

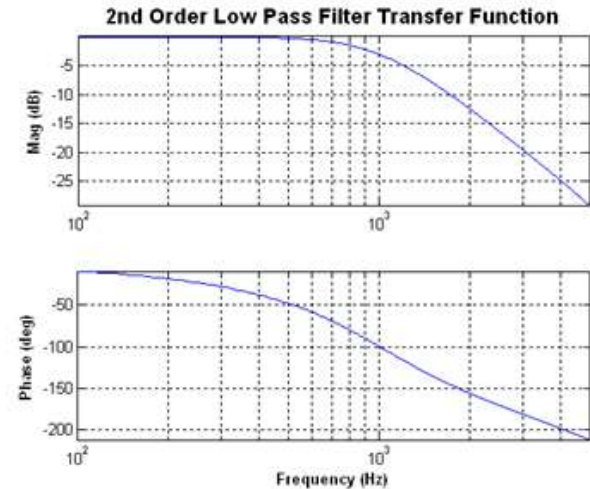
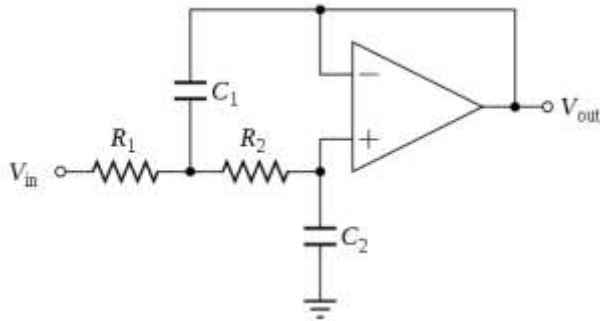
Procesorski sistemi v telekomunikacijah
Signalni procesorji in digitalna obdelava signalov



(c) Arpad Bűrmen, 2010-2012


Digitalna obdelava signalov

- ▶ Analogni obdelava signala – npr. filtriranje



- ▶ Digitalna obdelava signala



 zvezen (analogen) signal

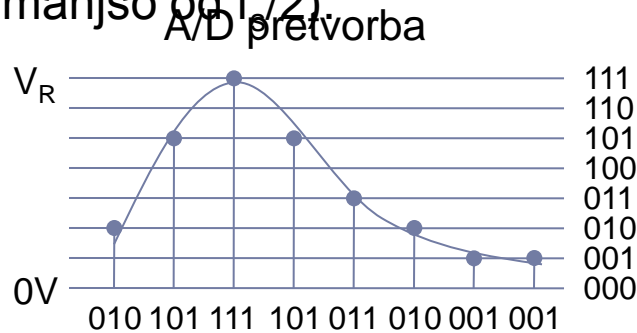
 časovno in vrednostno diskreten (digitalen) signal

A/D pretvorba

- ▶ Signal se vzorči s periodo T_s (frekvenca vzorčenja $f_s=1/T_s$)
- ▶ Vzorčene analogne vrednosti se pretvorijo v digitalno obliko. Vsaka digitalna vrednost je zapisana z n-biti.
- ▶ Pretvorba v digitalno obliko poteka s pomočjo primerjave vzorčene vrednosti z referenčno napetostjo (V_R).
- ▶ 1 LSB v digitalnem zapisu ustreza analogni napetosti $V_R/2^n$. Pri $V_R=1V$ in $n=10$ bitnem A/D pretvorniku je 1LSB enak 0.98mV.
- ▶ Vhodni analogni signal sme vsebovati le frekvenčne komponente med 0 in $f_s/2$, ker se pri A/D pretvorbi frekvence $f \geq f_s/2$ preslikajo v pas med 0Hz in $f_s/2$ (aliasing).
- ▶ Pred A/D pretvornikom imamo ponavadi analogen filter (anti-aliasing filter), ki ima zgornjo frekvenčno mejo manjšo od $f_s/2$.
- ▶ Končni rezultat A/D pretvorbe je digitalen signal - zaporednje n-bitnih vrednosti, ki ustrezajo vzorcem analognega signala ob $t=T_s$, $t=2T_s$, $t=3T_s$, ...

D/A pretvorba

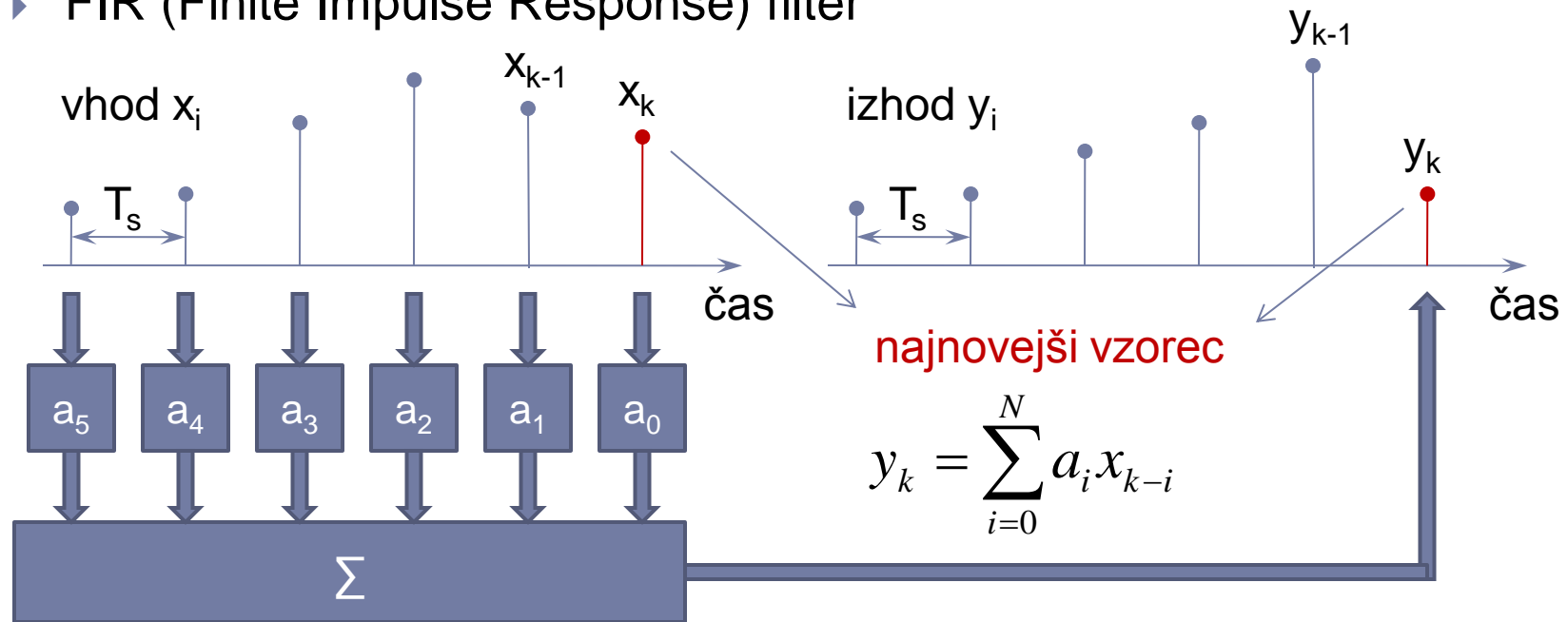
- ▶ D/A prevornik sprejema digitalni signal od procesorja in generira analogno napetost, ki je sorazmerna digitalnim vzorcem.
- ▶ Tudi D/A pretvorba poteka s pomočjo referenčne napetosti.
- ▶ Izhodna analogna napetost se spreminja ob trenutkih $t=T_s$, $t=2T_s$, $t=3T_s$, ...
- ▶ Rezultat je stopničast izhodni signal, ki ga zgladimo s pomočjo rekonstrukcijskega filtra (nizkopasovni filter z zgornjo mejno frekvenco manjšo od $f_s/2$)



Digitalni filtri

FIR filtri

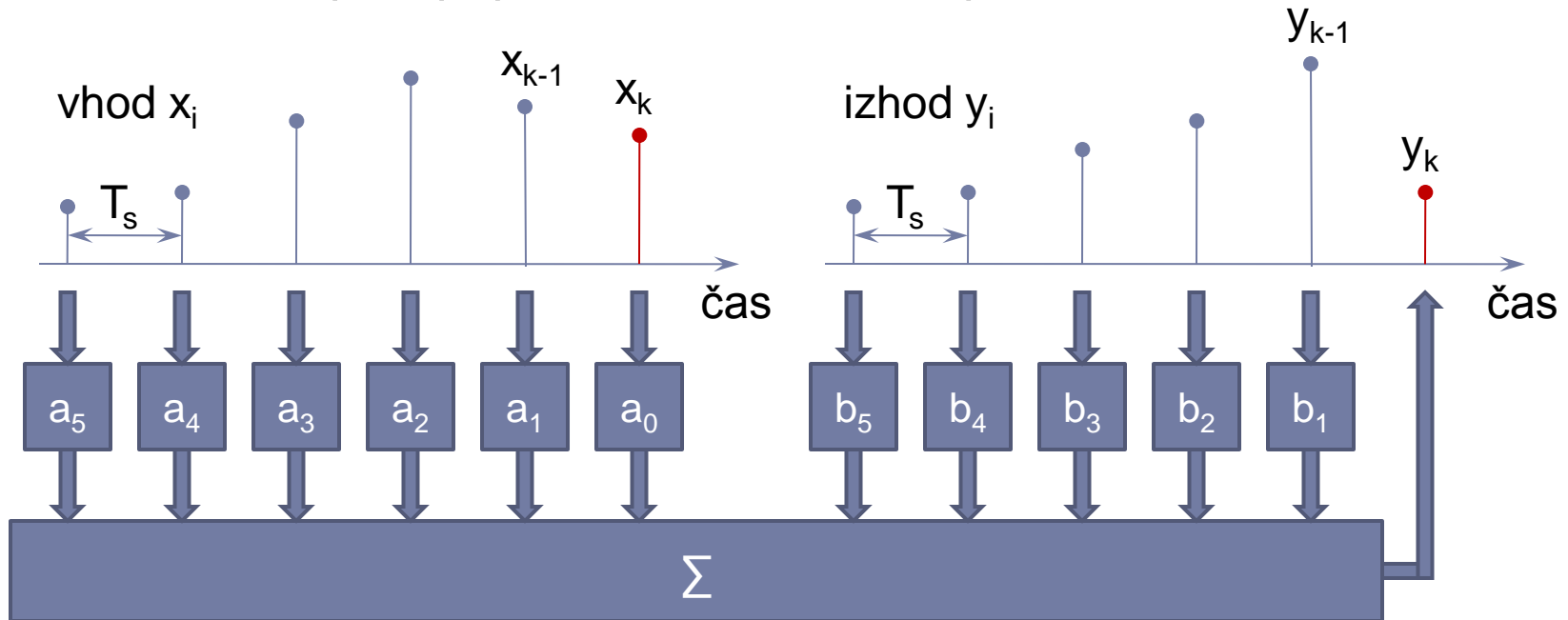
- ▶ Vhodnemu digitalnemu signalu spremenijo frekvenčno vsebino.
- ▶ FIR (Finite Impulse Response) filter



- ▶ Odziv na impulz je končno dolg (popolnoma zamre po končnem času).
- ▶ Karakteristiko filtra določajo koeficienti $a_i, i=0, 2, \dots, N$

IIR filtri

- ▶ Infinite Impulse Response (IIR)
- ▶ Sorodni analognim filtrom
- ▶ Odziv na impulz popolnoma zamre šele po neskončnem času.



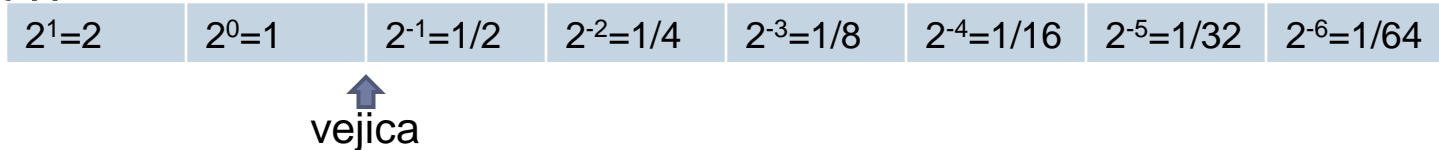
$$y_k = \sum_{i=0}^N a_i x_{k-i} + \sum_{i=1}^M b_i y_{k-i}$$

Pomen in izvedba digitalnih filtrov

- ▶ Operacije s signali, kot sta FFT in DCT (JPEG, MPEG) = FIR filtri.
- ▶ IIR filtri zahtevajo relativno malo operacij na sekundo (N in M sta majhna), vendar moramo poznati zgodovino tako vhodnega, kot izhodnega signala.
- ▶ Koeficienti a_i in b_i so konstante.
- ▶ Za izvedbo filtra je dovolj, da procesor zna seštevati in množiti.
- ▶ Da dobimo en izhodni vzorec FIR filtra potrebujemo N+1 množenj in N seštevanj (za IIR filter pa N+M+1 množenj in N+M seštevanj).
- ▶ Omenjene operacije moramo opraviti tekom ene periode vzorčenja.
Primer: FIR, N=50, $f_s=44100\text{Hz}$...
... 2 249 100 množenj/s in 2 205 000 seštevanj/s
1 operacija = 1 cikel ure ... potrebujemo procesor z min. 4.45MIPS
- ▶ Predvsem množenje je problem, ker njegova programska izvedba zahteva več urinih ciklov za eno samo množenje dveh števil.
- ▶ Namenska vezja za množenje.

Dvojiška števila s fiksno vejico

- ▶ Fiksno število dvojiških mest, fiksen položaj vejice.
- ▶ Nepredznačena: skupaj n mest, od tega r mest leži desno od vejice
Primer: $n=8$ mest, od tega 2 celi mesti (pred vejico) in $r=6$ mest za vejico



$$11.011011 = 2+1+1/4+1/8+1/32+1/64 = 3.421875$$

- ▶ Nepredznačena števila s fiksno vejico

Uteži: $2^{n-r-1} \dots 2^{-r}$

Območje: $0 \dots 2^{-r}(2^n - 1)$

Natančnost: 2^{-r}

Primer: $n=8, r=6 \dots$ območje $0 \dots 3.984375 = 0 \dots 4-1/64$, natančnost $1/64$

- ▶ Predznačena števila s fiksno vejico ... najbolj leva utež (MSB) je negativna

Uteži: $-2^{n-r-1}, 2^{n-r-2}, \dots 2^{-r}$

Območje: $-2^{-r} 2^{n-1} \dots 2^{-r} (2^{n-1} - 1)$

Natančnost: 2^{-r}

Primer: $n=8, r=6 \dots$ območje $-2 \dots 1.984375 = -2 \dots 2-1/64$, natančnost

▶ $1/64$

Dvojiška števila s plavajočo vejico

- ▶ Zapis zelo majhnih in zelo velikih števil ima enako relativno natančnost.
- ▶ Število zapišemo kot produkt predznaka, mantise in eksponentnega člena

▶ Primeri:

$$\begin{array}{lcl} -1110.10111100 & = & \underbrace{-1}_{\text{predznak}} \times \underbrace{1.110101111}_{\text{mantisa}} \times \underbrace{2^3}_{\text{eksponentni člen}} \\ 0.00000010111100 & = & \underbrace{+1}_{\text{predznak}} \times \underbrace{1.01111000000}_{\text{mantisa}} \times \underbrace{2^{-7}}_{\text{eksponentni člen}} \\ 1101111011.01 & = & \underbrace{+1}_{\text{predznak}} \times \underbrace{1.10111101101}_{\text{mantisa}} \times \underbrace{2^7}_{\text{eksponentni člen}} \end{array}$$

- ▶ Predznak je lahko le +1 ali -1 in ga zapišemo z enim bitom.
 - ▶ Mantisa je zapisana v zapisu s fiksno vejico.
 - ▶ Eksponentni člen je oblike 2^k , kjer je k pozitivno ali negativno število.
 - ▶ Množenje je enostavno. Posebej množimo predznak in obe mantisi, eksponente obeh eksponentnih členov pa seštejemo.
 - ▶ Pri seštevanju najprej poravnamo obe mantisi tako, da oba eksponentna člena postaneta enaka, nakar sledi
-
- ▶ 9 seštevanje/odštevanje mantis (predznak!).

IEEE 754 števila s plavajočo vejico...

- ▶ Standarden format zapisa. Uporablja se za tipa float in double v jeziku C.
- ▶ Pozna 16, 32, 64 in 128 bitne dvojiške in desetiške zapise. Tipa float in double sta v 32 in 64 bitnem dvojiškem zapisu IEEE 754.
- ▶ **32-bitni dvojiški format (float) – biti od MSB proti LSB:**
 - 1 bit za predznak (0=+, 1=-)
 - 8 bitov za eksponent (e), tolmačimo jih kot nepredznačeno število
 - 23 bitov za ulomljeni del (f) mantise
- ▶ **+0 in -0 ... e=000..0, f=000.0**
(float) -0.0 = 1 00000000 000000000000000000000000
- ▶ **+/- neskončno (Inf) ... e=111..1, f=000.0**
1/0=+Inf
(float) +inf = 0 11111111 000000000000000000000000
- ▶ **NaN (not a number) ... e=111..1, f≠000..0**
0/0=NaN, 0*Inf=NaN, Inf-Inf=NaN
(float)NaN = 0 11111111 100000000000000000000000

... IEEE 754

▶ Normalizirana števila ... e≠000..0

Sem spada večina števil.

mantisa = 1.f (nepredznačena)

k = e-127

Primer:

1 01111111 10000000000000000000000000000000

predznak = -1

k=01111111₍₂₎-127=127-127=0

mantisa=1.10000000000000000000000000000000₍₂₎=1.5

vrednost = -1 x 1.5 x 2⁰ = -1.5

▶ Denormalizirana števila ... e=000..0, f≠000..0

Števila, ki so bližje 0, kot absolutno najmanjši normalizirani števili.

Dobimo jih kot rezultat odštevanja dveh zelo majhnih, a skoraj enakih števil.

mantisa = 0.f (nepredznačena)

k=-126

Primer:

0 00000000 11000000000000000000000000000000

predznak = +1

k=-126

mantisa = 0.11000000000000000000000000000000₍₂₎=0.75

vrednost = +1 x 0.75 x 2⁻¹²⁶ = 8.8162076 x 10⁻³⁹



Natančnost IEEE 754 zapisa

- ▶ float (32-biten binarni) zapis

denormalizirana števila: $\pm 2^{-149} \dots \pm (1 - 2^{-23}) \times 2^{-126}$

normalizirana števila: $\pm 2^{-126} \dots \pm (2 - 2^{-23}) \times 2^{127}$

relativna natančnost normaliziranih števil

2^{-24} oziroma 5.96×10^{-8}

7 desetiških mest

- ▶ double (64-biten binarni) zapis

denormalizirana števila: $\pm 2^{-1074} \dots \pm (1 - 2^{-52}) \times 2^{-1022}$

normalizirana števila: $\pm 2^{-1022} \dots \pm (2 - 2^{-52}) \times 2^{1023}$

relativna natančnost normaliziranih števil

2^{-53} oziroma 1.11×10^{-16}

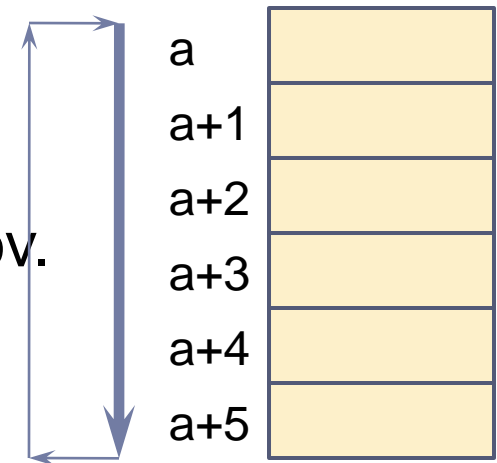
15 desetiških mest

Signalni procesor (SP)

- ▶ V bistvu čisto navaden mikroprocesor
- ▶ Specializiran za digitalno obdelavo signalov
- ▶ Pogosto se uporablja (modificirana) Harvardska arhitektura
- ▶ Lahko ima več vodil do posameznih pomnilnikov
- ▶ Omogoča več dostopov do pomnilnika v enem urinem ciklu
- ▶ Vgrajen množilnik
- ▶ Specializirani ukazi
 - MAC (Multiply-ACcumulate) operacije: $A=A+B*C$
 - naslavljanje krožnih vmesnikov
 - učinkovito izvajanje kratkih zank
 - naslavljanje z obrnjenimi biti (bit-reversed addressing)
 - ...
- ▶ Brez MMU
 - MMU vnaša nepredvidljive zakasnitve pri dostopu do pomnilnika

Krožni vmesnik (Circular Buffer)

- ▶ Primer: krožni vmesnik dolžine $LR=6$
- ▶ Trenutni položaj v vmesniku določa indeksni register.
- ▶ Ko indeksni register (IR) povečamo za 1, se pomakne za en naslov naprej.
Če je $IR=a+3$, je po povečanju IR enak $a+4$.
Če je $IR=a+5$, je po povečanju IR enak a ... krožni vmesnik.
- ▶ Za hranjenje preteklih vzorcev vhodnega in izhodnega signala.
- ▶ Za hranjenje koeficientov FIR in IIR filtrov.



Primeri operacij

▶ Naslavljanje krožnih vmesnikov

Primer: Analog Devices ADSP21xx

Naloži konstanto iz programskega pomnilnika z naslova I0 v register MX0.

Krožno (z dolžino L0) povečaj I0 za M1.

```
MX0=DM(I0, M1)
```

▶ Kratke zanke

Primer: ADSP21xx

10 iteracij. 1 cikel ure = 1 iteracija zanke. Preverjanje pogoja = 0 ciklov ure.

Ena vrstica je en ukaz (ukazi ADSP2181 so 24-bitni).

```
CNTR=10
```

```
DO sop UNTIL CE;
```

```
sop: MR=MR+MX0*MY0(SS), MX0=DM(I0, M1), MY0=PM(I4, M5);
```

▶ MAC operacije – osnova digitalnih filtrov.

Izvajata jih množilnik in seštevalnik.

Ponavadi ena MAC operacija na urin cikel.

Primer: ADSP21xx

Množi 16-bitni predznačeni vsebini MX0 in MY0 ter ju prištej k MR.

```
MR=MR+MX0*MY0(SS)
```

MAC operacije

- ▶ Manjši signalni procesorji – množilnik za števila s fiksno vejico
Primer: ADSP21xx – 16-bitna števila s fiksno vejico formata 1.15 ($n=16$, $r=15$)
- ▶ Večji signalni procesorji – množilnik za števila s plavajočo vejico
Primer: ADSP-TS201S – poleg 32-bitnega množilnika s fiksno vejico
 - 32-bitni množilnik s plavajočo vejico (IEEE754, 23-bitni ulomljeni del) in
 - 40-bitni množilnik (kot 32-bitni IEEE754, le da je ulomljeni del 31-biten)
- ▶ MAC operacije s fiksno vejico lahko privedejo akumulator do prenosa/preliva. Zato je akumulator daljši od vsote dolžin vhodnih podatkov množilnika.
Primer: ADSP21xx - 16-biten množilnik, 40-biten akumulator
- ▶ Če pride do preliva, nasičimo akumulator.
(nastavimo ga na najvišjo/najnižjo možno vrednost).
Če tega ne naredimo, dobimo velike skoke v signalu (šum).
Posledica nasičenja je porezan (nelinearno popačen) signal.
- ▶ Množenje dveh kompleksnih števil = 4 množenja in 2 seštevanji.

Druge operacije v signalnih procesorjih

- ▶ Manipulacije bitov

Za branje/spreminjanje poljubnega dela besede pri kodiranju/dekodiranju avdio/video tokov.

- ▶ Naslavljanje z obrnjenimi biti – pri hitri Fourier-jevi transformaciji

(Fast Fourier Transformation, FFT)

Če je naslov $01001010_{(2)}=74$, vzamemo podatek z naslova $01010010_{(2)}=82$

- ▶ Izračun posebnih funkcij, kot naprimer funkcija $\max(a, b) + \ln(1 + e^{-|a-b|})$

Uporaba – dekodiranje tki. turbo kod

- ▶ Pomožne funkcije za dekodiranje na osnovi mrežnih (trellis) diagramov

- ▶ Izračun kompleksne korelacije (uporaba v CDMA sistemih)

Posebnosti ukazov signalnih procesorjev

▶ Primer ukaza – vse v enem urinem ciklu

ADSP21xx – en 24-biten ukaz

V enem ciklu ure zmnoži vsebini MX0 in MY0 in prišteje rezultat k MR, naloži novo vrednost v MX0 z naslova I0 v podatkovnem pomnilniku, krožno poveča I0 za M1, naloži novo vrednost v MY0 z naslova I4 v programskem pomnilniku, krožno poveča I4 za M5.

$$MR=MR+MX0*MY0(SS), \quad MX0=DM(I0, M1), \quad MY0=PM(I4, M5);$$

▶ VLIW ukazi (Very Long Instruction Word)

izvedejo več operacij vzporedno v enem urinem ciklu

ADSP-TS0xx (TigerSHARC) – en 32-biten ukaz sestavljen iz največ 4 podukazov
Podukazi se izvršijo vzporedno v enem ciklu ure.

▶ VLIW je del družine MIMD (Multiple Instruction Multiple Data)

▶ Družina SIMD (Single Instruction Multiple Data)

Primer: MMX ukazi v procesorjih Intel

Vzporedno opravijo eno operacijo na 8 podatkih.

▶ Meja med signalnimi procesorji in mikroprocesorji postaja

▶ 18 zabrisana

Analog Devices BlackFin družina – mikrokrmilnik + DSP