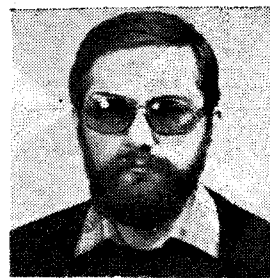


# Az INTEL D-2920 analóg mikroprocesszor alkalmazása

FAZEKAS DÉNES  
Távközlési Kutató Intézet



## ÖSSZEFOGLALÁS

Az INTEL D—2920-at kifejezetten analóg feladatok megoldására fejlesztették ki. Segítségével olyan feladatok is könnyen meg lehet oldani, melyek korábban sok fejtörést okoztak a tervezőknek. Ilyenek pl. nagypontosságú szűrők, hullámaforma generátorok, nemlineáris funkciók realizálása. A cikk megfigyeli a mikroprocesszor felépítésével és működésével. Speciális példákat is bemutat, amelyek telefoncsatorna foglaltság figyelésére szolgálnak.

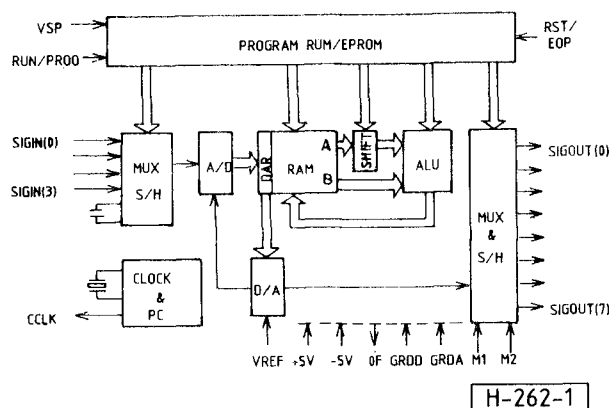
## Bevezetés

A D—2920 egychipes mikrocomputer, melyet kifejezetten analóg jelek real-time feldolgozására fejlesztettek ki. Tartalmazza a program memóriát, a RAM-regisztereket, a D/A és A/D konvertert, valamint az I/O áramköröket. A maszkprogramozott változat — ha a teljes program memóriát kihasználjuk — kb. 13 kHz-es mintavételi frekvenciával dolgozik. (Az EPROM-változat ennél valamivel lassúbb, a mintavételi frekvencia kb. 8,5 kHz.) Rövidebb program esetén ez az érték nagyobb, így a feldolgozható jel sávszélessége is megnő.

## A mikroprocesszor felépítése, működése

### Funkcionális egységek

A mikroprocesszor felépítése az 1. ábrán látható. A program a maximum 192 lépésben programozható programmemóriában található. Az aritmetikai egység tartalmazza a 40 szavas, tetszőleges hozzáférésű RAM-memóriát és az ALU-t. Ez



1. ábra. Az INTEL D—2920 felépítése

Beérkezett: 1986. XI. 5. (□)

FAZEKAS DÉNES  
Diplomáját 1979-ben szerzte a Budapesti Műszaki Egyetem Villamosmérnöki Karán 1982-óta dolgozik a Távközlési Kutató Intézetben. Digitális jelfeldolgozással foglalkozik, s

eddig munkája során több, a gyakorlatban alkalmazott analóg mikroprocesszoros egység fejlesztését végezte. Kutatási területe ezen kívül a nagy felbontóképességű, gyors A/D konverterek mérése.

utóbbi egyik bemenetén megjelenő adatok egy barrel shifter segítségével  $2^n$ -nel szorozhatók ( $-13 \leq n \leq 2$ ).

Az analóg rész a következőket tartalmazza:

- 4 bemenetű multiplexer,
- mintavevő-tartó áramkör,
- D/A-konverter,
- komparátor,
- kimeneti demultiplexer, 8 db mintavevő-tartó áramkörrel.

Egy speciális, DAR-nevű regiszter hozza létre a kapcsolatot a digitális és az analóg egységek között.

### Analóg műveletek

A bemenő multiplexer által kiválasztott jel a mintavevő-tartó áramkörre kerül. Ezután a processzor A/D konverziót hajt végre, melynek eredménye, egy 9 bites szó (8 bit plusz előjel), a DAR-ban jelenik meg.

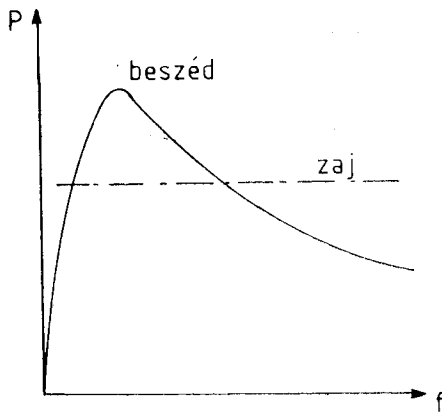
A feldolgozott jel kimenetre juttatásához a kiválasztott RAM-regiszter tartalmát a DAR-ba töltjük. Ezután a D/A-konverter áram a kimenő demultiplexer és mintavevő-tartó áramkör közvetítésével a 8 db kimenet egyikére jut.

### Digitális műveletek

A digitális műveleteket a D—2920 az analóg műveletekkel egy időben végzi. A program egyszerre két RAM-regisztert címez meg (A és B regiszterek). Az ALU elvégzi a kijelölt műveletet, majd az eredményt B-be írja vissza.

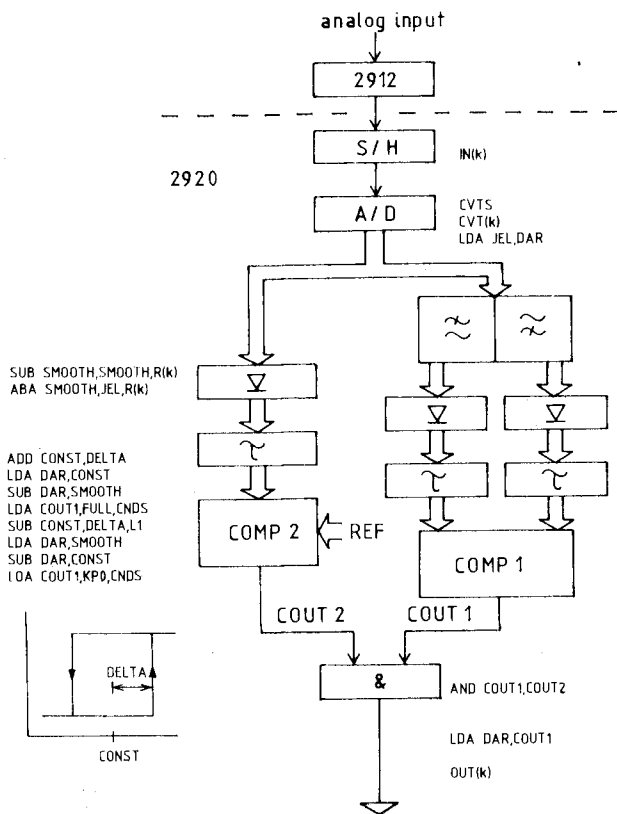
Az elvégezhető digitális műveletek:

- értékadó utasítás (LDA),
- összeadás (ADD),
- kivonás (SUB),
- logikai szorzás (AND),
- kizáró vagy (XOR),
- abszolút érték képzés (ABS),
- egy regiszter tartalmához egy másik regiszter abszolút értékének hozzáadása (ABA),
- az ábrázolható legnagyobb és legkisebb érték előállítás (LIM).



H-262-2

2. ábra. Fehér zaj és beszéd jel spektruma



H-262-3

3. ábra. Fehér zaj és beszéd jel megkülönböztetése váltószűrővel

Az első négy művelet végrehajtása egy feltétel bit logikai állapotához is köthető. Ha ez „0”, akkor az adott utasítás helyett egy üres utasítás kerül végrehajtásra.

#### Alkalmazási példák

A mikroprocesszor felépítésének, működésének rövid áttekintése után bemutatunk két megvalósított, a gyakorlatban is működő alkalmazási példát.

##### Első példa

Egy adott információátvételi csatornáról meg kell állapítani, hogy az foglalt-e. — Azaz, van-e

rajta terhelés, és ha igen, az hasznos-e. (Jelen esetben hasznos terhelésen a beszéd jelet értjük.)

A megoldásnál abból indulunk ki, hogy a vizsgált csatormán esetleg megjelenő zaj a 200—2500 Hz tartományban fehér zaj jellegű (2. ábra), a beszéd jel teljesítménye ezzel szemben a magasabb frekvenciákon erősen csökken.

A jelfeldolgozás folyamatábrája a 3. ábrán látható. A mintavételezés miatt szükséges sávkorlátozást egy 2912 típusú áramkör végzi. A további feldolgozás már a processzoron belül történik.

A jelet mintavételezzük (S/H), majd A/D konverziót hajtunk végre. A jel útja itt elágazik. Az egyik út egyenirányítás és szűrés után egy komparátorra vezet. A komparátor referenciája egy software úton programozott konstans. Amennyiben a beérkező jel nem éri el ezt a szintet, úgy azt minden esetben zajnak tekintjük — függetlenül az egyéb jellemzőitől.

A másik út egy digitális váltószűrőre vezet. A keresztelési frekvenciát úgy választottuk meg, hogy ha a szűrő bemenetére beszéd jellegű jel kerül, akkor az aluláteresztő kimeneten jelenik meg a nagyobb teljesítmény.

A szűrőkimenetek jelét a korábbiakhoz hasonlóan egyenirányítjuk és simítjuk. Egy differenciál komparátor dönti el, hogy melyik a nagyobb, azaz, a jel spektruma beszéd vagy zaj jellegű.

A zaj/beszéd döntés túlságosan gyakori ingadozásának elkerülésére a komparátor megfelelő hiszterézissel rendelkezik.

Ezután nézzük meg, hogy a 2920 segítségével hogyan realizálhatók az előzőekben leírt funkciók.

A mintavevő-tartó az IN(k) analóg utasítással valósítható meg. Itt k a kiválasztott bemenet száma, értéke 0 és 3 között lehet. Tekintve, hogy a tartó kapacitás feltöltődéséhez szükséges idő általában több, mint egy utasítás végrehajtási ideje, az IN utasítást néhányszor meg kell ismételni.

Az A/D konverzió a DAR-regiszter nullázásával kezdődik, pl. LDA, DAR, KP 0. Ezután a CVTS, CVT(k) utasításokkal szukcesszív approximációs A/D konverziót hajtunk végre. A konverzió közben tekintettel kell lenni a D/A konverter beállási idejére, ezért a CVT utasítások között üres analóg utasításokat kell hagyni.

A konverzió útján kapott érték a DAR-ban található, mivel várhatóan többször is fel fogjuk használni, célszerű áttölteni egy külön regiszterbe, az LDA JEL, DAR utasítással.

Az egyenirányítás és simítás igen egyszerűen, mindössze két utasítással elvégezhető. Legyen SMOOTH az a regiszter, amelyben a simítás után az eredményt kapjuk. A program a következő:

```
SUB SMOOTH, SMOOTH, R(k),
```

```
ABA SMOOTH, JEL, R(k).
```

— Azaz a SMOOTH regiszterből kivonjuk a saját,  $2^k$ -nal osztott értékét, majd a következő lépésben hozzáadjuk a bemenő jel ugyanannyival osztott értékét.

Ennél valamivel bonyolultabb a hiszterézises komparátor: a komparálási szintet meghatározó konstans értékét a pszeudoregiszterekből állítjuk elő és betöltjük a CONST-regiszterbe. A hiszterézis

hasonlóan előállított értéke a DELTA-regiszterben található. Első lépésként előállítjuk a hiszterézis tartomány felső határát:

ADD CONST, DELTA.

Ezután CONST-értékét áttöltjük a DAR-ba, és kivonjuk belőle SMOOTH-értékét.

Ha DAR a kivonás után negatív, akkor a komparátor bemenetére adott jel nagyobb, mint a felső komparálási szint, a COUT1-regiszterbe „1” írható. Ezt végzi el az

LDA COUT1, FULL, CNDS

utasítás. (Full-regiszterbe korábban  $1-2^{-24}$  értéket töltöttünk be.) A CNDS-feltétel megadásával a processzor csak akkor végzi el a műveletet, ha a DAR-előjel bitje 1. Ellenkező esetben egy üres utasítást (NOP) hajt végre.

Az alsó komparálási szinttel való összehasonlítás — amint az a 3. ábrán látható — hasonlóan történik.

Visszatérve az elágazáshoz, látjuk, hogy a jel a másik úton egy váltószűrőre kerül. Az egyenirányító és simító algoritmus hasonló az előzőekben ismertetethez, ugyanez mondható el a differenciálkomparátorról is.

Az eredmény a két komparátor kimenetének logikai ÉS függvénye. Áttöltve ezt az értéket a DAR-ba az

LDA DAR, COUT1 utasítással, a kimeneten OUT(k)

(Itt  $k$  a kiválasztott kimenet száma,  $0 \leq k \leq 7$ .)

#### Második példa

A feladat hasonló az előbbihez, azzal a kiegészítéssel, hogy egy mikroprocesszorral több csatorna jelének feldolgozását végezzük el, és a csatornákon megjelenő távíró jelet is hasznos jelnek kell tekinteni.

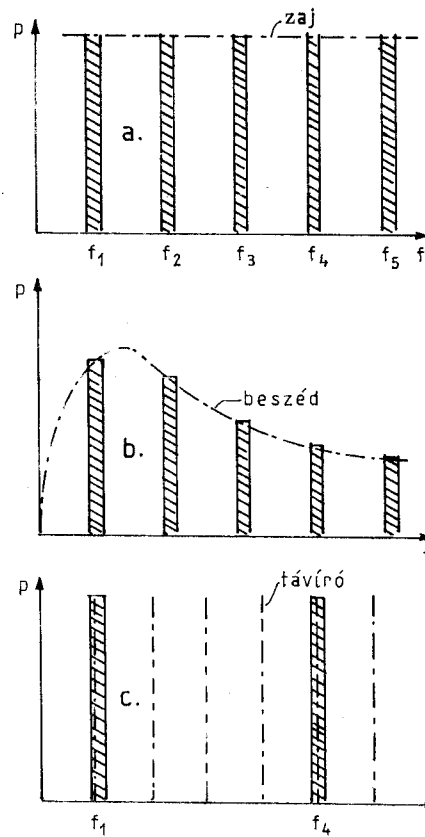
A döntés itt is spektrális jellemzőkön alapul. Megvizsgáljuk a jel teljesítményét 5, néhányszor 10 Hz-es frekvencia tartományban. A 4. ábrán ezeket  $f_1 \dots f_5$ -tel jelöltük. Ez az öt pont 200 és 2500 Hz között úgy helyezkedik el, hogy 6, 12 vagy 24 csatornás távíró jel esetén is legyen legalább egy olyan sáv, amelyik egybe esik valamely távíró frekvenciával (pl. a 4c. ábrán  $f_4$ ), és legyen legalább egy olyan sáv, amelyikbe nem esik távíró jel (pl.  $f_5$ ).

Az egyes frekvencia tartományokban megjelenő teljesítményeket alkalmas időállandóval simítva a következőt tapasztaljuk: fehér zaj esetén az egyes sávokban mért teljesítmények aránya 1-hez tart. Beszéd jel esetén a mért minimum és maximum között jelentős az eltérés. Távíró jel esetén lesz olyan sáv, ahol csak a csatorna alapzaj teljesítménye mérhető, és olyan is, ahol teljesítmény maximumot kapunk.

A jel feldolgozásának folyamatábráját az 5. ábra mutatja.

A csatorna kiválasztást egy analóg multiplexer végzi. Utána a már ismert 2912 sávkorlátozó szűrőt látjuk.

Olyan digitális sáváteresztő szűrőt, amelynek karakterisztikája változtatható, bonyolult létrehozni, ezért a program egy fix karakterisztikájú szűrőt és egy változtatható frekvenciájú oszcillá-



H-262-4

4. ábra. Az információ átviteli csatorna jelei és a kiválasztott frekvenciasávok közötti kapcsolat

tort tartalmaz. Az oszcillátor frekvenciája  $T$  időnként változik. A frekvenciákat úgy választottuk meg, hogy keverés után a bemenő jelnek a korábbiak szerint kiválasztott tartományai jussanak a sáváteresztő szűrőre, azaz  $f_{oszc}^k = f_{BP} - f_{in(k)}$ .

A jel átlagos teljesítményét a  $T$  időtartamra az

$$\frac{1}{T} \int_0^T x^2(t) dt$$

alapján kapjuk meg. Ezt jelen esetben úgy valósítjuk meg, hogy a sáváteresztő szűrő kimenő mintáit négyzetre emeljük és  $T$  ideig összegezzük. Mivel számunkra az így kapott értékek hányadosa az érdekes, ezért a  $T$ -vel való osztás elhagyható.

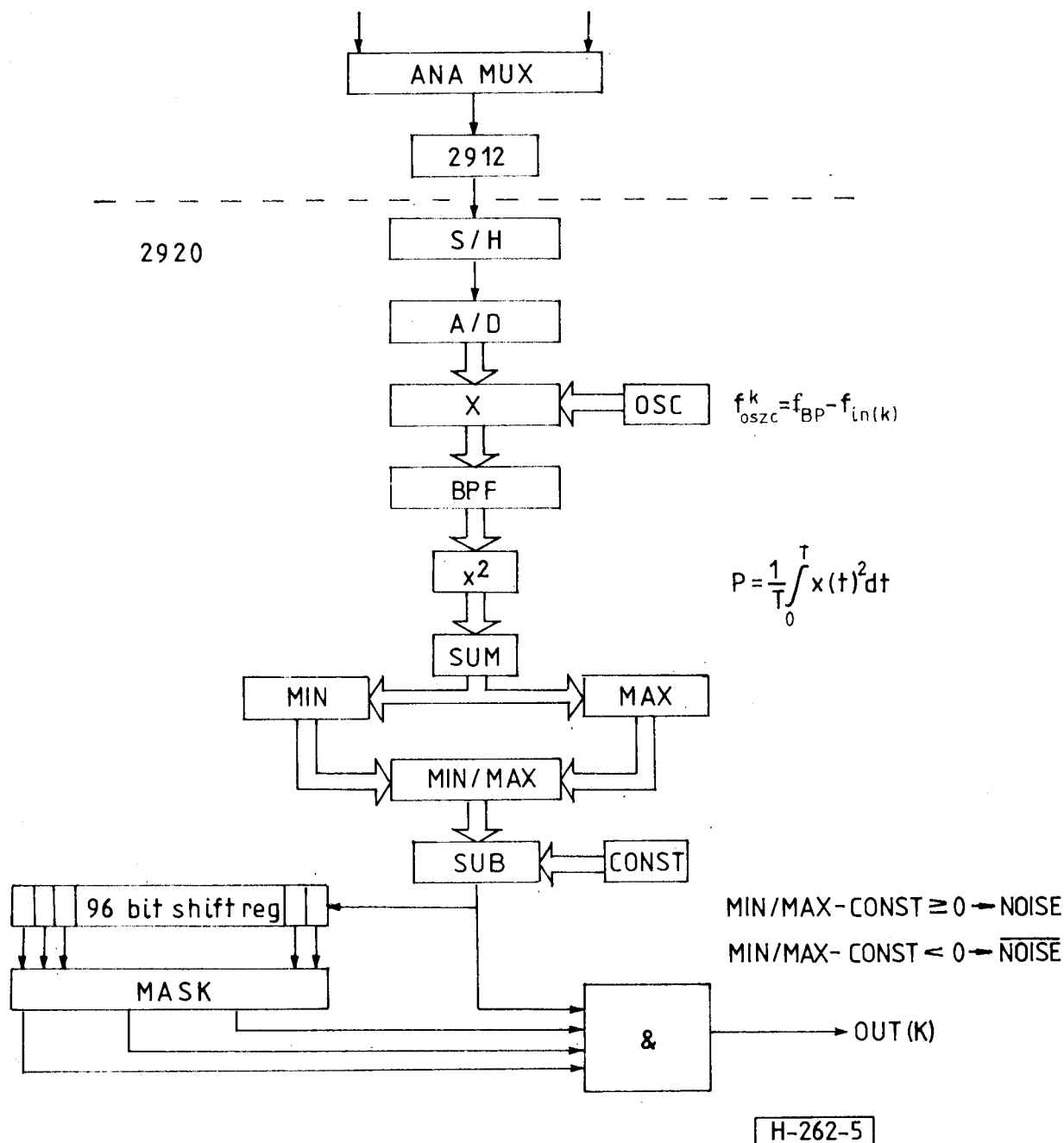
Egy csatorna vizsgálata öt lépésben történik. Az elsőben az oszcillátor  $f_{oszc}^1$  frekvenciát állít elő. Keverés és szűrés után a kapott mintákat négyzetre emeljük és  $T$  ideig összegezzük.  $T$  idő eltelével az összeget beírjuk a MIN és MAX regiszterekbe és nullázzuk az összeget korábban tartalmazó SUM-regisztert.

A következő négy lépésben az összegzés hasonló módon történik, azonban a  $T$  időtartamok végén megvizsgáljuk, hogy teljesül-e a

SUM < MIN vagy

SUM > MAX feltétel, és ha szükséges, a MIN vagy MAX értékét felülírjuk.

5  $T$  idő elteltével meghatározzuk a MIN/MAX



5. ábra. Foglaltságfigyelés sáváteresztő szűrő segítségével

arányt. Ha ez kisebb, mint egy előre megállapított konstans, azaz  $\text{MIN/MAX-CONST} > 0$ , akkor a vizsgált jel táviró vagy beszéd jel, ellenkező esetben pedig zaj.

Ezután a processzor egy impulzussal a következő csatornára lépteti az analóg multiplexert, és az öt lépésből álló ciklus újra kezdődik.

Az eddigiekből kiderül, hogy egy csatornát csak igen rövid ideig (kb. 300 ms) vizsgálunk, ezért ha csak egy vizsgálat eredményére hagyatkoznánk, kicsi lenne a valószínűsége annak, hogy helyes döntést hozunk. — Gondoljunk csak pl. a szavak, mondatok között tartott szünetekre, amikor a csatornán csak annak saját zaja észlelhető.

Elfogadható értékű lesz a helyes döntés valószínűsége, ha az ugyanazon csatornán végzett korábbi mérések eredményét is figyelembe vesszük.

Az ismertetett áramkör 32 csatorna figyelését végzi. Ha a három utolsó döntést a szokásos módon tárolni akarjuk, ez összesen 96 RAM regisztert igényel. — Viszont a 2920-nak csak 40 RAM regisztere van. Kihasználhatjuk azonban, hogy a tárolandó információ 0-k és 1-ek sorozata, és hogy egy regiszter hossza 24 bit. 4 db 24 bites regiszterből software úton kialakítottuk egy 96 bites shift regisztert, amelybe sorban betöltjük az egyes mérések eredményét. A shift regiszter tartalmát a csatornák váltásával szinkronban léptetjük. Az

így tárolt 96 bitből egy software maszk segítségével kiválasztjuk azt a hármát, amely az aktuális csatornán végzett korábbi vizsgálatok eredményét hordozza. Ha a rendelkezésünkre álló négy bit logikai szorzata „1”, a csatorna szabadnak tekinthető, ellenkező esetben beszéd vagy távíró forgalom zajlik.

### Összegzés

Az itt ismertetett példákból is látható, hogy analóg

mikroprocesszor alkalmazásával egy sor analóg áramkör megépítését, bemérését kerülhetjük el. Az áramkörök tervezése jelentősen egyszerűsödik, egy analóg mikroprocesszoros egység több feladat megoldására is felhasználható, hardware változtatás nélkül. Az áramkörök tervezése során a software fejlesztés kerül előtérbe.

### I R O D A L O M

- [1] INTEL: 2920 Analog Signal Processor Design Handbook.
-