Arhitekture i algoritmi DSP 2

Auditorne vežbe AU-0[0]
Pregled kursa i priprema

Odsek za računarsku tehniku i računarske komunikacije

Pregled kursa

- Platforma: Canvas FTN, kurs AADSP2 https://canvas.ftn.uns.ac.rs/courses/119
- Auditorne vežbe (priprema za lab. vežbe)
- Laboratorijske vežbe (priprema za finalni projekat)
- Predispozicije:
  - vladanje C jezikom (i delimično C++)
  - razumevanje osnova digitalne obrade signala
  - rad u okruženjima Visual Studio i Eclipse
  - podsetnici:
    - c_podsetnik.pdf
    - Osnovne_komponente_digitalne_obrade_signala.pdf

Pregled bodova

- Predispit **60+ bodova** (potrebno 31 bod za prolaz):
  - Predispit PE x.0: Testovi u sklopu auditornih vežbi **3 boda**
  - Predispit PE x.1: Aktivnost u sklopu lab. vežbi **6 bodova**
  - Predispit PE x.2: Predispitni zadatak **6 bodova** (2b prolaz)
  - *Predispit **PE x.3**: Finalni projekat **45 bodova** (22b prolaz)*
  - Predispit PE x.4*: Bonus testovi tokom predavanja (5-7b)
- Ispit **40 bodova** (potrebno 20 bodova za prolaz).
  - Teorijski ispit, online, Canvas FTN.

*x – redni broj modula/dana

Raspored vežbi i aktivnosti

Dan	Auditorna priprema	Laboratorijska vežba	Predispitne obaveze
#0 – Pre 07.11.	AU-0		
#1 – 07.11.	AU-1	VE-1	PE 1.0 PE 1.1
#2 – 08.11.	AU-2	VE-2	PE 2.0 PE 2.1
#3 – 09.11.		VE-3	PE 3.1
#4 – 10.11.	AU-4	VE-4	PE 4.0 PE 4.1 **PE 4.2**
#5 – 14.11.	AU-5	VE-5	PE 5.0 PE 5.1
#6 – 15.11.	AU-6	VE-6	PE 6.0 PE 6.1
#7 – 16.11.	AU-7	VE-7	PE 7.0 PE 7.1
#8 – 17.11.	AU-8	VE-8	PE 8.0 PE 8.1
#9 – 18.11.		VE-9	PE 9.1 **PE 9.3***
#10 – 21.11.	AU-A	VE-A	PE A.1

Raspored vežbi i aktivnosti

Dan	Auditorna priprema	Laboratorijska vežba	Aktivnost
#11 – 22.11.			PE 11.3*
#12 – 23.11.			PE 12.3*
#13 – 24.11.			PE 13.3*
#14 – 25.11.			PE 14.3*
#15 – 28.11.			PE 15.3*
#16 – 29.11.			PE 16.3*
#17 – 30.11.			PE 17.3*
#18 – 01.12. i 02.12			**PE 18.3** (odbrana projekata)

*PE x.3 – samostalni rad na izradi projektnih zadataka i konsultacije.

AADSP2 nastavno osoblje

- Predavanja:
  - Doc. dr Jelena Kovačević: jelena.kovacevic@rt-rk.com
- Vežbe:
  - Azra Samac: azra.samac@rt-rk.com
  - Andrej Popović: andrej.popovic@rt-rk.com
  - Nenad Pekez: nenad.pekez@rt-rk.com

Arhitekture i algoritmi
DSP 2

Auditorne vežbe AU-0[1]
Pregled komercijalnih uređaja i top-dijagram

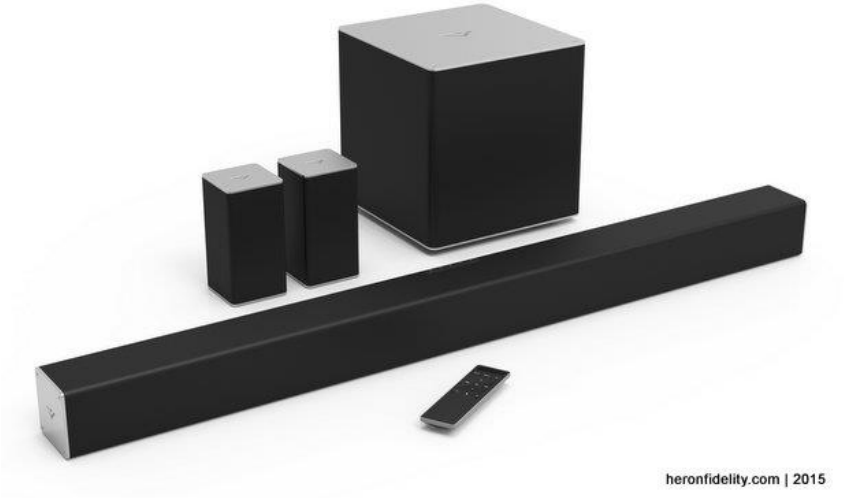
Odsek za računarsku tehniku i računarske komunikacije

Cirrus Logic DSP na tržištu

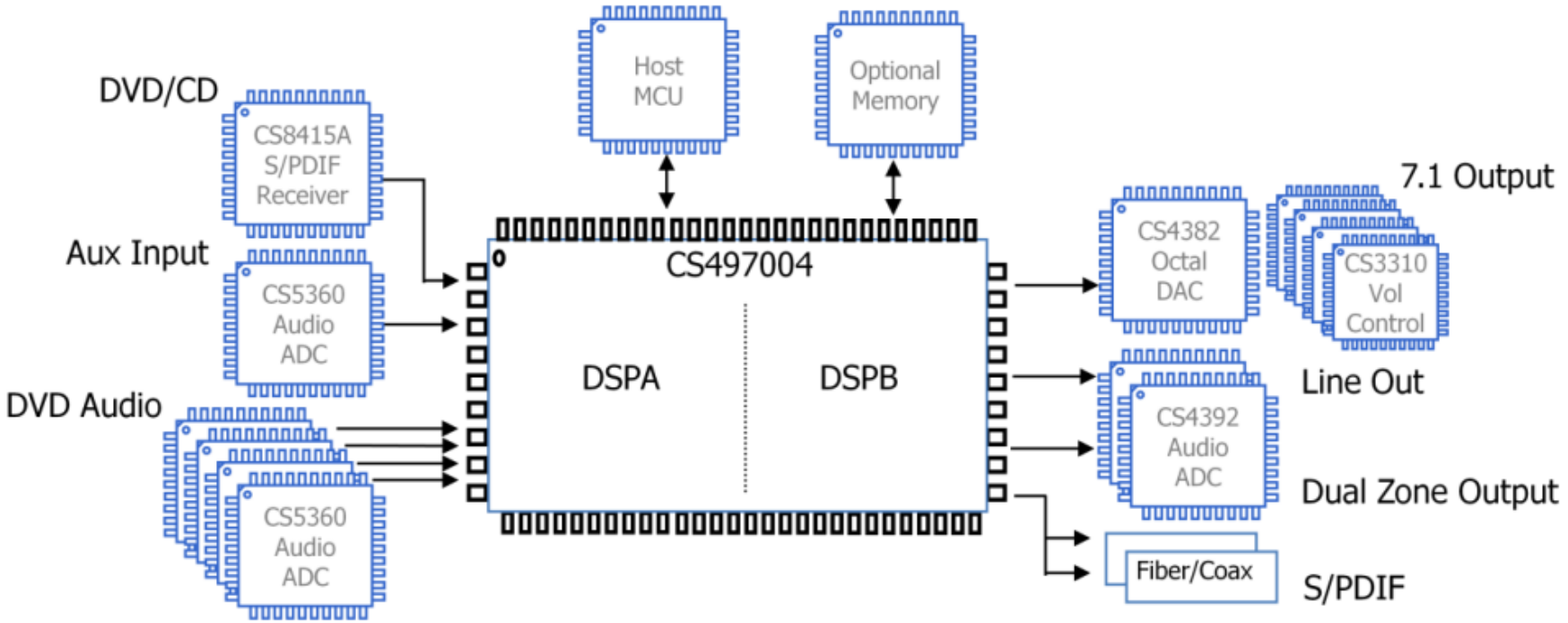
- Cirrus Logic DSP CS497xx, CS498xx
- AVR

(pojačalo)

- Saundbar



CS497004 DSP u uređaju



Slika 2.1 – CS497004 DSP u AVR uređaju

Firmverski moduli u CS497xx

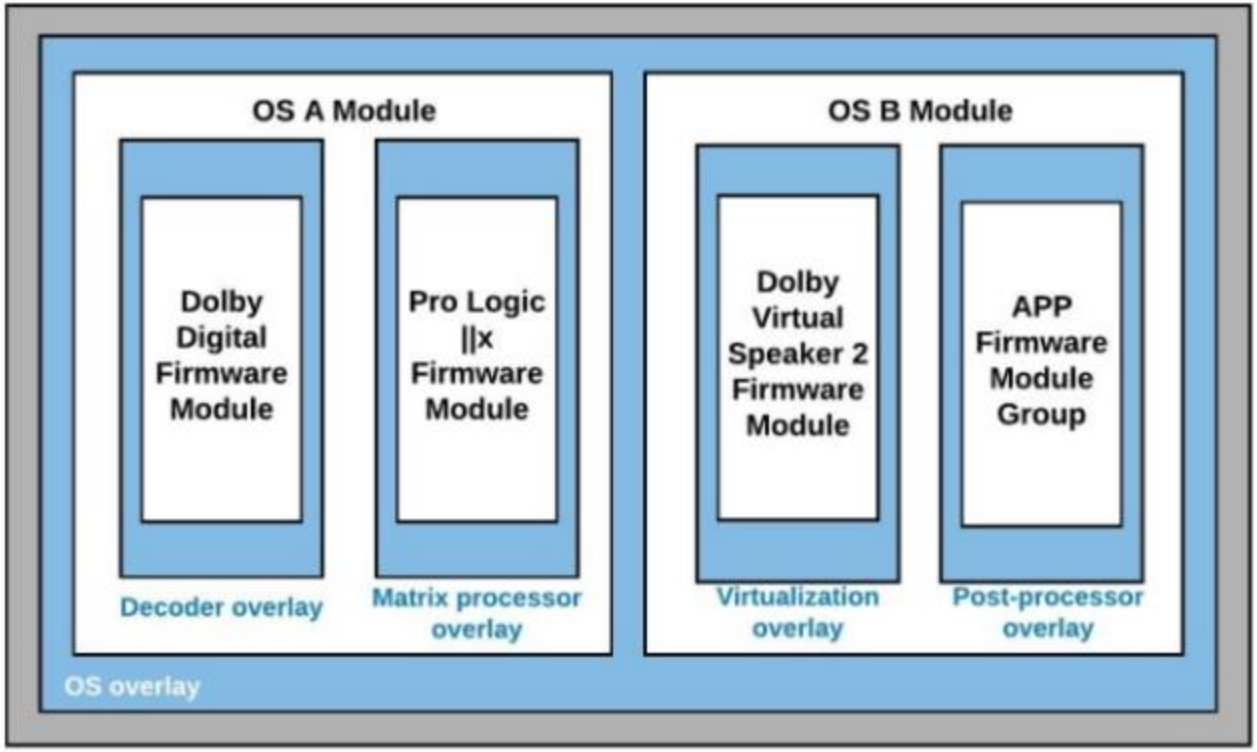


Fig. 2. DSP application example [8]

Arhitekture i algoritmi DSP 2

Auditorne vežbe AU-0[2]
ARM FIR vs CS FIR

Odsek za računarsku tehniku i računarske komunikacije

Zašto CS497xx DSP?

- Specijalizovana namenska hardverska arhitektura za DSP optimizacije!

FIR filter C

```c
/* calc FIR and shift data */
ret_val = 0;
for (i = n_coeff - 1; i >= 0; i--)
{
    ret_val += coeffs[i] * history[state];
    if (++state >= n_coeff) /* incr state and check for wrap */
    {
        state = 0;
    }
}

*p_state = state;                /* return new state to caller */
```

```
# for k=1,..N-1
#   out[n]+=h[k]*in[n-k]
```

FIR filter ARM Cortex-M3

```
# C declaration:  int64_t fir64(const int32_t*
a,
#       int32_t* x, uint32_t n);
# Parameters:
#     r0: coef ptr
#     r1: input ptr
#     r2: filter length (must be 4, 8,
#                        a multiple of 4)

fir64:
    push {r4-r11}

    mov r12, r2

    mov r3, #0
    mov r2, #0

fir64_loop_start:
    ldmia r0!, {r4-r7}
    ldmia r1!, {r8-r11}

    smlal r2, r3, r4, r8
    smlal r2, r3, r5, r9
    smlal r2, r3, r6, r10
    smlal r2, r3, r7, r11

    subs r12, r12, #4
    bgt fir64_loop_start

    mov r0, r2
    mov r1, r3

    pop {r4-r11}

    bx lr
```

FIR filter CS497xx

```
 do (i4),>block_loop
                                                                  i4=i6
                                                                  x0=ymem[i2]; i2+=n
    a1 =+ x0;                                                     x1=ymem[i2]; i2+=n
    b1 =+ x1;                                                     y0=ymem[i4]; i4+=1
    a0=y0*x0;    b0=y0*x1;    x1=xmem[i0]; i0-=1;                 y0=ymem[i4]; i4+=1
    do (i1),>
      a0+=y0*x1;  b0+=y0*x0;  x0=xmem[i0]; i0-=1;   y0=ymem[i4]; i4+=1
%:    a0+=y0*x0;  b0+=y0*x1;  x1=xmem[i0]; i0-=1;   y0=ymem[i4]; i4+=1
    a0+=y0*x1;  b0+=y0*x0;                                        i0+=n
    b1 =+ a0;    a0 = b1
                                xmem[i0]=a1; i0+=1; ymem[i5]=b1; i5+=n
%block_loop:                    xmem[i0]=a0;        ymem[i5]=b0; i5+=n
                                                    xmem[i3]=i0
```

Broj instrukcija po bloku

ARM Cortex-M3

- 8 instructions per tap/coeff
- For taps = 16, 16 * 8 = 128 instructions per sample
- For samples = 32, 32 * 128 = **4096 instructions per block**

CS497xx

- 2 instructions per tap + 2 for pre- and post-fill
- For taps = 16, 16 * 2 + 2 = 34 instructions per sample
- For samples = 32, 32 * 34 + 8 = **1096 instructions per block**

MIPS Profiling

- MIPS – Millions of instructions per second
- Formula za računanje MIPS-a:
  - $\text{MIPS} = \frac{\left(\frac{Fs}{Bs}\right) * N}{10\text{^}6}$
  - Fs – frekvencija odabiranja (44k1, 48k…)
  - Bs – broj semplova po bloku obrade
  - N – broj instrukcija po bloku obrade
  - $\text{MIPS} = \frac{\left(\begin{matrix}broj\ poziva\ bloka\ obrade \\ po\ sekundi\end{matrix}\right) * (broj\ instrukcja\ po\ bloku\ obrade)}{10\text{^}6}$

MIPS [ARM M3 vs CS497xx]

ARM Cortex-M3

- For Fs = 44100 1/s, Bs = 32, N = 4096
- $MIPS = \frac{\left(\frac{44100}{32}\right) * 4096}{10\text{^}6}$
- MIPS = **5.64**

CS497xx

- For Fs = 44100 1/s, Bs = 32, N = 1096
- $MIPS = \frac{\left(\frac{44100}{32}\right) * 1096}{10\text{^}6}$
- MIPS = **1.51**