalakítók hibái:



zaj torzítás véges sáv

zaj torzítás véges sáv

apertúra jitter

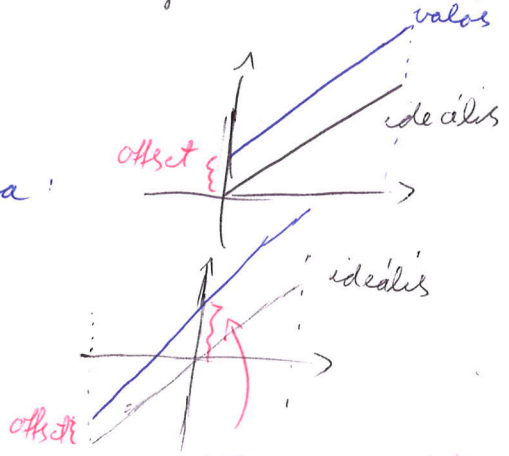
kvantálási zaj torzítás → integrális → differenciális

↳ a mintavételezés ideje nem abban van, amikor akarjuk

DC hibák

offset: unipolárisra:

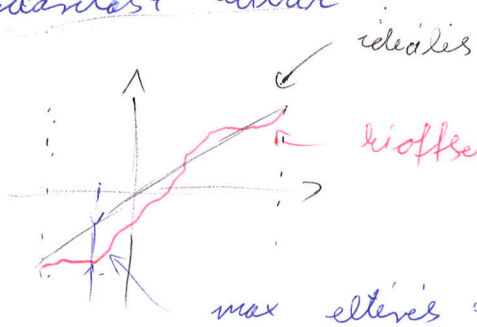
bipolárisra:

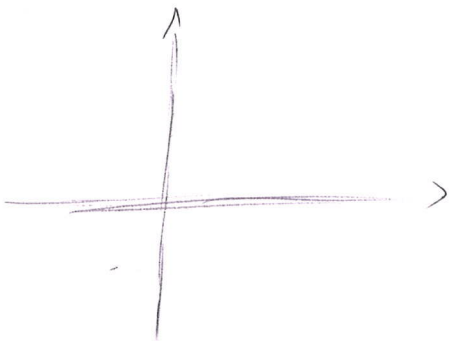


Itt offset mindig a -U fullscale-nél van.

erősítési hiba: kioffsetelejük ($-U_{fullscale}$-re), majd jellemezhetjük a gain hibát.

linearitási hibák:



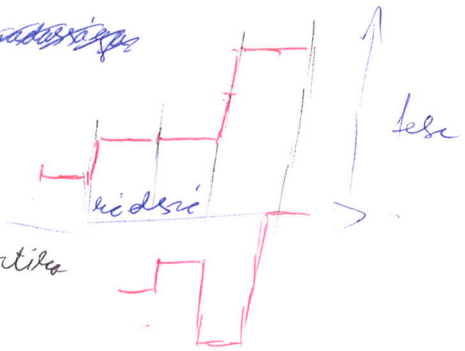


Megadhatják az INL-t a végpontokra vonatkoztatott és a regressziós lineáris modellen képezt is

INL regressziós a legkisebb

DNL: $\frac{q[n+1] - q[n]}{q_{névleges}}$

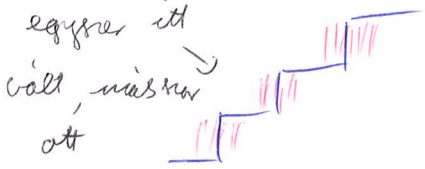
q: kódszint szélessége



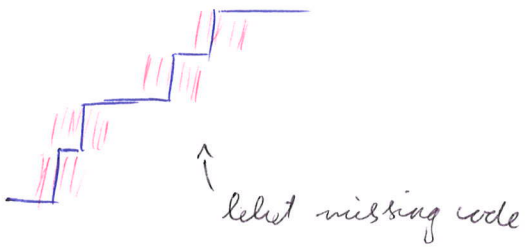
ha van $DNL > 1$, akkor lehet missing code

ha $DNL < -1$ => nem monoton karakterisztika

Kód kódátmeneti zaj



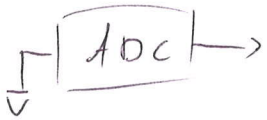
itt nagyobb:

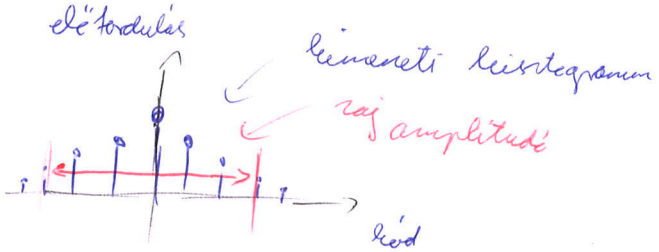


Nagy felbontásnál vagy nagy frekvencián gond

AC hibák

Termikus zaj:





↳ olyan csúcstól csúcsig zaj, mely a kódok 99,9%-át fedi

Gaussra ez $\pm 3{,}3\,\sigma$

Harmonikus torzítás

A: gerjesztő jel eff. értéke

H_i: felharmonikusok eff. értéke

$N_i = H_{z_i}$ zajkomponensek

THD: total harmonic distortion $\quad THD = 20 \log \frac{\sqrt{\sum_{i=1}^{N} H_i^2}}{A}$

N tipikusan kb 5, felette már zajba csúszik

BEÁGY MIT
18. EA
2C 11. CS. 14.

THD+N: THD + noise

$$THDN = 20\log_{10}\left(\frac{\sqrt{\sum_{i=1}^{M^*} N_i^2} + \sqrt{\sum_{i=1}^{N} H_i^2}}{A}\right)$$

kb 4-5

M^*: előre definiált, a jel természetétől függő.
felső $\rightarrow \frac{f_s}{2}$-ig felmegyünk, egyébként elég lecsesélt

SNR:

$$SNR = 20\log\frac{A}{\sqrt{\sum_{i=0}^{M} N_i^2}}$$

M: teljes sávszélre vesszük a jelet

SINAD: signal to noise and ~~distortion~~ distortion

$$SINAD = 20\log\frac{A}{\sqrt{\sum_{i=1}^{M} N_i^2} + \sqrt{\sum_{i=1}^{N} H_i^2}}$$

4-5

C: Full scale fele

B bites kvantálás:

$$P_{jel} = \left(\frac{C}{\sqrt{2}}\right)^2 = \frac{C^2}{2}$$

$$SNR = 10\log\frac{P_{jel}}{P_{zaj}}$$

$$P_{zaj} = \frac{q^2}{12} = \frac{4C^2}{12\cdot 2^{2B}} = \frac{C^2}{3\cdot 2^{2B}}$$

$$SNR = 10\log\frac{\frac{C^2}{2}}{\frac{C^2}{3\cdot 2^{2B}}} = 1{,}76 + 6{,}02\cdot B\ [dB]$$

$$SNR = 10\log\frac{\frac{C^2}{3\cdot 2^{2B}}}{\frac{q^2}{12}}$$

=> akkor van B bitnek értelme, ha tudunk olyan kisebb jel/zaj viszonyt

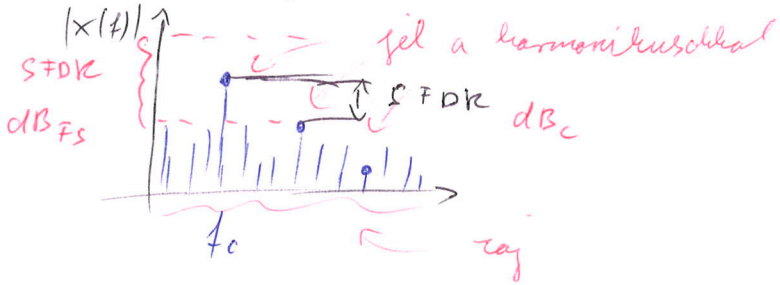
Effektív bitszám (ENOB) $= \frac{SINAD - 1{,}76}{6{,}02}$

Erdat teljesen kiveréselt f_D-n kell mérni

(különben nem látjuk a teljes tartományt és rosszabb lesz a jel/zaj viszony

Spurious free dynamic range

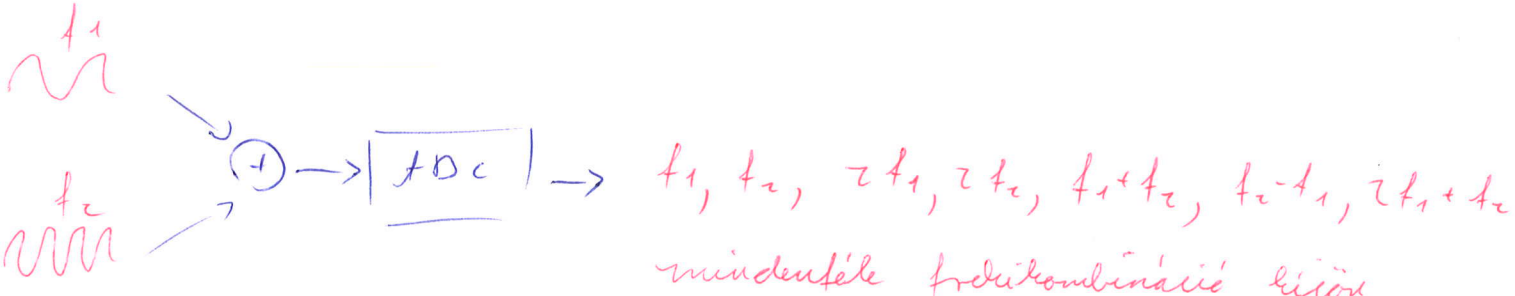
szinuszos gerjesztés:



SFDR: az a dinamikai tartomány, ahonnan a hasznos jel még kiválasztható

[dB c]: decibel relative to carrier

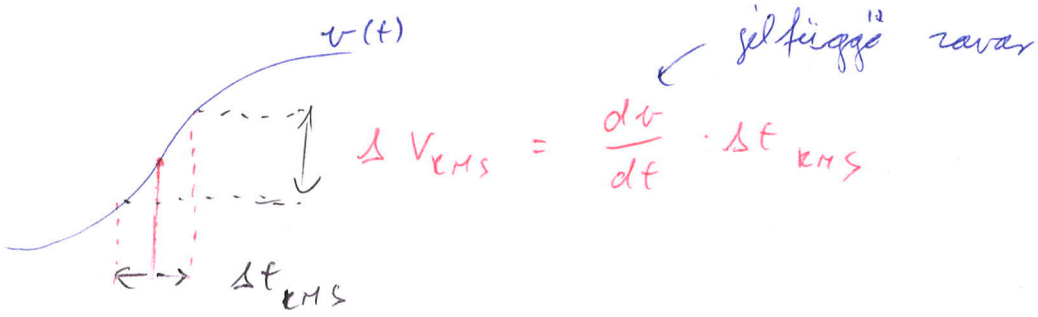
[dB FS]: decibel relative to Full scale



mindenféle frekvenciakombinációk lépnek fel

→ intermodulációs torzítás

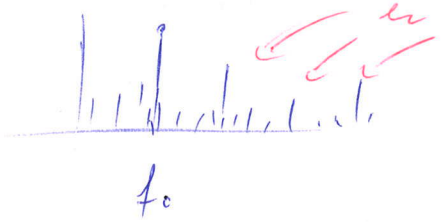
Apertúra jitter:



BE VIK MIT
18.E1
2011. ősz. 14.

A jellemzők mérése:

Ha minden torzításmentes és pontos és szép, ez kapjuk:



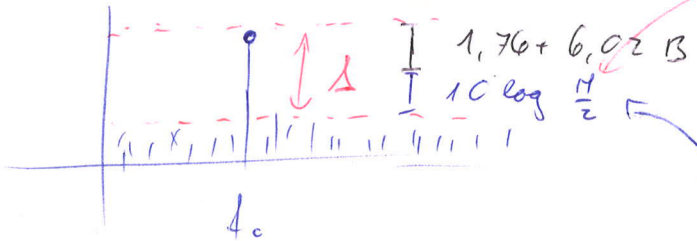
ez mi?

→ mivel rajtalan a mérés (mert minden szép), a kvantálási zaj determinisztikus lesz → torzításként megjelenik.

⇒ zajt kell bekeverni, vagy nem koherens mintavételezés kell. Ekkor kifehéredik a kvantálási zaj

de ez igazi csak szimulációnál jön elő, az életben minden zajos

⇒



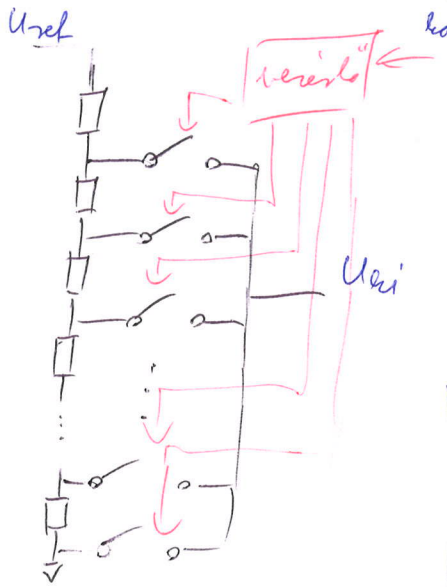
M: mintaszám

ezt az FFT vágja ki a zajból, mert csak abból a frekvenciafelbontásból rántja ki a zajt, ahol jel is volt.

⇒ így lesz meg a maximálható bitszám

D/A

String leire:



~~2^n db kell~~

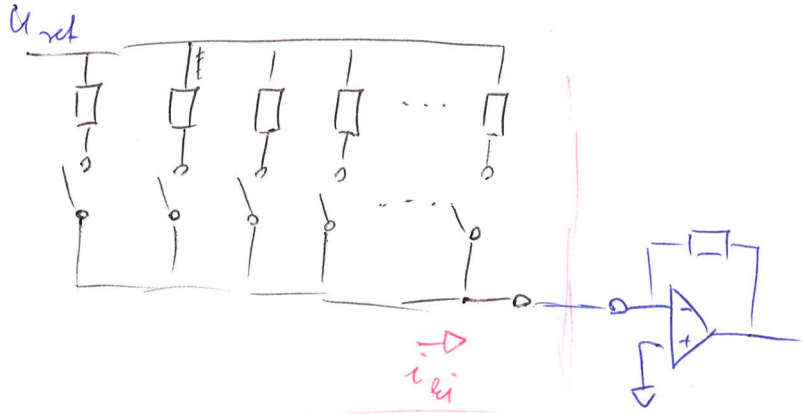
2^n-1 db kell

2^n-1 db kapcsoló

de a glitch kódfüggetlen

Áram kimenetes változat.

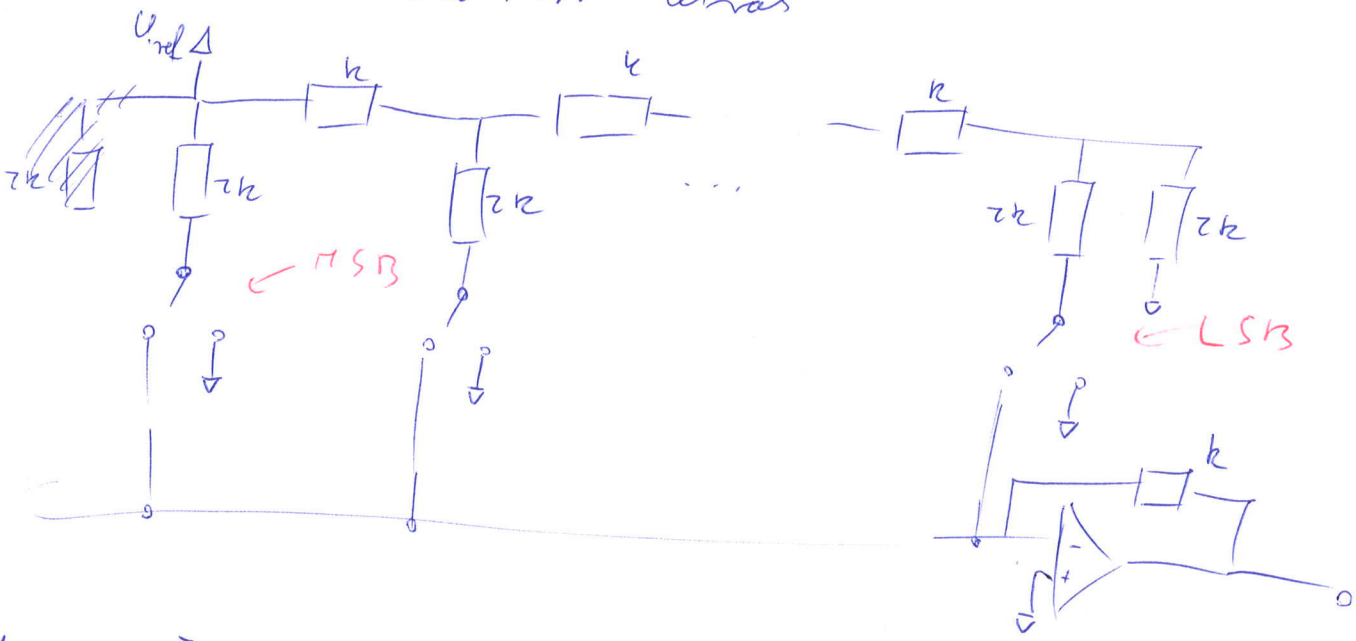
kódfüggetlen glitch, de sok elem



eddig van az IC

B = 4 CV MIT
19. E 1
2011. 04. 14. 19.

R - 2R D/A konverter: létrás



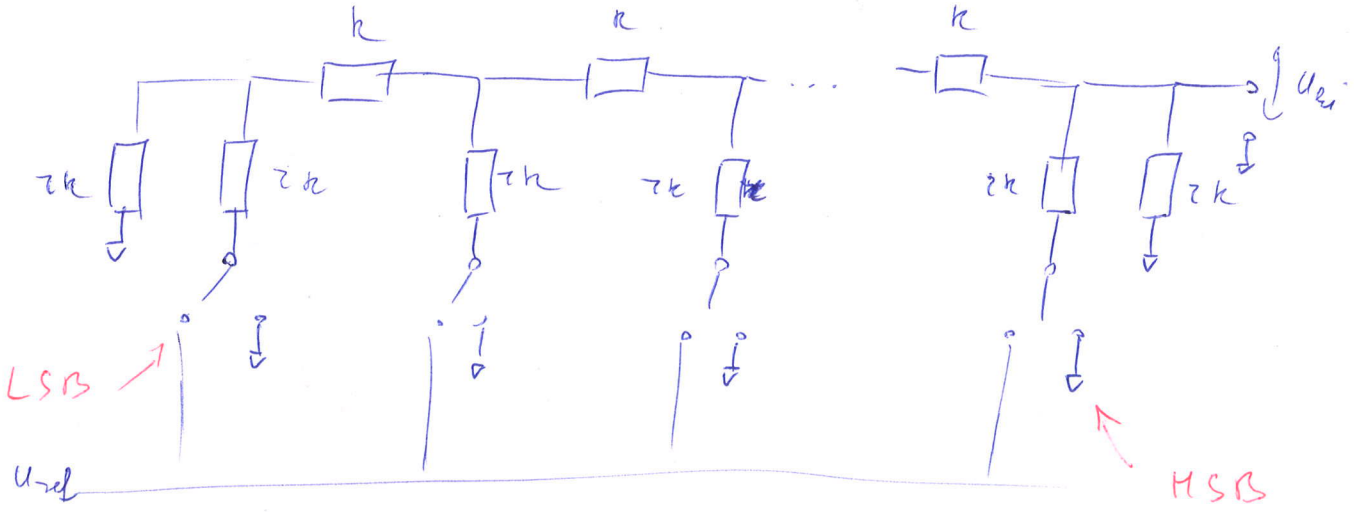
$U_{ref} \rightarrow I_{out} \rightarrow U_{ki}$

Nem lehet U_{ref}-nél pontosabb

$$I_{out} = \sum_{i=0}^{N-1} \frac{U_{ref}}{2R} \cdot a_i \cdot 2^{-i} = \frac{U_{ref}}{R} \cdot R$$

(1)

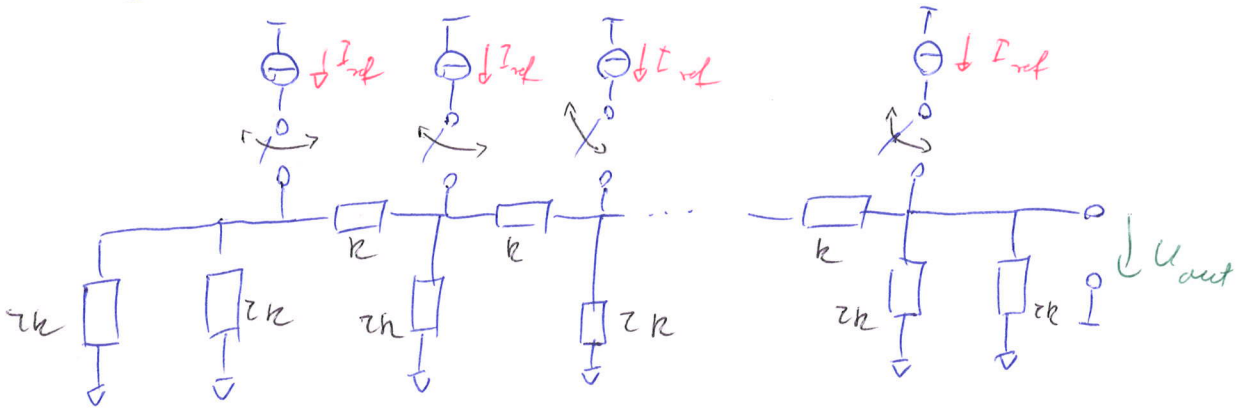
$U_{ref} \rightarrow U_{out}$



$$U_{ki} = \sum_{i=0}^{N-1} \frac{U_{ref}}{3} a_i \cdot 2^{-i} = \frac{2U_{ref}}{3} \cdot R \quad 2N+1 \text{ ellenállás kell}$$

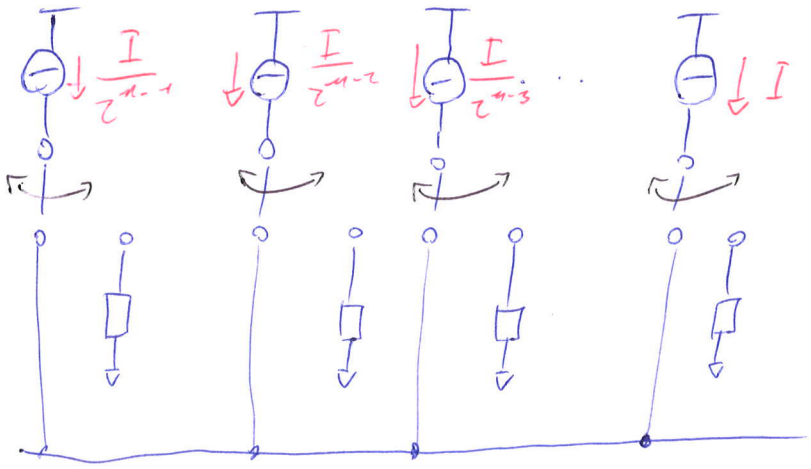
N bitnél

$I_{ref} \rightarrow U_{out}$



$$U_{out} = \sum_{i=0}^{N} I_{ref} \frac{1}{3} \cdot 2R \cdot a_i \cdot 2^{-i} = I_{ref} \cdot \frac{4R}{3} \cdot R$$

$I_{ref} \rightarrow I_{out}$



$$= I \cdot R$$

$$I_{out} = \sum_{i=0}^{N-1} a_i \cdot \frac{I}{2^i}$$

De itt kódlépésből a glitch

Szegmentált DAC

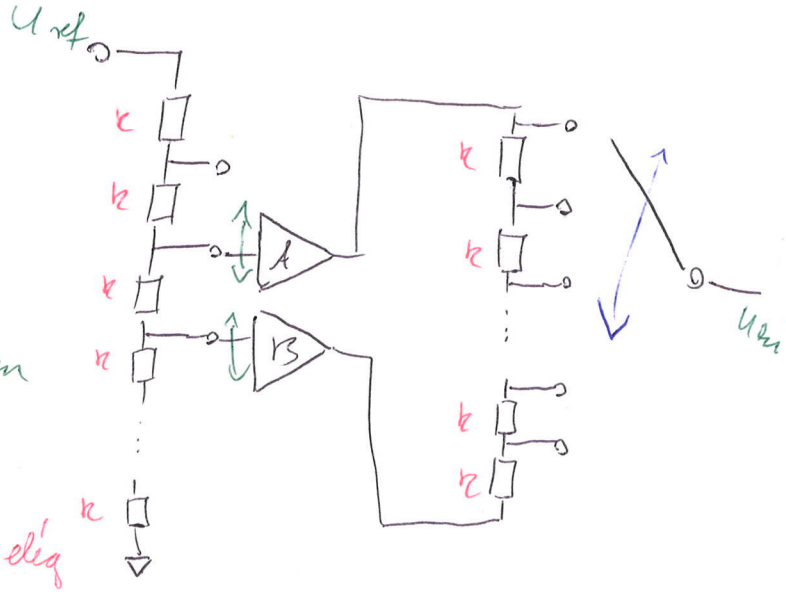
Kalvin - Vorley osztó

A: minden páros ponton van

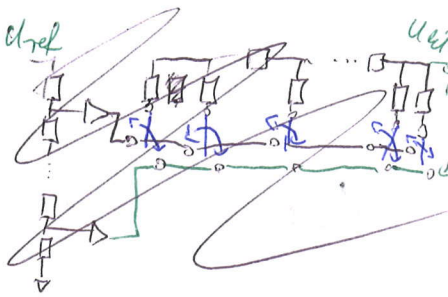
B: minden páratlan ponton van

→ a kivágott fest osztja tovább

N bitre 2^N helyett $2 \cdot 2^{\frac{N}{2}}$ R elég



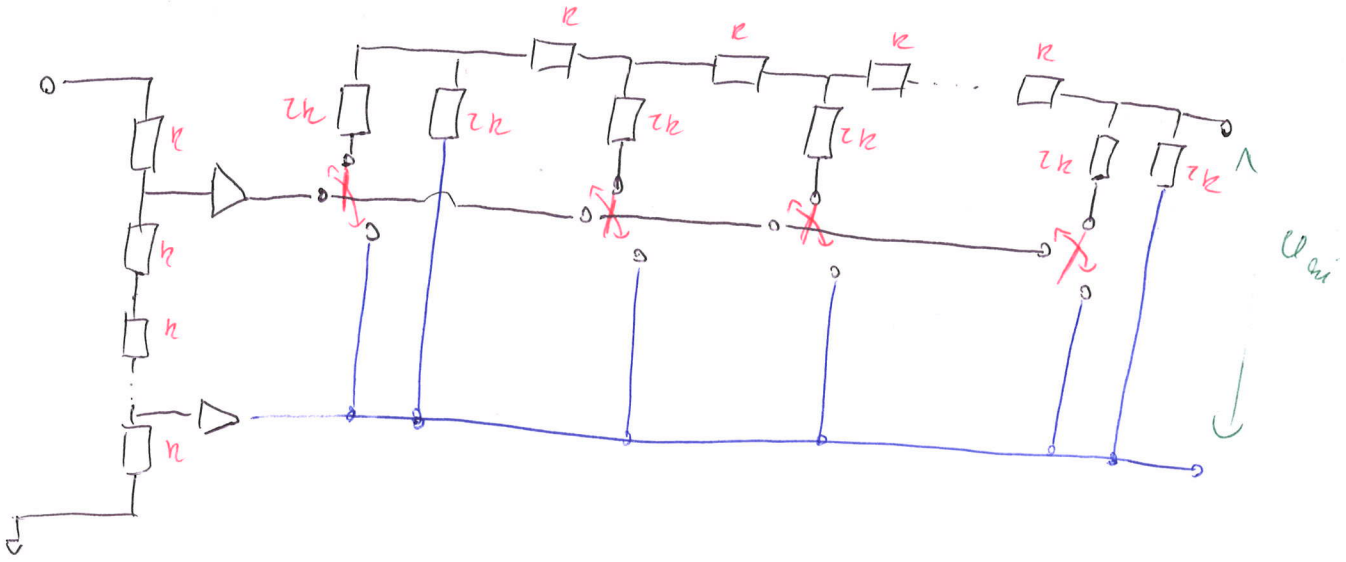
BÉLGY MIT
19.E1
2011 ~~febr.~~ 04. 19.



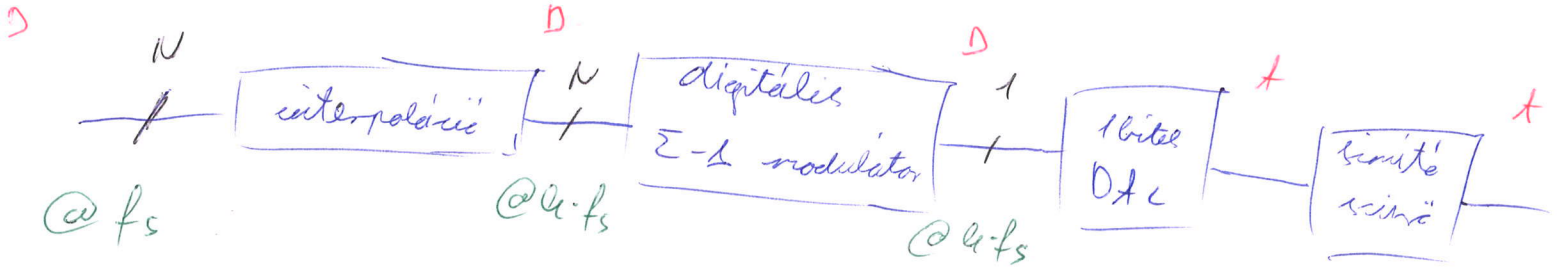
String - R - 2R létra

Egy string bőrét vesszük a létra referencia feszültségét az előző módon

7 + 7 = 14 bitre $2^7 + (2 \cdot 7 + 1)$ ellenállás kell



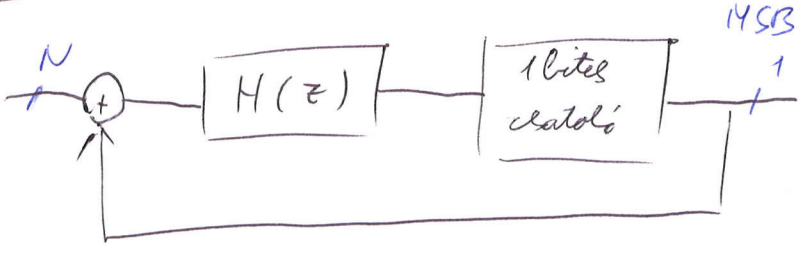
Σ - Δ DAC



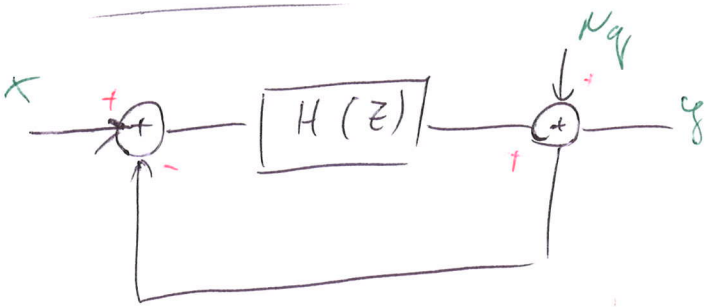
(olyan, mintha PWM lenne)

1 Bites DAC: kapcsolt referencia feszültség

Σ - Δ



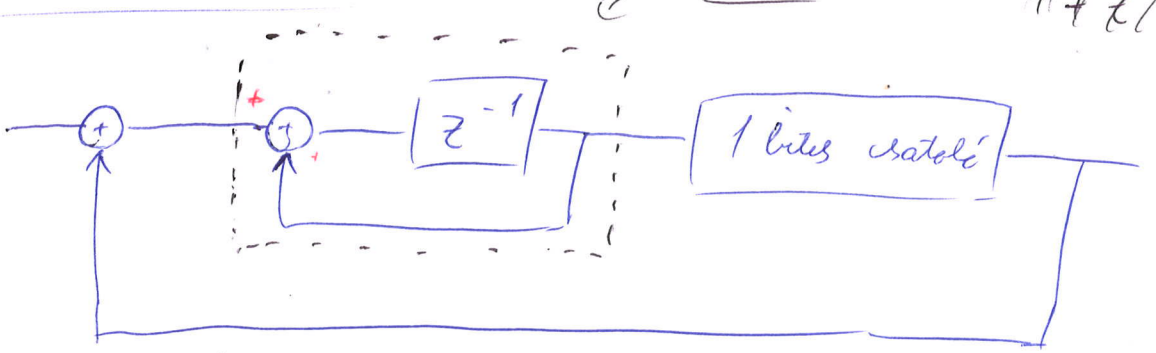
Lineáris modell:



$$Y(z) = \frac{H(z)}{1+H(z)} X(z) + \frac{1}{1+H(z)} V_q(z)$$

Elsőfokú modulátor

$$|H(z)| \rightarrow \frac{z^{-1}}{1+z^{-1}} \qquad \frac{z^{-1}}{1-z^{-1}}$$

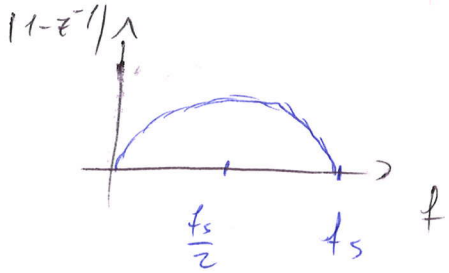


$$Y(z) = \frac{\frac{z^{-1}}{1-z^{-1}}}{1+\frac{z^{-1}}{1-z^{-1}}} X(z) + \frac{1}{1+\frac{z^{-1}}{1-z^{-1}}} V_q(z) = z^{-1} X(z) + \left(1-z^{-1}\right) V_q(z)$$

ez csak fázist tol

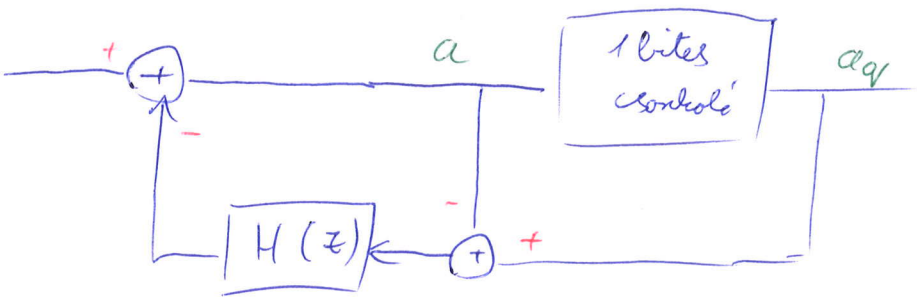
$$\left|1-z^{-1}\right| = \left|1 - e^{-j 2\pi \frac{f}{f_s}}\right| = \left|e^{-j\pi \frac{f}{f_s}} \left(e^{j\pi \frac{f}{f_s}} - e^{-j\pi \frac{f}{f_s}}\right)\right| =$$

$$= \left|2j \sin\left(\pi \frac{f}{f_s}\right)\right| = 2 \sin\left(\pi \frac{f}{f_s}\right)$$



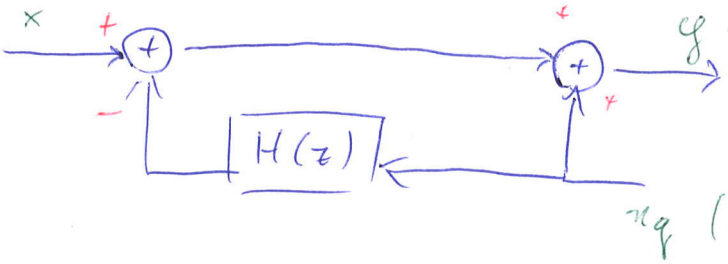
BÉAGY M II
19.I.A
2011. 04. 19.

Hiba visszacsatolásos modulátor



lineáris modell:

Transzformálatlan tartomány jel



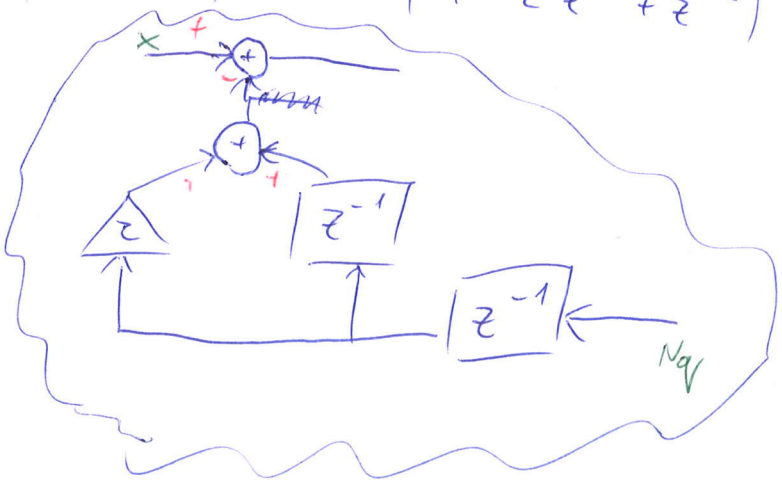
$$Y(z) = X(z) + (1 - H(z))\, n_q(z)$$

a) $H(z) = z^{-1} \Rightarrow X(z) + (1 - z^{-1})\, N_q(z)$

b) $H(z) = \left(1 - (1 - z^{-1})^2\right) \Rightarrow X(z) + (1 - z^{-1})^2 N_q(z)$

$\hookrightarrow |\dots| = 4 \sin^2\left(\pi \frac{f}{f_s}\right)$

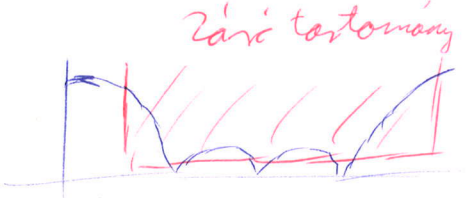
$$1 - (1 - z^{-1})^2 = 1 - (1 - 2z^{-1} + z^{-2}) = z^{-1}(2 - z^{-1})$$



$\Rightarrow$ $H(z)$

BEÁG MIT
~~III.~~ ea. EA
2011. ~~ősz~~ 11.

Interpolációs / decimációs szűrő: az átmeneti tartomány-ban a specifikációja ∅



lineáris interpolációs szűrő: $sinc^2$. De baromira belelóg a záróba

Glitch:

elvárt működés

unipoláris glitch

bipoláris glitch

Mi a jobb? Pár helyen nagy glitch, vagy ~~kicsi~~ mindenhol kicsi?

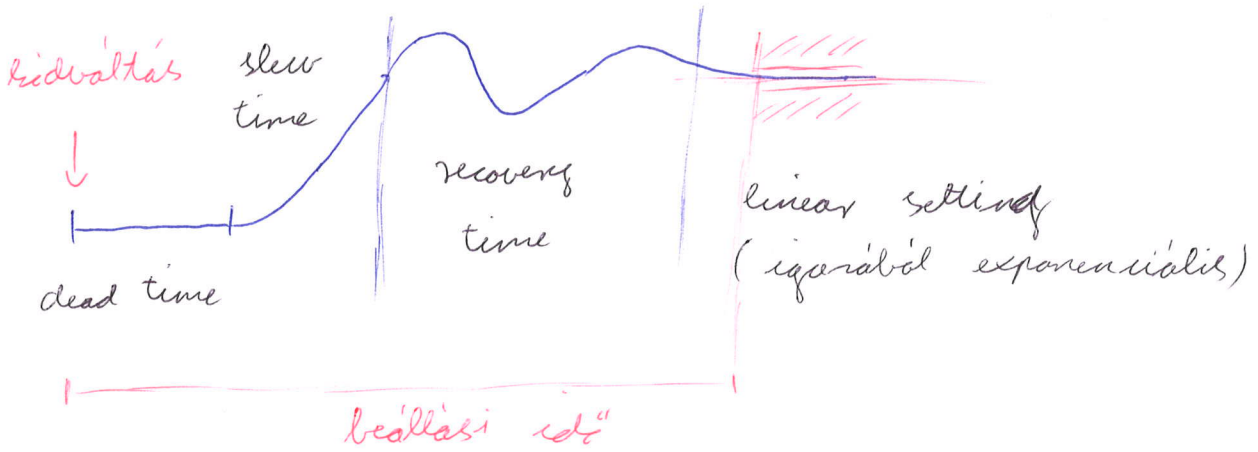
~~Frekvenciatartományban~~ a glitch alapharmonikusa egyezik a periodikus jel alapharmonikusával, ha a glitch a jel bizonyos értékeinél ugrik be.

Ha mindenhol van, akkor f_s-sel lehet periodikus => szűrhető

Jobb a sok kicsi

=> string típusú jelet ilyen szembontból.

D/A kódváltás menete:

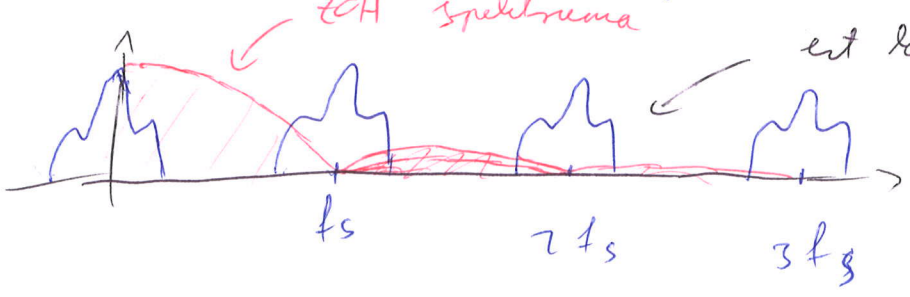


+ mid-scale beállási idő" a leggyorsabb, ezért külön téma

-> addig, hogy elhagyja a ±0,5 LSB sávot, addig, hogy beáll ±0,5 LSB-re. -> Midscale glitch

Glitch terület: [μVs] - ban szokták megadni

A ZOH sinc átviteli függvényének torzítása miatt a jelet a D/A-ra rakás előtt FIR szűrővel előkompenzáljuk



=> Vágásra és fáziskorrekcióra is alkalmazható

BEÁGY MIT
20.ÉA
2011. 04. 21.

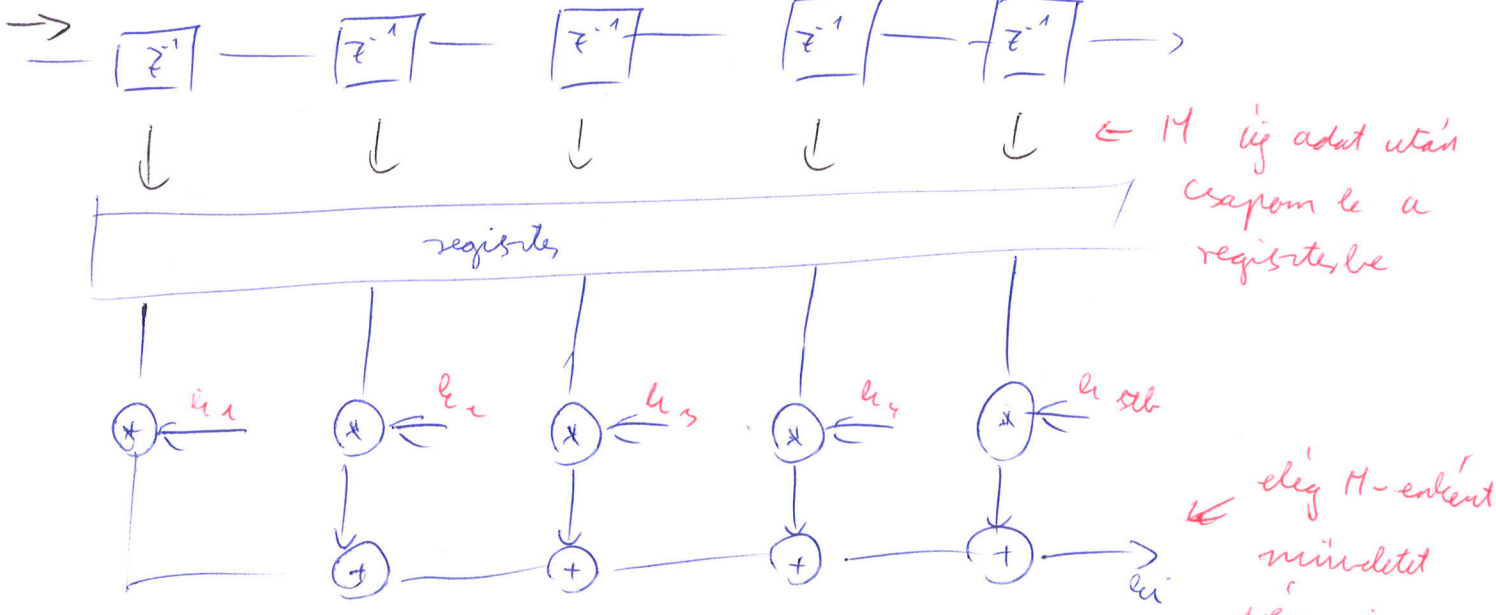
Vélni élesen vágó, jó fáziskarakteri szűrőt csinálni, ami pedig nagyon kéne pl. az audió alkalmazásokhoz.

=> előtte interpolálunk és így ráhajtjuk ki az adatot → sokkal lomhább szűrő is elég

Tudjuk alkalmazás: a hallható spektrum felső részét azért vagdossuk ki, hogy a hangfal ne gerjedjen vele össze. Különben összeadódva a felharmonikusok túl is vezérlik az egészet

1. csoport /10: M-es decimálás számításhatékonysága, 7 IR szűrővel

f_s-sel jön az input

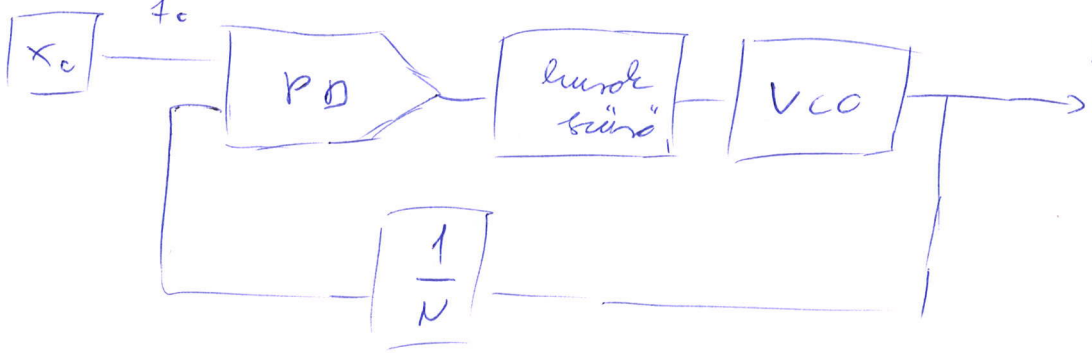


Ne Nem kell így f_s-sel forgatni és summázni

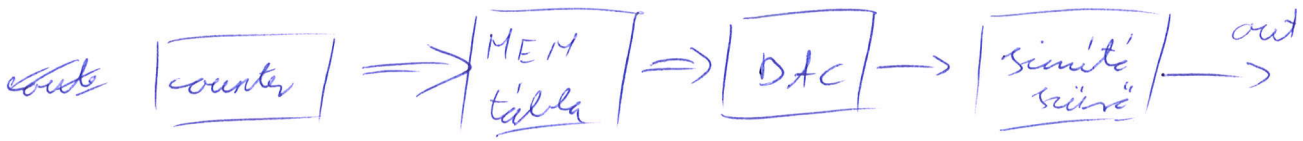
Jelszintézis

Hagyományos: oszcillátorokat szorzogattunk, ebből volt összeg és különbségi frekvenciák. Ebből lehetett szűrni

PLL



Digitális jelszintézis

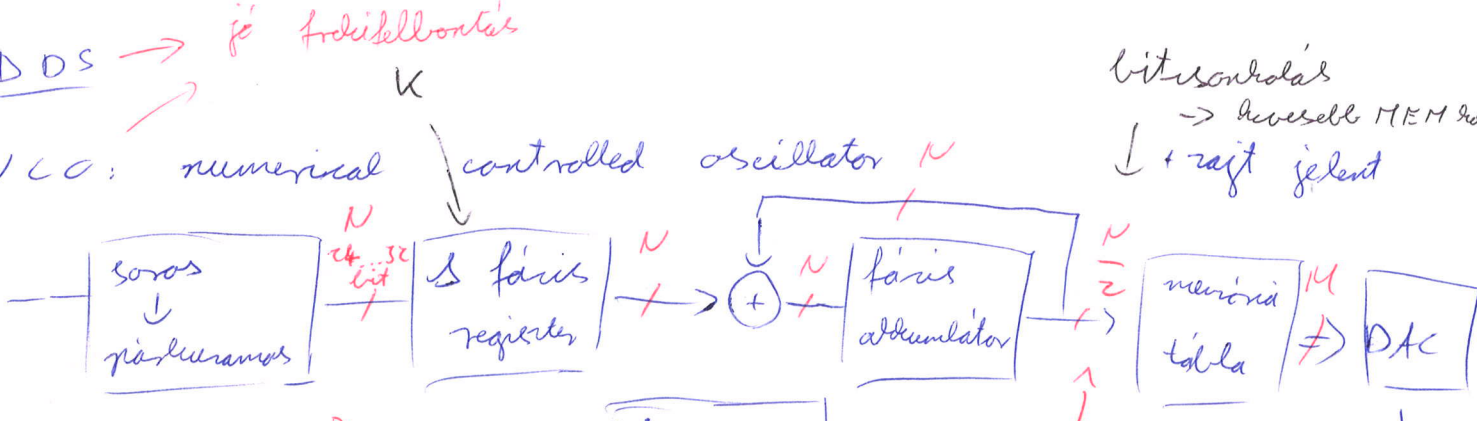


t free nehezen kezelhető, mert az órajelet kéne kezelni.

Hogyan lehet szabadon kezelni őt?

DDS → jó frekvenbontás

NCO: numerical controlled oscillator



bitsorolás → kevesebb MEM kell
↓ + zajt jelent

N: 24...32 bit } fázisfelbontás
N/2: 12...16 bit }
M: 10...12 bit : amplitudó felbontás

$f_{out} = \frac{K \cdot f_c}{\frac{N}{2}}$

De nem akkumulálódik a hiba, mert a fázis akkoriban pontos a számvetés

BÉ ÁGY MIT
11. ÉA
2011. 04. 26.

kommunikáció

- pont - pont vagy busz
- soros vagy párhuzamos
- vezetékes vagy wireless
- szinkron, vagy aszinkron

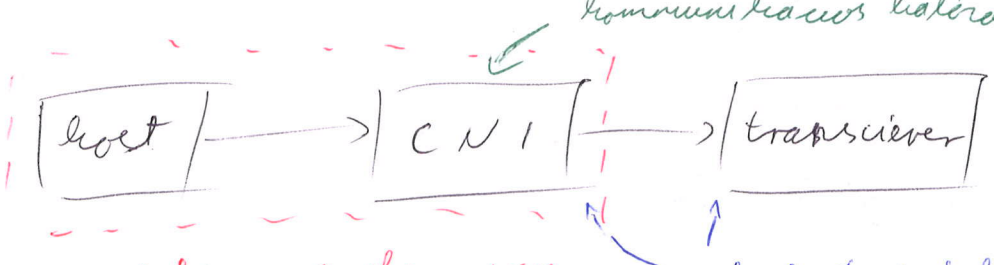
- pont - pont: nincs ütközés → egyszerűbb
de sok pontra drága

- wireless: nincs globális láthatóság, nincs ütközés-detektálás, lehal gathető

- soros: kevés kábel kell, de a párhuzamos gyorsabb
+ sorosban kisebb csatlakozó, kevesebb pin elég

- szinkron: gyors, de órajelet kell küldeni, ami plusz kábel
aszinkron: az órajelelosztás miatt korlátozott a küldési adatsebesség, vagy gyakrabban kell szinkronizálni
→ lehet szinkronizálni kódolással, beszúrt bitekkel vagy headerekkel

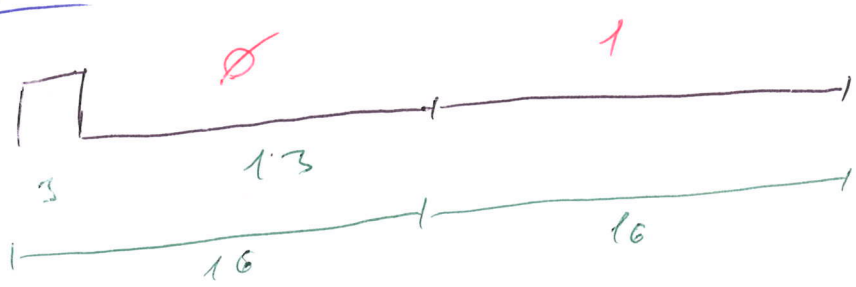
kommunikációs rendszer



↑ transceivernek szüksége kell valami egyedi felismerhetőség, amit elő kell ~~állítani~~ állítani

↑ figyeljünk rá, hogy terhelhető-e a feszültségforrás,

IKDT:



Adat végén CRC a külső fények miatt. Többszörös csomagküldés

Iterációs sebesség: adás előtt addig növelik a bitsebességet, amíg valaki már nem bírja, utána visszaállnak az így megállapított legnagyobb sebességre.

fél duplex

Kommunikáció

Adó: adatot küld Vevő: adatot fogad

Master: forgalmat indít és leállít, órajelet generál

Slave: megcímzett eszköz

Arbitráció: master jelöltek versenye a buszért

↳ minden master jelölt figyeli a buszt és csak szabad buszon ad.

BEÁLY MIT
21.ÉA
2011. 04. 26.

A buson lehet ütközés, ami roncsolásos vagy roncsolás mentes.

- Dominánsabb - recessziv bitek: roncsolás mentes (pl. CAN)

Figyeli, hogy mit adott ki és ha más jön vissza, leszáll a buszról (mert valaki más, magasabb prioritással van rajta)

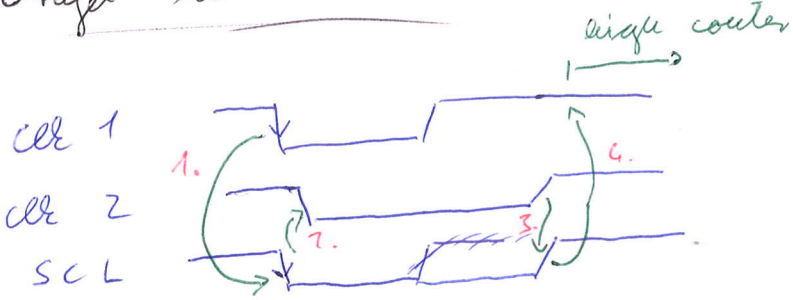
Ezt az ötletet más multimaster rendszerek is használják

I^2C arbitráció

Itt is roncsolás mentes ütközés detektálás van

Az I^2C címet, a CAN csak az üzenet tartalmát küldi ki

Órajel szinkronizáció



1: a leggyorsabb periféria elindítja az óráját, ezzel az SCL-t is

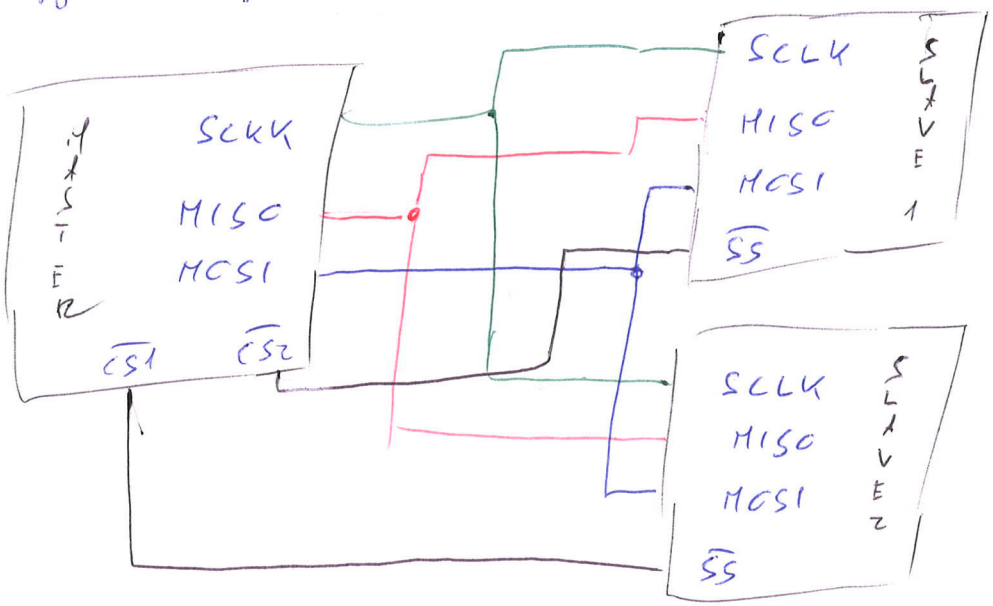
2: ez resetelni a többi órajelet

3: ha minden órajel visszatért magasba, az SCL is visszaugrik

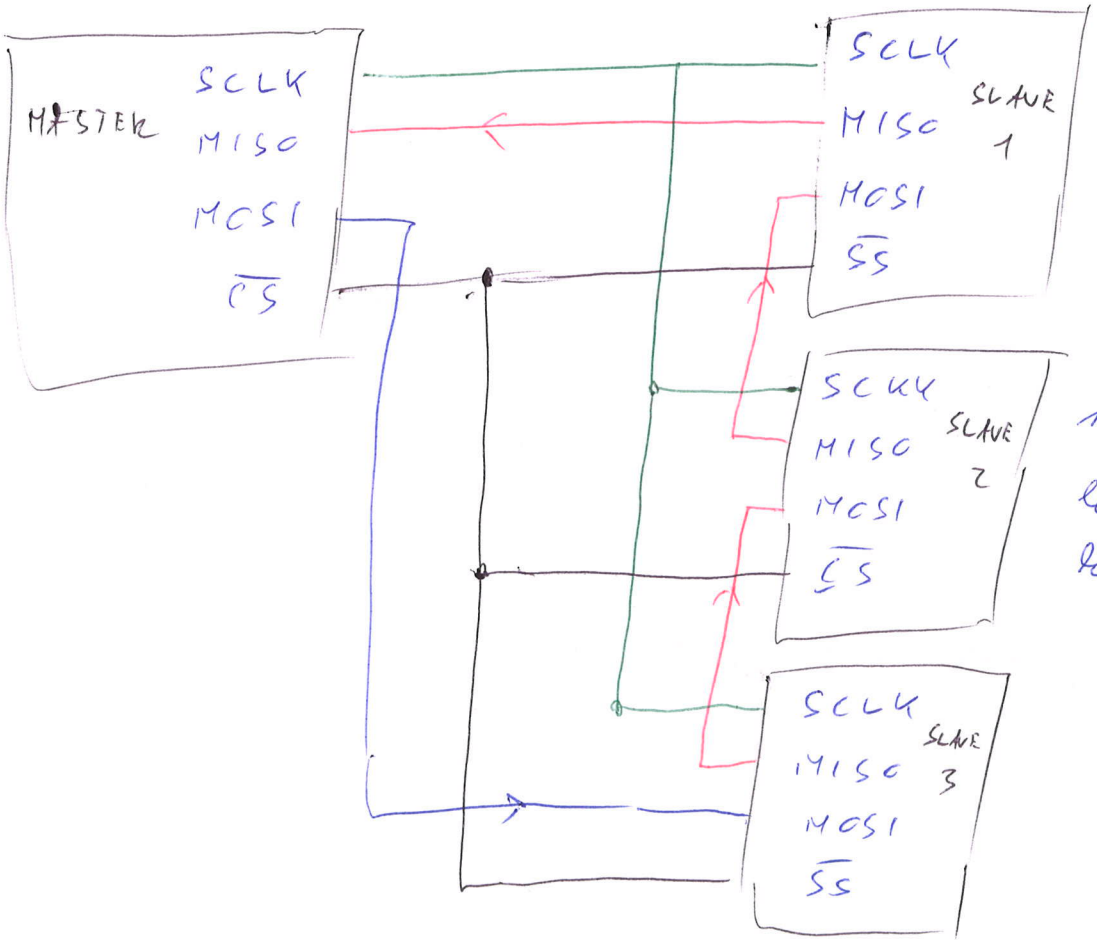
4: az SCL magas után újra indul az egész

SPI busz

Hogyan kötjük össze?



De lánc is lehet fűzni:



1 db mindenki mindent megkap és ~~kiolvasást~~ kiolvasásuk.

Címzéltet és nem szelektív, de lehet gyors és nem kell sok CS

MIT BEÁLY
21.EA
2011. 04. 26.

CAN busz:

Adat integritás, konzisztencia ellenőrzés

- kötött frame formátum → adó/vevő ellenőrzi
- adat vagy mindenkinél jó, vagy az error frame mindenkinél bekúszik
- CRC
- ACK jelet küld a vevő
- megkülönbözteti az átmeneti hibát az állandó hibától
  ↳ aki sok hibás framet ad, az egy idő után kikapcsolja magát (adáshiba számláló)

De ha az egész CAN controller meghülyül, az ellen nem véd

BME AUT MIT
12. EA
2011. 04. 18.

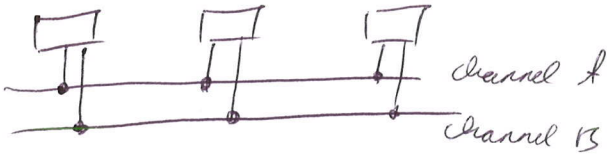
Flexray: biztonsági kritikus alkalmazásokhoz találták ki.

két csatornája van, ezen mehet ugyanaz (→ redundancia)
vagy más (→ nagyobb átviteli sebesség)

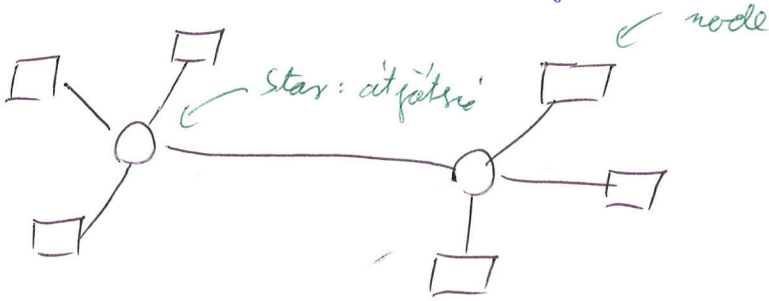
Lehet busra, kapcsolható csillagba → aktív
→ passzív

- aktív: van egy jelátjátszó benne

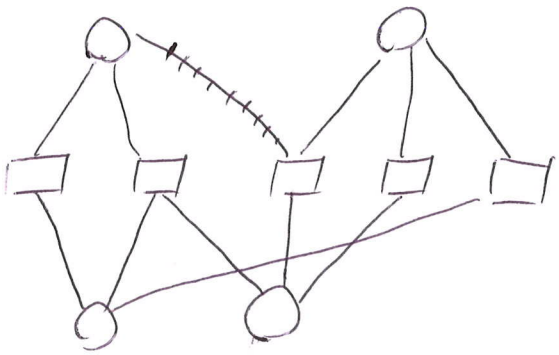
Busz:



1 csatornás hasztád csillag:



2 csatornás hasztád csillag:



Node felépítése:

Van benne TxEN láb: küldés engedélyezése.

Ez fontos, kontrollálni kell, hogy ki mikor ad.

Differenciális jelszintek vannak

itt nincs megfojtás

Van Idle_LP jelszint: alacsony fogyasztású nyugalmi állapot

árnyékolt

Kábel: UTP vagy STP vagy optikai

BME AUT MIT
12. EA
2011. 04. 12.

Kommunikációs szakaszok:

- statikus szegmens ← kötelező rész
- dinamikus szegmens
- szimbólum ablak
- network idle time ← hogy ne csússzanak össze a keretek

↑ szegmenseken belül slotok vannak, azon belül

makrotic ← ezeket tartjuk globálisan szinkronban

- Statikus szegmens: konfigurálhatóan azonos hosszúságúak itt megy ki, hogy hányadik kommunikációs ciklusban vagyunk
- Dinamikus szegmens: változó hosszú, minislotokból áll. ↑ minislotok különböző időpontokban, de makrotic szinkronnal kerödik.

┌ 64 kommunikációs ciklus van egymás után!

- Idle: senki nem ad.
- Szimbólum ablak:
  itt lehet szinkronizálni, ébresztési, arbitrációs helyett itt versenyhelyzetek

Egy frame: ← ebből áll a szegmens

- fejléc, hasznos adat, CRC
  - ↳ ID: hányadik ciklusban vagyunk?
  - ↳ milyen hosszú az adat?
  - ↳ vészjelző bitek

- Ha adó minden byte elején kiad egy Byte - Start - Sequence - t (BSS): 1 elváltás ← ennek segítségével összeszinkronizálja magát a vevő
→ ezt megy a CRC

Egyébként NRZ kódolás

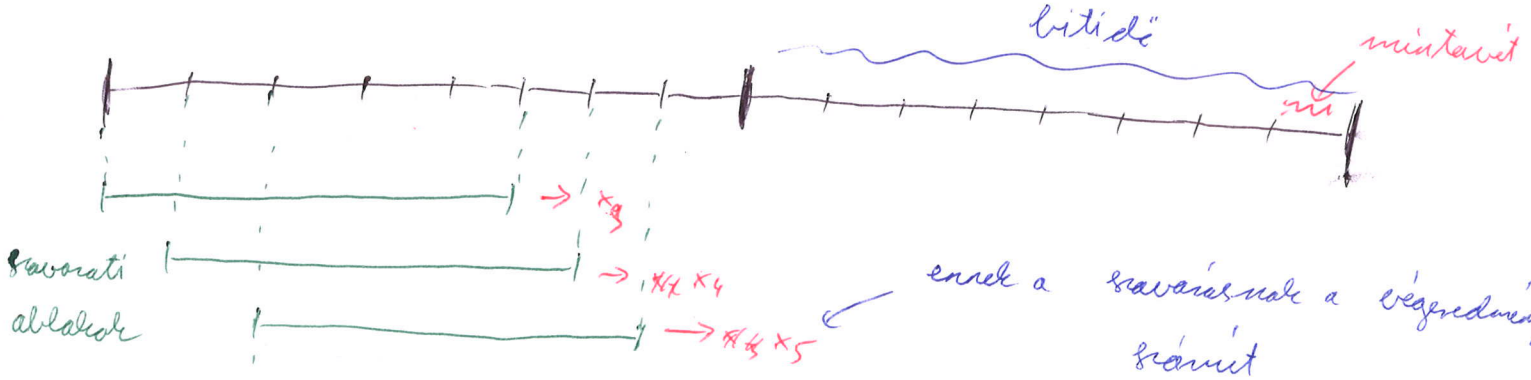
- Dinamikus szinkron: a végén van egy DTS, ami kimutatja a buszfoglalást, ha az adás előbb végetért

Bit szint:

- bitenként 8 mintavételezés.

5 - ös mintavételi ablakkal megnézik végig a jelfolyamon és többségi szavazással döntünk

→ egy biten belül 8 szavazott érték van. 4 bit kezdő értéke a bit közepén a szavazati érték (5. minta)



BME AUT MIT
22.ea
2011.04.28.

Mindegyiknek saját órája van
-> szinkronizáció kell.

- Sűrűbben szinkronizálunk, mivel a lokális idők nagyon szétcsúszhatnak.
- A szinkron célja az együttjárás, a globális időtől lehető el lehet térni
- A makrotickeket kell összefogni

Szinkronizálás páros számú kommunikációs ciklusban, a statikus szegmensben van, külön szinkronizációs ciklusra.

Makrotic offset korrekció van, a node-ok saját mért eltérései alapján. Az eltéréseket mindig jegyezzük -> frekvencia korrekció

Offset korrekció:

- mérés -> időeltérés kijelölése
- ha nincs sync frame -> error
- kibatűrő középérték számítás: a szélső értékeket eldobja.
- ha a korrekció így is limiten kívül esik, akkor is csak limitnyit korrigál -> nincs nagy ugrás
- külső szinkronizációs forrás is lehet

Rate korrekció:

- két egymás utáni kommunikációs ciklus eltéréseit jegyzik -> N adat
- lineáris közelítéssámítás
- csak akkor korrigálunk, ha a frekvenciaeltérés tűréshatáron kívül van: kiba - limit = korrekció
  => nem ~~kell~~ rángat feleslegesen
  -> ennek is van felső határa: nem ránt nagyot.
- ltt és összeszinkronizálhat ~~kell~~ másik clusterrel

Wakeup

- a busrendszernek kell táp
- A busrendszer ébresít. Ébreszthetnek ketten is az A és B csatornán

Startup:

- cold start: a szimbólum ablakban a leading cold start node - ok fogják magukat összeszinkronizálni. Ha sikerült, a follow-cold start ~~ott~~ ebben igazodik

BÉKY MIT
11.EA
2011. 04. 28.

Bus Guardian:

- arra figyel, hogy minden jókor adjon adni.
Ellen figyeli minden csatorna TX-en vételét.
Így nem lehet a buszra bárom módon kiírogatni

↑ Bus Guardiant is figyeli a kommunikációs interface

BG: nem küld, csak felügyel.

Central Bus Guardian

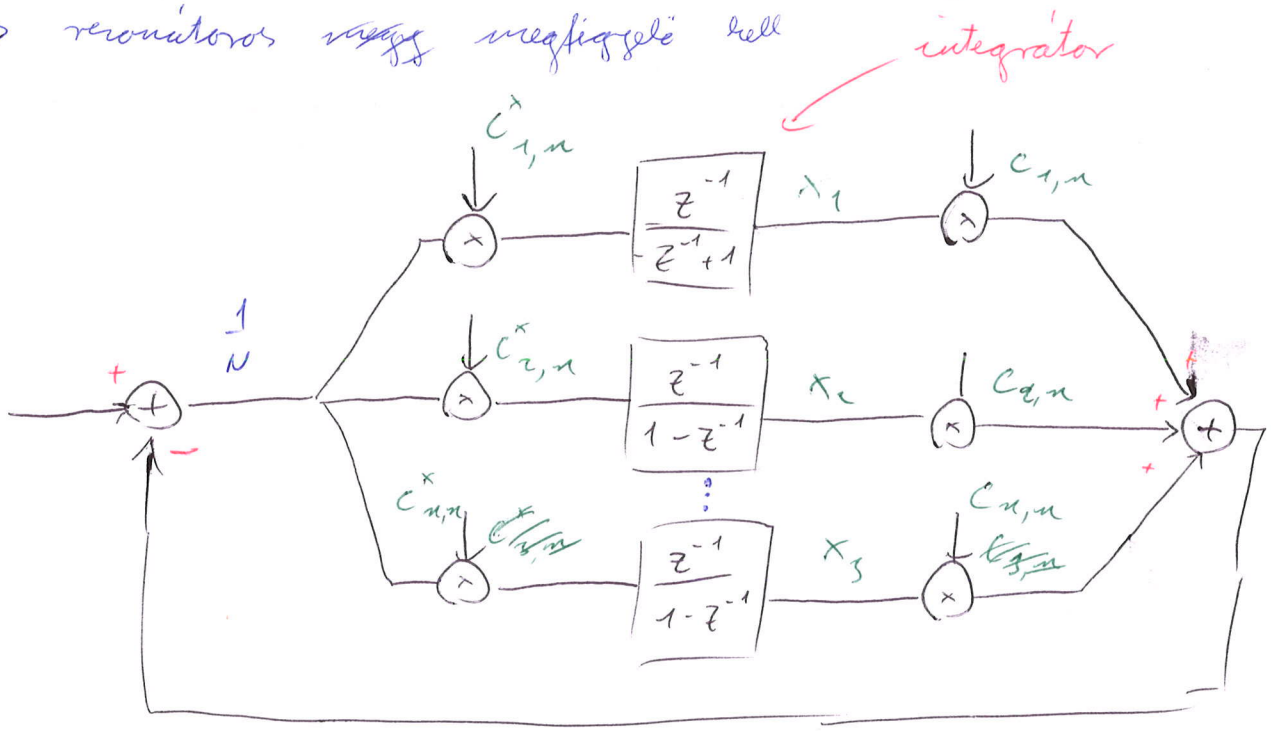
Aktív csillagban az átjátszóba tehetünk. Ez más hibákat is elleőrizni tud.
Értelmezi az üzenetet és ellenőrzi az időzítést
→ intelligensnek kell lenni

SZÁLY MIT
2011.05.05.

Fourier analizátor:

- nem tudjuk egy raj alapharmonikusát csak úgy megállapítani, mert nem biztos, hogy ~~Fourier~~ DFT gridre esik.

=> rezonátoros ~~vagy~~ megfigyelő kell



$$c_{i,n} = e^{j2\pi n \cdot i f}$$

Akkor nyugszik meg a rendszer, ha teljesen elő tudja állítani az eredeti jel spektrumát csatornánként

Itt a ANC-nél hiába mérünk jól, a hangszórónk karakterisztikája még ellenvolltat mindent -> ezt kompenzálni kell

-> fontos az identifikáció

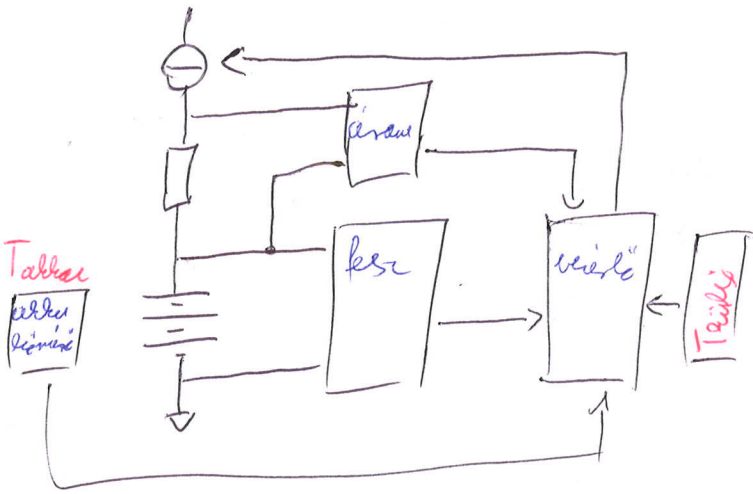
BÉKÉSY MIT
23.E.1
2011. 05. 03.

Töltés

gyorstöltés: < 3 h

lassú töltés: > 10 h

csepp : folyamatos



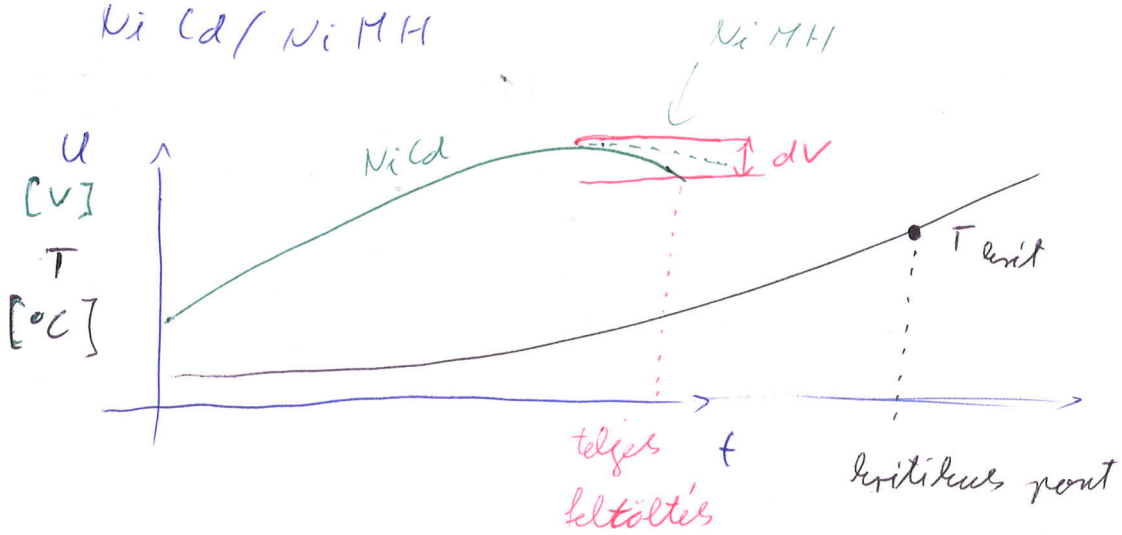
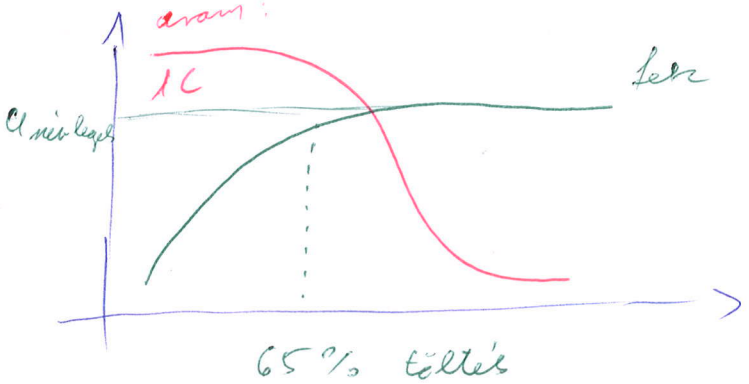
lassú töltés:

	ólom	NiCd	NiMH	Li ion
töltőáram	0,25 C	0,1 C	0,1 C	0,1 C
töltőfeszültség	2,4 V	1,5 V	1,5 V	4,1 vagy 4,2 V
idő	24 h	16 ~~h~~ h	16 h	16 h
terminálás	-	-	timer	fesz

gyors töltés

	ólom	NiCd	NiMH	Li ion
töltőáram	$\geq$ 1,5 C	$\geq$ 1 C	$\geq$ 1 C	1 C
töltőfeszültség	2,45 V	1,5 V	1,5 V	4,1 vagy 4,2 V $\pm$ 50 mV
idő	$\leq$ 1,5 h	$\leq$ 3 h	$\leq$ 3 h	2,5 h
terminálás	T_{min} $\Delta T C O$	$-\Delta V$ $\frac{dT}{dt}$	$\frac{dT}{dt}$; $\frac{dV}{dt} = 0$	
terminálás II	timer $\Delta T C O$	$\Delta T C O$ timer	$\Delta T C O$ timer	$T C O$ timer

Ni Cd / Ni MH



Ni MH