

D/A átalakítók - "algebrai" modell

A D/A (szorzási) algoritmus megvalósításának *domináns* módszere az összegzés : kapcsolt referencia növekmények lineáris szuperpozíciója

(a) Legyen a N numerikus minta (mérészám) **n bites bináris** kódolású (\rightarrow kapcsolás : $b_i = 0, 1$), a választott **unipoláris** tartomány : X_{FS} , így a mértékegység : $\Delta x = X_{FS}/2^n$. Az alábbi, algebrailag *ekvivalens* formák igen *eltérő* topológiákat eredményeznek

$$\begin{aligned}
 x_0 &= N \cdot \Delta x = \left(\sum_{i=1}^n b_i \cdot 2^{-i} \right) \cdot X_{FS} && \text{összegzés (sum)} \\
 &= \sum_{i=1}^n (X_{FS} \cdot b_i) \cdot 2^{-i} = \sum_{i=1}^n \left(\frac{X_{FS}}{2^i} \right) \cdot b_i && \text{súlyozás (weight)} \\
 & && \text{kapcsolás (switch)} \\
 \text{LSB first} & \quad \left[\left[\left[0 + X_{FS} \cdot b_n \right] \cdot \frac{1}{2} + \dots + X_{FS} \cdot b_2 \right] \cdot \frac{1}{2} + X_{FS} \cdot b_1 \right] \cdot \frac{1}{2} && \text{soros (bit-at-a-time)} \\
 \text{MSB first} & \quad \left[\left[0 + \Delta x \cdot b_1 \right] \cdot 2 + \dots + \Delta x \cdot b_2 \right] \cdot 2 + \Delta x \cdot b_n
 \end{aligned}$$

A **párhuzamos (word-at-a-time)** forma referencia "súlyozás": *illesztett* áramkörti komponensek felhasználásával, *egy órajel* ütemben¹ **összegzi** az eredményt, és a kapcsolókat közvetlenül a b_i adat-bit értékek vezérlik (amelyeket belső *digitális* regiszter tárol: NRZ üzemmód).

Az alkalmazáshoz igazodva, és a jel (feszültség, áram, töltés), a komponensek (R, C, félvezető terület), a technológia (bipolar, CMOS) függvényében igen eltérő struktúrák implementálhatók

A **soros (bit-at-a-time)** forma *változatlan* referencia értékeket kapcsolva, bitenként : *n lépésben* **akkumulálja** a kimenetet, a bit-sorrendtől függően analóg *felező* vagy *kétszerező* módszerrel.

A struktúra *azonos* fokozatok **kaszkád** (egymást követő)² kapcsolásán alapulhat; vagy "takarékos" megoldás is lehetséges analóg memóriával : *egyetlen* fokozat **ciklikus** (ismételt)³ működtetése. Megjegyzendő, hogy az átalakítás végén, az újabb adat-frissítésig, szükséges a kimenő érték *analóg* tárolása ("dinamikus, szekvenciális" D/A)

(b) **Speciális kódolási formák** segítenek - *párhuzamos* átalakítónál - a bináris hatvány szerinti változás dinamikájának enyhítésére : lazább komponens illesztés és - különösen MSB váltásnál (major carry) - kedvezőbb tranzienis viselkedés eléréséhez

Garantált monotonitást ad az azonos súlyozású, egység elemekből felépített, $n \rightarrow 2^n - 1$ átkódolást igénylő átalakító⁴. Az elem-szám exponenciális növekedése miatt azonban csak viszonylag kis bitszám esetén alkalmazható ez az **ún. hőmérő (redundáns) kódolású** D/A.

Az elv speciális esete a **szint-kiválasztó** D/A (\rightarrow *azonos* R elemekkel felépített feszültség osztó), amely a megvalósított összes lehetséges kimenő értékből szelektálja⁵ ($n \rightarrow 1$ out of $2^n - 1$ pozíció kiválasztással) az éppen aktuális bemenő kódhoz tartozó jel-szintet

A kétféle kódolás előnyeit egyesíti a két eljárás kombinációja : az MSB biteket ($m \rightarrow 2^m - 1$ átkódolás után) egység elemekből realizáljuk, a többi ($n - m$) LSB biteket pedig *bináris* súlyozással⁶; a technológia és az alkalmazás befolyásolja a **szegmentált** D/A optimális struktúráját

¹ "flash DAC" (one-step); *parallel* decoder (instantaneous sum)

² "ripple through DAC" - a fokozatok közül csak egyetlen aktív. Analóg *memória* (tárolás) közbeiktatásával a fokozatok műveletvégzése nagymértékben párhuzamosítható : "pipeline DAC", így az átviteli képességet (word-rate) csak *egy* fokozat korlátozza - persze a kezdeti terjedési késleltetés (latency) megmarad (és *folyamatos* adatfrissítés szükséges, mert a fokozatok különböző - egymást követő - mintákat kezelnek)

³ "algorithmic (cyclic) DAC" - például SC (switched capacitor) technika felhasználásával

⁴ "thermometer (level) coded, unit element DAC" - a sea of reference sources

⁵ "walking-one coded, level-selection DAC" (\rightarrow decoder type, resistor string DAC)

⁶ "segmented DAC" - a split architecture

(1) ellenállás ("R/2R") hálózat resistor-ladder

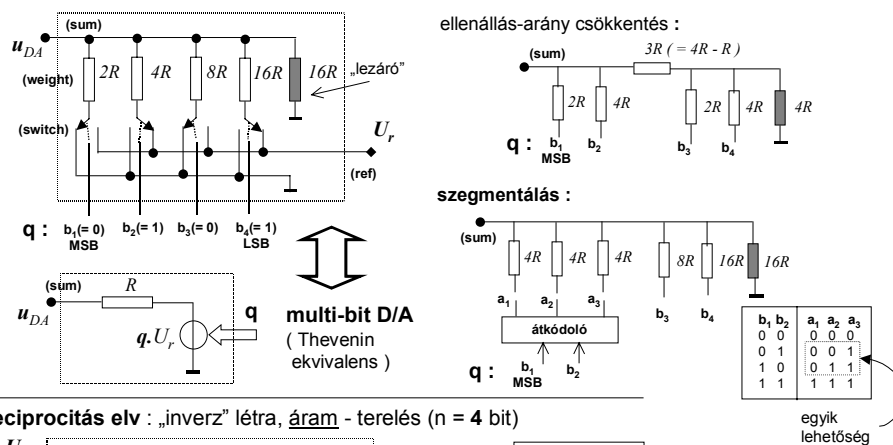
Feszültség kapcsolás, bináris arányú ellenállások⁷ ("létra") és lineáris szuperpozíció (összegzés) realizálja a **párhuzamos**, unipoláris **n bites D/A** átalakítót.

Az ekvivalens forma kézenfekvő: a **q** (normált) numerikus minta és az U_r referencia szorzata a forrásfeszültség: $U_0 = q \cdot U_r$, a forrás-ellenállás: R . (A lezáró ellenállás bit-számtól független forrás-impedanciát ad.) A terhelés **nem** módosítja a D/A funkciót (csak az u_{DA} kimenet tartománya, az átfogás változik), jó referencia kihasználáshoz "kis terhelés" kell; és a kimenet lehet áram (!) is

Impedancia eltolással csökkenthető a (nagy bitszámnál) szélsőséges ellenállás arány ($2-3-4R \rightarrow$ ismételve az eljárást: $R/2R$).

Szegmentálással - az MSB biteket „egység” elemekkel felépítve és átkódolással - javítható a linearitás (garantált monotonitás az MSB bitekre) és kisebb a kapcsolási tranzienst (az ún. glitch)

szuperpozíció elv : klasszikus „létra”, feszültség - kapcsolás (n = 4 bit)



Áram kimenet (virtuális föld pontra !!) és "azonos potenciálú" pontokat szelektáló kapcsolók (áram-terelés⁸) az ún. "inverz" létra jellemzője. A skálázásra is felhasználható áram-feszültség konverzió (invertáló műveleti erősítő) megfordítja a referencia előjelét

Ez az elrendezés - a kedvező kapcsolási feltételek kihasználásával - alkalmas **szorzó D/A** megvalósítására (az U_r referencia bemenetre kapcsolt jel amplitúdó programozása **q** értékével)

A **q** numerikus mintát tároló - az ábrán *nem* vázolt - belső adat-regiszter (DAC register) közvetlenül vezéri a kapcsolókat (NRZ mode). *Dupla* puffer szükséges (és külön adat-érvényesítő logikai jel), ha az adatátvitel szélessége kisebb a bit-számnál, vagy igen egyenletes adatfrissítés az igény

Integrált áramköröknél a *soros* adatátvitel csökkenti hatásosan a kivezetések számát, ilyen esetben az első adat-regiszter (input register) a soros/párhuzamos átalakító

⁷ Alternatív megoldás : töltés átrendezés (**charge redistribution**) elvét felhasználó kapcsolt kapacitások alkalmazása (**SC**: switched capacitor technology), elsősorban CMOS technológiánál és A/D átalakítók belső magjaként

⁸ exponenciálisan csökkenő (!) áramokat szelektálnak a kapcsolók

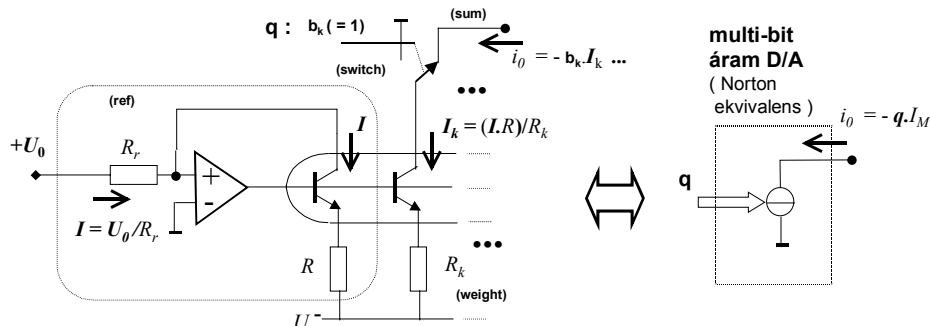
Megjegyzés: ennek az *áram*-üzemmódnak az „inverzeként” is szokásos a másik, *feszültség*-kapcsolás mód megnevezése

(2) kapcsolt áram-források current-steering

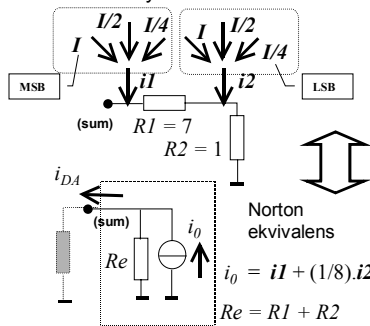
Áram összegzéssel - igazi (aktív: tranzisztoros) áram-generátor kimenet *gyors* átkapcsolásával - nagy adatfrissítési gyakoriságú, **párhuzamos D/A** átalakító valósítható meg.

Visszacsatolt szabályozási kör (feszültség – áram konverzió) formálja a konstans áram *referencia* forrást, amelynek **skálázott másolása** (áram "tükör" és súlyozás) adja a kódolásnak megfelelő és a bit-értékkel **kapcsolt** áram-forrásokat - az ábra példája csak *egyetlen* kapcsolt (aktív) forrást szemléltet ($I \rightarrow I_k$)

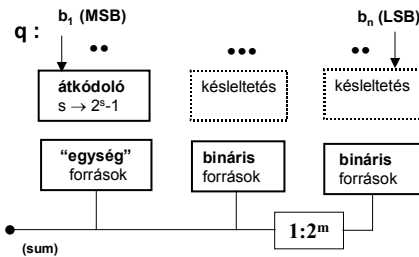
A kimenet áram ellenállással alakítható feszültséggé⁹, de korlát a források megengedett feszültség változása (**compliance range**)



áram arány csökkentés :



nagy felbontású áram D/A
(szegmentálás és áram osztás)



A nagy bit-számnál igen szélsőséges áram arány ellenállás-osztóval csökkenthető le, 2^m értékű osztáshoz $R1/R2 = 2^m - 1$. Az ekvivalens formából számítható a terhelés hatása (amely csak a referencia "kihasználást" módosítja, maga a D/A funkció változatlan!)

Az ábra $n = 6$ bit, két *egyforma* - bináris súlyozású - blokk és 1:8 osztás esetét szemlélteti.

Megjegyzés: ismételve az eljárást, határesetben *azonos* áramforrások egymást követő 1:2 osztáshoz R/2R hálózat adódik. (Ebben az esetben az R/2R „létra” csomópontjaira vezetjük a kapcsolt áram-forrásokat, tehát *nem* a létra hálózat ágaiban vannak a kapcsolók.)

Szegmentálással - átkódolással és az MSB biteket "egység"elemekkel (azaz redundáns módon) felépítve, javul a linearitás és lecsökken a kapcsolási tranziens (glitch)

A hőmérő kódolású szegmensben a szomszédos állapotok között csak *egyetlen* elem átkapcsolása szükséges, nagyobb lépésnél pedig *ugyanolyan irányú* az „egység” elemek kapcsolása

Nagy felbontáshoz az eljárások *kombinálása* a célszerű megoldás (a bináris kód „gazdaságos”, a hőmérő kódolás jó „dinamikus” tulajdonságú)

⁹ Kiseb sebesség igény esetén a linearitás is javítható: ellenállással visszacsatolt műveleti erősítő *virtuális föld* pontján keresztül végzett áram-feszültség konverzió alkalmazásával (de megfordul a polaritás).

Igen gyors átalakításhoz viszont differenciális áram kimenet (és pl. transzformátoros csatolás) célszerű

A/D átalakítók - "algebrai" modell

(a) A teljes kvantálási intervallum felosztást (az összes átváltási pontot) megvalósítva és *minden* átváltási pontot komparálva, egy órajel ütemben, közvetlenül előállítható a numerikus minta értéke ($2^n - 1 \rightarrow n$ átkódolással). Ez a **párhuzamos (word-at-a-time)** módszerű átalakító¹⁰ a *leggyorsabb* A/D; hátránya: a komparátorok száma exponenciálisan növekszik a bit-számmal

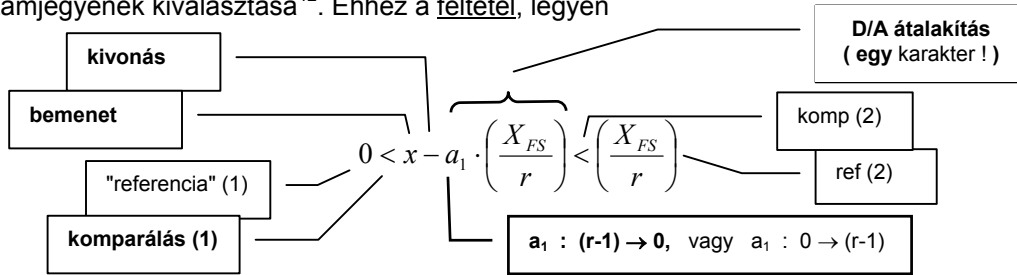
(b) Az A/D (osztási) algoritmus megvalósításának *domináns* módszere a bit keresés, vagy általánosabb - tágabb kitekintést adó - nézőpontból: a **karakterenkénti (character-at-a-time)**, *több lépéses*¹¹ (szekvenciális) módszerű mérőszám generálás.

Kódoljuk **r alapú** számrendszerben és **n helyértéken (karakterrel)** az adatot, legyen **unipoláris** az analóg tartomány: X_{FS} , így az adott **x** bemenet és a (normált) numerikus minta kapcsolata

$$x \approx q \cdot X_{FS} = \sum_{j=1}^n a_j \cdot \left(\frac{X_{FS}}{r^j}\right) = a_1 \cdot \left(\frac{X_{FS}}{r}\right) + \dots \quad \text{ahol} \quad \begin{matrix} a_j = 0, 1, 2, \dots, (r-1) \\ r \geq 2 \quad (\text{radix}) \end{matrix}$$

Speciális egyszerűsítés (topológiában is) a közvetlen **bináris** kódolás : **r = 2** és **a_j ≡ b_j = 0,1**.

Az ismeretlen a_j számjegyek megadásához az **első lépés** : a q hányados *legnagyobb helyértékű* a₁ számjegyének kiválasztása¹². Ehhez a feltétel, legyen



Ezt a feltételt teljesítő a₁ értéket (karaktert) kell "megtalálni", döntően analóg műveletekkel¹³.

Ha a karakter módosítás *szisztematikus: egyirányú* lépések, akkor elég **csak egy (!)** referenciaszint figyelés (a feltétel teljesülésének indikálásához) - ez fontos praktikus szempont. Ha pedig a szám-változtatás **a₁ : (r-1) → 0** sorrendű, akkor a kedvező **nulla komparálás** használható !! Amelyik karakter értéknél *először* teljesül a feltétel, az lesz az aktuális : $a_1 = (a_1)^*$ számjegy.

(b1) Hasonló eljárást követve, a **részmaradék (x_k)** felhasználásával¹⁴ adódnak a *kisebb helyértékű* számjegyek. Általánosan, a **k-adik lépés** : ismert minden megelőző **(a_j)^{*}**, $j = 1, 2 \dots k-1$ számjegy és ezzel a k-adik karakter-képzéshez szükséges **x_k** részmaradék is (!), a feltétel pedig

$$0 < x_k - a_k \cdot \left(\frac{X_{FS}}{r^k}\right) < \left(\frac{X_{FS}}{r^k}\right), \quad \text{ahol} \quad x_k = x - \sum_{j=1}^{k-1} (a_j)^* \cdot \left(\frac{X_{FS}}{r^j}\right)$$

Above the equation, a box states: **a_k : (r-1) → 0** karakter-módosításnál elegendő **nulla komparálás !**

(b2) Részmaradék **szorzás** művelettel (az előző referencia *megosztás* helyett), egészen más topológiákat ad a feltétel alábbi, algebrailag *ekvivalens* formája:

$$0 < r^{k-1} \cdot x_k - a_k \cdot \left(\frac{X_{FS}}{r}\right) < \left(\frac{X_{FS}}{r}\right)$$

Erre alapozva, uniformizálhatók (!) a lépések - *helyérték független* növekmények és *fokozatos* részmaradék-képzés alkalmazásával, és bár mindkét sorrendi karakter-kereső stratégia praktikus lehet, itt a közvetlen ("keresés nélküli") karakter-megadás előnyös (lásd 49. oldal)

¹⁰ "flash ADC" (one-step); *fully-parallel* (en)coder ($\rightarrow 2^n - 1$ ZX (zero crossings) per step)
¹¹ "sub-ranging ADC" (multi-step)
¹² *Emlékezzünk az osztás "papíron-ceruzával" végzett műveletére (!)*
¹³ Alternatív megoldás: "keresés nélküli", *közvetlenül egy* lépésben (!) is megadható a karakter ("flash ADC" alkalmazása, itt persze a felbontás még csak : X_{FS}/r)
¹⁴ az aktuális bemenet az *első* "részmaradék" : $x = x_1$

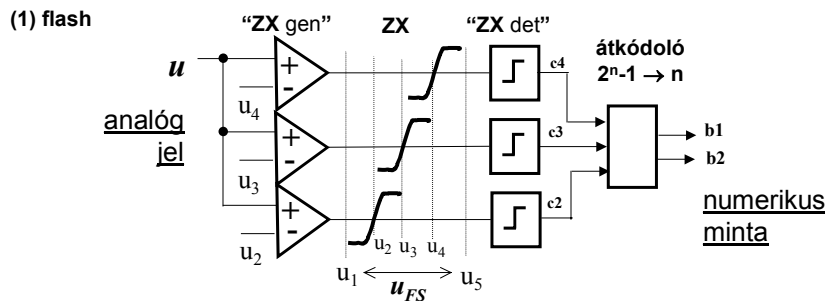
(1) közvetlen minta-kódolás flash (and interpolating¹⁵)

The ADC operation can be viewed as a collection of ZX (zero crossings) [Y. Wang]

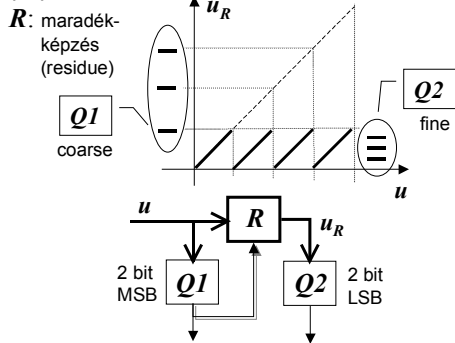
Igen gyors (ezért kis felbontású) mérőszám képzéshez, a kvantáló minden átváltási (ZX) pontot generál (pl. ellenállás osztóval: u_k) és azokat közvetlenül összehasonlítja a bemenettel

Az ábra $n = 2$ bites parallel (flash) A/D átalakítót vázol - a null-átmenetek (ZX) feltüntetésével. A döntés eredményéből, az ún. hőmérő kódból (c_k) átkódolás adja a bináris kódot (b_k).

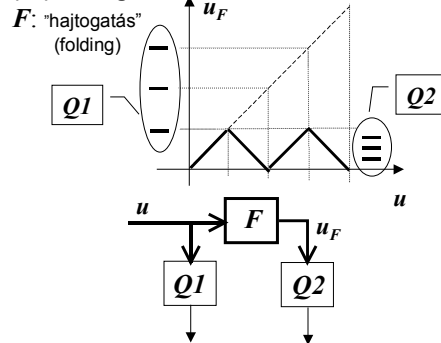
A határpontok is komparálva tartomány túllépés (OVER/UNDER range: $u > u_5$, $u < u_1$) indikálható



(2a) half-flash



(2b) folding



(2) durva/finom érték-közelítés half-flash (semi-flash)¹⁶, folding¹⁷

A megosztást az is motiválja, hogy a parallel átalakító komparátorai közül csak néhány végez igazi kritikus döntést (ezek a bemenő jelszint "közeliek"), a többi telítésben van!

A komparátorok nagy számát a bemeneti jeltartomány lineáris szegmensekre történő periódikus leképezése ("modulo" művelet) csökkenti; a rész-adatok súlyozása és összegzése az eredmény

Az ábrán vázolt mindkét - $n = 4$ bites - topológiában a szegmenst a Q1 kvantáló jelöli meg (2 bit : coarse sub-ADC; segment pointer), ezen belül Q2 ad finom felbontást (2 bit : fine sub-ADC).

(a) esetben megnő az átalakítási idő, mert a maradék-képzéshez Q1 adata szükséges, és csak ezt követi Q2 működése (half-flash; sawtooth "folding")

(b) esetben viszont Q1 működésével közel egyidejű a maradék jellegű (szakaszonként lineáris), "hajtogatott" jelforma képzése¹⁸ és Q2 döntése (folding; repetitive triangular shape)

¹⁵ Speciális analóg áramköri technika a bemenetet terhelő komparátorok számának csökkentésére (analog preprocessing): előerősítők jel-kimeneteiből lehet – aktív vagy passzív (pl. ellenállás osztó) eszközzel – újabb átváltási (ZX) pontokat generálni ("interpolálni") a felbontás növeléséhez

¹⁶ "two-step sub-ranging ADC" (residue "with postprocessing")

¹⁷ "continuous-time sub-ranging ADC" - a simultaneous coarse/fine quantizer (residue "on the fly")

"FAI (folding and interpolating) ADC"

¹⁸ Látszólag ellentmond a "hamar digitalizálni" elvnek a nemlineáris analóg blokk (F: folder) megjelenése, de éppen ez a (nagyfrekvenciás !!) konkurrens művelet segíti a maradék-képzési idő csökkentését. (Az ábrán: $F = 4$ szegmens, $F \rightarrow 1$ leképezés közvetlenül a bemenő jelből.)

Megjegyzendő, hogy csakis az átváltási (ZX) pontokat kell pontosan leképezni !!

(3) fokozatos érték-közelítés successive approximation

A **k**-edik karakter: a_k előállítását meghatározó (**b1**) feltétel (46. oldal) egyszerű átrendezésével, a részmaradék képzés magja **n**-karakteres párhuzamos D/A átalakító (referencia megosztás):

$$0 < x - \underbrace{\sum_{j=1}^n a_j \cdot \left(\frac{X_{FS}}{r^j} \right)}_{\substack{\text{n-karakteres} \\ \text{párhuzamos} \\ \text{D/A átalakító}}} \quad \text{ahol} \quad a_j = \begin{cases} (a_j)^*, j = 1, 2, \dots, (k-1) & \text{ismert(!)} \\ a_k = ? : (r-1) \rightarrow 0 \\ 0, j = (k+1), \dots, n & \text{(!)} \end{cases}$$

A D/A átalakítót vezérlő **adat-regiszter** a kiindulásnál előírt tartalmú és meghatározott karakter-keresési algoritmust követ (lásd a_j), **különbségképzés** és **nulla-komparálás** művelet adja - minden egyes számjegy-módosítást követően - a döntési információt¹⁹.

A "karakter-kereső" egy számláló (DOWN counter) - célszerűen bináris kódolással, a megnevezés: **részekre osztott számláló** típusú A/D ($r > 2$)²⁰. Maximálisan **n**-**r** számú órajel ütem (**n** lépés és egy lépésben belül max **r** számú érték-módosítás) kell egy numerikus minta megadásához.

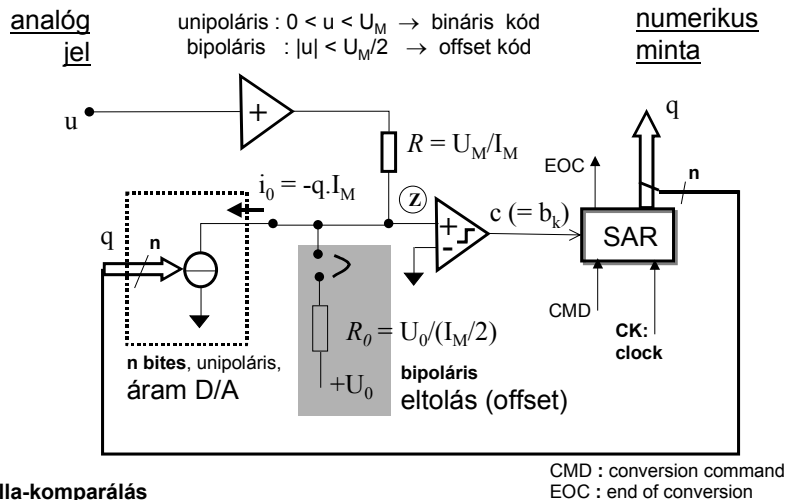
Karakter-keresés közben az **x** bemenet **nem változhat meg**²¹ (az algoritmus "nem lép vissza") !

A közvetlen **bináris** kódolás: **r = 2** és $a_j \equiv b_j = 0,1$ nagymértékben egyszerűsíti az algoritmust.

Ha ugyanis $b_k = 1$ (**TESZT**) beállításnál a komparátor a feltétel teljesülését indikálja ($c = 1$), akkor $(b_k)^* = 1$, ha pedig nem ($c = 0$), akkor **feltétlenül** $(b_k)^* = 0$ az aktuális bit érték [vagyis $(b_k)^* = c$]; és ennek rögzítésével egyidőben a következő, kisebb helyértékű bit-teszt is beállítható [$b_{k+1} = 1$].

Az adat-regiszter meghatározott - c értékétől függő - állapotokat "végigjáró" szinkron szekvenciális hálózat : **SAR**²² (a kimenet közvetlenül a tárolók kimenete, és állapot-váltás órajelre történik). A numerikus minta **n lépésben** áll elő, az MSB-vel kezdve : **bináris keresés (bit-at-a-time)**

Tipikus, feszültség bemenetű **n bites SAR A/D** topológiát szemléltet az ábra (áram-forrás D/A²³, skálázó R ellenállás, feszültség komparátor és SAR) - egyszerű bipoláris (offset kódú) opcióval



Ⓣ **nulla-komparálás**
ad döntési információt (a bipoláris eltolás opcionális)

$$u \cdot \frac{R_0}{R + R_0} - q \cdot I_M \cdot \frac{R \cdot R_0}{R + R_0} + U_0 \cdot \frac{R}{R + R_0} = \left(u - \left[q \cdot U_M - \frac{U_M}{2} \right] \right) \cdot \frac{1}{1 + (R/R_0)}$$

¹⁹ jellegzetes visszacsatolt topológia (**feedback** ADC)

²⁰ Speciálisan, "egy részre osztásnak" felel meg a **lineáris keresés (leve-at-a-time)** módszere (\rightarrow az adat-regiszter egyetlen bináris számláló !); itt lehet "UP counting" is, vagy ún. "követő" (UP/DOWN) eljárás

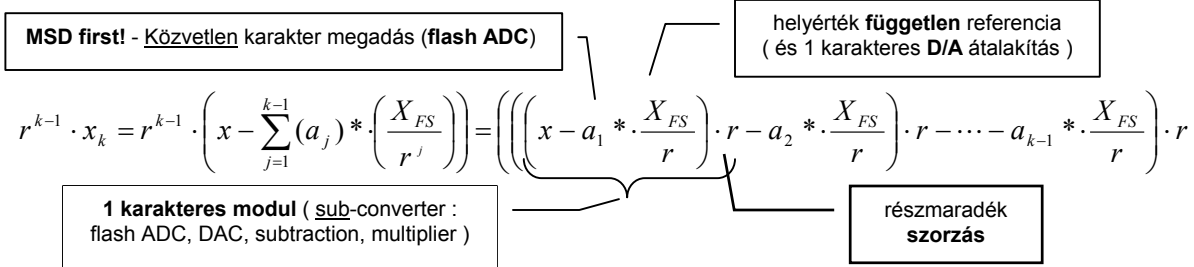
²¹ Változó jel (x) esetén "érték-megtartó" **mintavevő** : **THA** (vagy SHA) **kell** a bemeneten

²² "**SAR (successive approximation register) ADC**" (\rightarrow 1 ZX (zero crossing) per clock-step)

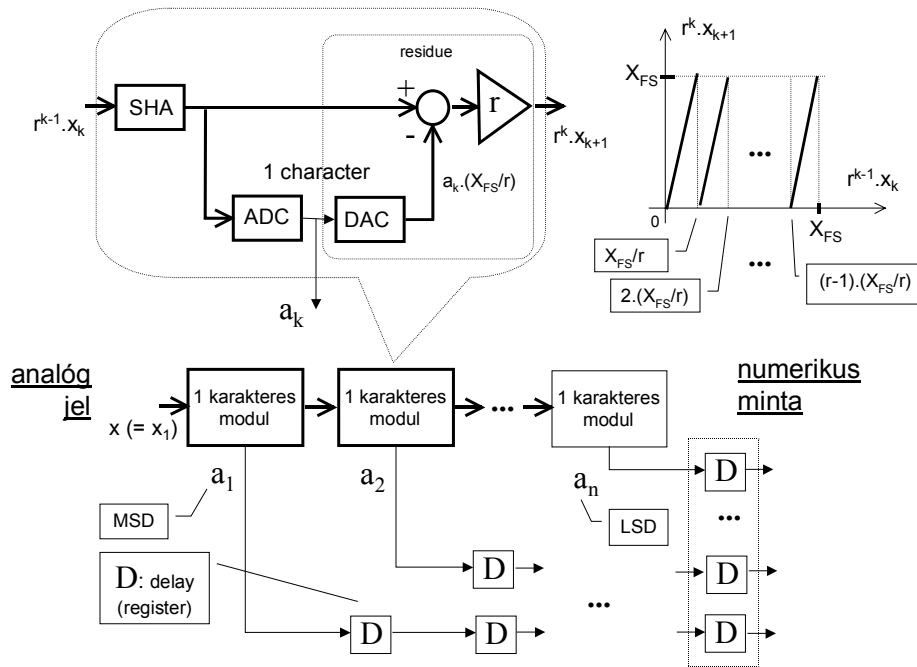
²³ Alternatív megoldás a kapacitív D/A (amely egyben mintavevő is!) : "**charge redistribution SAR ADC**"

(4) konkurens minta-kezelés pipeline

A (b2) feltétel (46. oldal) részmaradék tényezőjének alábbi *ekvivalens* formája alapján, fokozatos részmaradék képzést használva és egyforma 1 karakteres modulok egymást követő (kaskád) kapcsolásával is generálható a mérőszám



Az azonos felépítésű, **kaskád** fokozatok közül csak egyetlen aktív ("terjed az érték-közelítés")²⁴. A fokozatok műveletvégzése párhuzamosítható **tárolás** (analog memória: SHA) beiktatásával! Ilyen, **n** modulból álló, **konkurens minta-kezelésű A/D** átalakító²⁵ topológiát vázol az ábra



A modulok *különböző* (egymást követő) mintákat kezelnek, ezért - a karakterek ütemezését követően - csak **egy** fokozat műveleti ideje az alapkorilát. *Folyamatos* mintagyakoriság esetén tehát **radikálisan megnő** az átviteli képesség (word rate) - közelíti az "egy lépés (one step)" átalakítási időt, persze a kezdeti terjedési késleltetés (pipeline **latency**) megmarad

A közvetlen **bináris** kódolás: **r = 2** és **a_j ≡ b_j = 0,1** nagymértékben *egyszerűsíti* a felépítést.

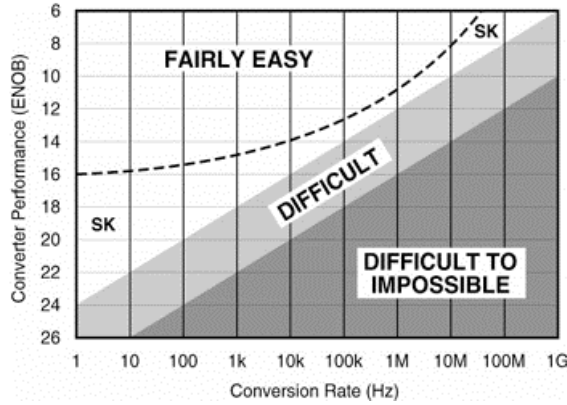
A modulok *illesztése* nehéz feladat. Egyik trükk: a modul-tartományok **átlapolása** (redundancia) és digitális **korrekció**; pl. három-szintű kvantálás ("1.5 bites" modulok), de - 2-szeres erősítéssel és utólagos korrekcióval - csak "1 bit" kihasználása fokozatonként (**DEC** : digital error correction)

²⁴ jellegzetes *előrecsatolt* topológia (**feedforward** ADC; ripple through)

²⁵ "**pipeline** ADC" - successive samples *simultaneously* in different stages

Alternatív, "takarékos" (de lassú: **n** lépéses) megoldás az ún. **ciklikus érték-közelítés**: *egyetlen* fokozat *ismételt* (ciklikus) felhasználása - "**algorithmic** (cyclic) ADC" (→ **R²** : recirculating remainder)

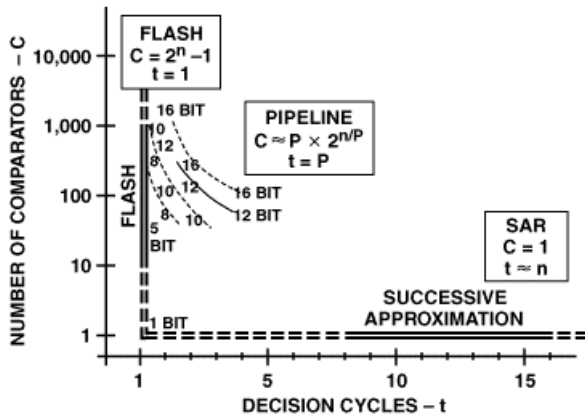
"Ever hear the story of the design engineer who wanted a 32-bit analog-to-digital converter (ADC) to go with his 32-bit microcontroller? Or the graduate student who desperately needed a 16-bit, 100 MHz ADC in order to build a prototype of his computer simulated design? The **figure** shows a rough estimate of the difficulty in obtaining a given digital representation of an analog signal *versus* conversion rate (**SK** : specialized knowledge).



The "difficult" area of figure changes over time, **moving slowly** down and to the right. As with digital ICs, mixed-signal ICs follow a trend similar to Moore's Law. Eric Swanson of Cirrus Logic is said to have offered a similar law for mixed-signal ICs which says that their dynamic range increases by about 2 dB per year or about 1 bit every three years *. That sounds about right. " [J. Horn 2000]

* The performance is improving much slower than Moore's law, Walden (1999) estimated "1.5 bits / 8 years" improvement rate

"The most popular ADC architectures available today are successive approximations (sometimes called SAR because a successive-approximations register is the key defining element), flash (all



decisions made simultaneously), pipelined (with multiple flash stages), and sigma-delta ($\Sigma\Delta$). All A/D converters require one or more steps involving *comparison* of an input signal with a reference. **Figure** shows *qualitatively* how flash, pipelined, and SAR architectures differ with respect to the number of comparators used *versus* the number of comparison cycles needed to perform a conversion.

While not exhaustive, the following **table** summarizes and *ranks* (in a generalized sense) the **relative advantages** of flash, pipelined, SAR, and sigma-delta architectures. A rank of 1 in a performance category indicates that the architecture is inherently better than the others in that category are. A * indicates that the architecture has the capability or characteristic listed.

Characteristics	Flash	Pipelined	SAR	$\Sigma\Delta$
throughput (sample rate)	1	2	3	4
resolution (ENOB)	4	3	2	1
latency	1	3	2	4
capability to convert non-periodic multiplexed signals	1	2	1	3
simplified anti-aliasing (AAF)				*
can undersample	*	*	*	
can increase resolution through averaging (with dither noise)	*	*	*	

Such generalizations are useful for the *system designer* to keep in mind when conducting a high level overview of a proposed system's requirements. " [B. Black 1999]

Characteristics of ADCs per application:

Application	Converter Architecture	Resolution	Sampling Rate
Audio	$\Sigma\Delta$ $\Sigma\Delta$, high-order (4th-7h) $\Sigma\Delta$ Successive approximation	14-18 bits for consumer audio 18-24 bits for professional equipment 10-16 bits	48-50 kS/s 48-96 kS/s (192 kS/s) 85 to 500 kS/s
Automatic control Sensors and Robotics	$\Sigma\Delta$ Successive approximations Integrating $\Sigma\Delta$ Half-flash (for high-speed servo control)	6 bits 8-16 bits 18-20 bits 16 bits 8 bits	250 S/s 40-500 kS/s 100-2000 S/s 192 kS/s 200-400 kS/s, 1MS/s
Data transmission (cable)	$\Sigma\Delta$ and high-order (4th) $\Sigma\Delta$ for ISDN and ADSL Pipeline Half-flash	12-16 bits for cancelling modems 13-16 bits for ISDN transceivers 12 bits for ADSL 12 bits for VDSL 8-12 bits	8 kS/s for modems 80-160 kS/s for ISDN transceivers 2.2 MS/s for ADSL 40 MS/s for VDSL 400 kS/s-1.5 MS/s
Digital high-speed instrumentation	Flash, interleaved flash Flash-successive approximations hybrid Successive approximations Pipeline	8 bits 10-12 bits 12-16 bits 8-12 bits 12 bits	150 MS/s-1 GS/s 10-40 MS/s 85-166 kS/s 500 kS/s - 1.5 MS/s 30-53 MS/s
Geophysical	$\Sigma\Delta$	16-24 bits	1-32 kS/s
Hard disk driving	Half-flash Pipeline Successive approximations Flash	10 bits 8-12 bits 8 bits 6 bits	320 kS/s 800 kS/s-1.5 MS/s 100 kS/s 30-140 MS/s
Medical	Interleaved flash (for CAT) Flash Successive approximations $\Sigma\Delta$ Pipeline	8 bits 8 bits 8-16 bits 16 bits 12-14 bits	150-750 MS/s 1 GS/s 50-500 kS/s 192 kS/s 30-70 MS/s
Military communications Electronic warfare	Flash, interleaved flash Subranging, pipeline, Folding and interpolating	8 bits 12-14 bits	150-750 MS/s 50-100 MS/s
Mobile telecommunications and Wireless communications	Successive approximation for GSM $\Sigma\Delta$ and high-order (4th) $\Sigma\Delta$ for GSM Half-flash Pipeline Flash Flash and interleaved flash Pipeline	8 bits for GSM 13 bits for GSM 8-10 bits 8-14 bits 6 bits for satellite 8 bits 12-14 bits	270 kS/s for GSM 320-500 kS/s, 1 MS/s 800 kS/s-65 MS/s 40-80 MS/s 150 MS/s-1 GS/s for RF 48-65 MS/s for IF
Monitoring, Test equipment and Instrumentation	$\Sigma\Delta$ Half-flash Successive approximations $\Sigma\Delta$	22-24 bits 8 bits 8-16 bits 16 bits	≤ 1 kS/s 400 kS/s 73-500 kS/s 192 kS/s
Radar and Sonar	Flash, interleaved flash Pipeline Subranging, pipeline, Folding and interpolating for radar $\Sigma\Delta$ for sonar	8 bits 12 bits 12-14 bits for radar 16-18 bits for sonar	150 MS/s - 1 GS/s 10-30 MS/s 50-100 MS/s for radar 200 kS/s for sonar
Spectrum analysis	Pipeline Successive approximations	10-12 bits 12-14 bits	10-40 MS/s 300-500 kS/s
Speech and voice communications	Successive approximation, $\Sigma\Delta$	11-14 bits	8 kS/s
Video and Television	Half-flash for professional video Pipeline	8-12 bits 8-12 bits	10-40 MS/s 30-50 MS/s

Feladatok - 5

5.1 Hány különálló kapcsolt forrás szükséges **(a)** bináris kódú, **(b)** tisztán egység elemekből álló (átkódolás után: hőmérő kódú), illetve **(c)** szegmentált n bites **D/A** átalakító felépítéséhez?

5.2 (a) Szemléltessük **D/A** átalakítónál a kapcsolási tranziens (**glitch**: error in transition instant) keletkezését: két forrás (referencia növekmény) egyidejű, *eltérő állapotú* szelektálása (ON/OFF) és a be- és kikapcsolási terjedési késleltetési idők eltérnek (data skew).

Miért MSB váltásnál várható a legrosszabb eset?

(b) Miért „garantált” a monoton átmenet hőmérő kódolásnál?

5.3 Nagy felbontású ellenállás (pl. R/2R) hálózat lezáró ellenállását egy *zárt* kapcsoló (dummy switch) köti a földhöz; vagy „inverz létra” esetén a visszacsatoló R_v ellenállással sorba kötve is találunk egy *zárt* kapcsolót (lásd **1.** példa). Mi ennek az oka?

5.4 Miért az analóg *felező* (LSB first) forma előnyös a soros **D/A** megvalósításánál (szemben az analóg *kétszerező* formával)?

5.5 Mennyire csökkenti le a két lépéses, durva/finom érték-közelítésű **A/D** átalakító (half-flash ADC) a komparátorok számát a flash A/D átalakítóhoz képest (n bit)?

5.6 Fokozatos érték-közelítésű **A/D** (SAR ADC).

(a) Az algoritmus miért az MSB-vel indul? Lehet-e előnyös, hogy az MSB bit(ek) – a mérőszám „jó” becslése – viszonylag gyorsan előáll?

(b) Adjuk meg a szukcesszív approximációs regiszter teljes állapotdiagramját (n = 3 bit), és kell konverzió kész: EOC állapot indikálás is. (Hogyan realizálható ez a szinkron szekvenciális digitális áramkör?)

(c) Lehet-e működési ciklust rövidíteni (felbontás „adaptálás”: pl. adott n = 12 → ebből elég 8 bit)? Mi indikálja a „rövid ciklus” végét?

(d) *round*() kvantálási művelethez a bemeneten $+1/2 \cdot \Delta x$ eltolás (**half-LSB offset**) szükséges, miért? (Hogyan látható ez be pl. a bit keresés feltételéből?) Ha nem alkalmazunk ilyen eltolást, milyen a kvantálási hiba tartománya?

(e) Bipoláris esetben (**MSB offset**) hogyan kaphatunk komplement kódú numerikus mintát?

Megjegyzés: vegyük észre, hogy (1) a bipoláris **eltolás** és az unipoláris áram **D/A** referenciája *ugyanaz* az érték ($+U_0$) lehet, és (2) ez az eltolás csak a feszültség komparálás amplitúdó viszonyait rontja, de *nem* módosítja A/D funkciót: a bit generálás feltételét! (Lásd még 9. oldal.)

(f) Hogyan változtatható meg az átalakító U_M bementi jeltartománya (skálázás)?

(g) Hasonlítsuk össze az algoritmust az egyszerű *lineáris* kereséssel (ha az adatregiszter *egyetlen* bináris számláló)! Hány órajel ütem kell n bites adat előállításához a kétféle esetben?

5.7 Hasonlítsuk össze az 1 bites modul használó ciklikus (R^2 : részmaradékot recirkuláló) **A/D** (→ hiba *kétszerezés*) és a SAR **A/D** (→ referencia *felezés*) struktúráját (n bit). Mennyi a numerikus minta előállításához szükséges órajel ütem-szám?

Lehet-e SAR A/D esetén *közvetlenül* soros adat-kimenetet előállítani?

5.8 Vessük össze a konkurens minta-kezelésű **A/D** (pipeline ADC) és a SAR **A/D** átalakítót, ha „multiplexelt (több csatornás)” a jel bemenet vagy igen egyetlen a mintavételi gyakoriság, illetve ha folyamatos és egyenletes a mintagyakoriság igény.

5.9 (a) Miért lehet előnyös a „beépített” FIFO (first-in first-out) memória egy **A/D** átalakítónál?

(b) Hogyan illeszthető „gyors” (nagy mintagyakoriságú) **A/D** átalakító „lassú” memóriához?

5.10 Nagy mintagyakoriság eléréséhez jól ismert *rendszeres* fogás: időben eltoló mintavételű és párhuzamos működésű két (vagy több) *azonos* **A/D** átalakító alkalmazása (time-interleaving; lásd **5.** példa).

Hogyan „működik” ez az elv **D/A** átalakítás adatfrissítési gyakoriságának növelésénél?