



УНИВЕРЗИТЕТ  
У НОВОМ САДУ



ФАКУЛТЕТ  
ТЕХНИЧКИХ НАУКА

Трг Доситеја Обрадовића 6, 21000 Нови Сад, Југославија  
Деканат: 021 350-413; 021 450-810; Централa: 021 350-122  
Рачуноводство: 021 58-220; Студентска служба: 021 350-763  
Телефакс: 021 58-133; e-mail: [ftndeans@uns.ns.ac.yu](mailto:ftndeans@uns.ns.ac.yu)



Сертификован  
систем  
квалитета



## PROJEKAT

### iz Računarskog projektovanja digitalnih integrisanih kola

#### TEMA PROJEKTA:

Hardverska implementacija FIR filtra

#### TEKST ZADATKA:

1. U programskom jeziku VHDL napisati kod kojim se realizuje FIR filter
2. Proveriti funkcionalnost realizovane funkcije.
3. Koristeći alat *Genus Synthesis Solution* programskog paketa *Cadence* generisati električnu šemu u AMS 0.35µm (C35B4) CMOS tehnologiji.
4. Koristeći alat *Innovus Implementation System* programskog paketa *Cadence* generisati fizičku realizaciju kola u AMS 0.35µm (C35B4) CMOS tehnologiji.

Mentor  
Kristina Nikolić

Student  
Čejić Ivan EE17/2019  
Milin Ivan EE179/2019

U Novom Sadu, 03.04.2023.

## 1. Teorijska analiza

Digitalni filtri predstavljaju frekvencijski selektivne sisteme, oni zapravo vrše diskriminaciju signala prema frekvenciji, osnovna funkcija digitalnog filtera je da izvrše uobličavanje spektra ulaznog signala.

U velikom broju slučajeva se to uobličavanje signala svodi na propuštanje signala koji leže u nekom opsegu učestanosti koji se naziva propusni opseg i na slabljenje signala koji leže van tog opsega, odnosno u nepropusnom opsegu.

Sledeći MATLAB kod ilustruje postupak projektovanja FIR filtra :

```
%broj bita odbirka (format je 1.23)
word_length = 24;
fraction_length = 23;
fs = 22050;
f1 = 400;
f2 = 4000;

%specifikacija NF filtra
fir_ord = 4;
Wn=[0.1];

%odbirci prozorske funkcije koja se koristi
pravougaoni = rectwin(fir_ord+1);

%projektovanje FIR filtera koriscenjem funkcije fir1
b = fir1 (fir_ord, Wn, pravougaoni);
a = 1;

%diskretno vreme
n = 0:149;

%definisane ulaznog signala u trajanju od 150 odbiraka
u = 0.85*cos(2*pi*f1/fs*n) + 0.2*cos(2*pi*f2/fs*n);

%filtriranje signala pomocu formiranog filtra
y_real = filter(b,a,u);

|struct.mode = 'fixed';
struct.roundmode = 'floor';
struct.overflowmode = 'saturate';
struct.format = [word_length fraction_length];
q = quantizer(struct);

%digitalizacija diskretnog signala
u_digital = quantize(q,u);

%koeficijenti filtra
fileIDh = fopen('coef_hex.txt','w');
for i=1:fir_ord+1
    fprintf(fileIDh,'x"');
    fprintf(fileIDh,num2hex(q,b(i)));
    fprintf(fileIDh,'"','\n');
end
fclose(fileIDh);

fileIDb = fopen('input.txt','w');
for i=1:length(u_digital)
    fprintf(fileIDb,num2bin(q,u(i)));
    fprintf(fileIDb,'\n');
end
fclose(fileIDb);
```

$$u = 0.85 * \cos\left(2\pi \frac{400}{22050} n\right) + 0.2 * \cos\left(2\pi \frac{4000}{22050} n\right)$$

1

Većina signala koje srećemo u prirodi je kontinualna, kako bismo mogli isti taj signal da obradimo kroz digitalni filter neophodno je da izvršimo neke modifikacije tog kontinualnog signala, to su diskretizacija po amplitudi (kvantovanje) i diskretizacija po vremenu (odabiranje), što se skraćeno zove analogno-digitalna (A/D) konverzija.

Nakon kvantovanja dobijamo diskretni signal, a nakon odabiranja dobijamo digitalni signal pogodan za dalju obradu.

Postupak kvantovanja ima za cilj da od kontinualnih amplituda dobijemo skup konačnih amplituda, neprebrojiv skup amplitudskih vrednosti mora se pretvoriti u prebrojiv skup, u suprotnom ne bi se moglo izvršiti kodiranje.

Postupak odabiranja dovodi do degradacije kontinualnog signala, jer obrađujemo signal samo u određenim vremenskim trenucima.

Kako bismo očuvali informaciju prvobitnog kontinualnog signala neophodno je da se pridržavamo teoreme odabiranja koja kaže da ukoliko želimo da prikazemo sve komponente nekog signala neophodno je da frekvencija odabiranja bude bar dva puta veća od najveće frekvencije koju ima neka komponenta polaznog kontinualnog signala.

Na ulaz filtra dovodimo signal iz jednačine 1, filter je NF tipa čija je frekvencija odabiranja postavljena na 11025 Hz (fs/2).

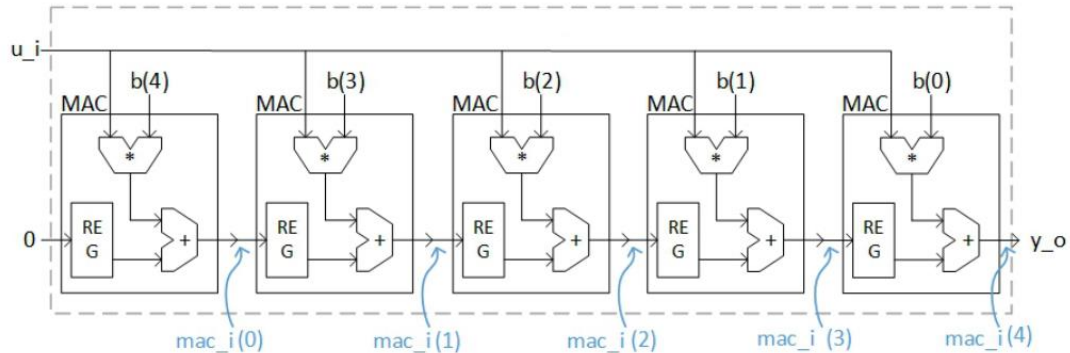
U MATLAB kodu smo naznačili da filter ne propušta frekvencije koje su veće 10% od frekvencije odabiranja, znači da komponenta signala koja premašuje 1100 Hz neće biti propuštena kroz filter.

Očekujemo da će biti propuštena samo komponenta od 400Hz, dok komponenta od 4000 Hz neće biti propuštena.

Cilj MATLAB koda je da na osnovu željene specifikacije generišemo koeficijente filtra, test signal i koeficijenti filtra biće sačuvani u dva tekstualna fajla.

## 2. Projektovanje digitalnog integrisanog kola

Blok šema FIR filtra prikazana na slici 1.



Slika 1. Blok šema FIR filtra

Kao što se može videti na slici 1. FIR filter sačinjen je od 5 MAC modula, a svaki MAC modul sastoji se je od registra, množača i sabirača.

U nastavku biće priloženi dizajn fajlovi :

### mac.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity mac is
    Port ( clk_i : in STD_LOGIC;
          u_i : in STD_LOGIC_VECTOR (23 downto 0);
          b_i : in STD_LOGIC_VECTOR (23 downto 0);
          mac_i : in STD_LOGIC_VECTOR (47 downto 0);
          mac_o : out STD_LOGIC_VECTOR (47 downto 0));
end mac;

architecture Behavioral of mac is
    signal reg_s : STD_LOGIC_VECTOR(47 downto 0) :=(others=>'0');
begin
    process(clk_i)
    begin
        if (clk_i'event and clk_i = '1') then
            reg_s <= mac_i;
        end if;
    end process;

    mac_o <= std_logic_vector(signed(reg_s) + (signed(u_i) * signed(b_i)));
end Behavioral;
```

MAC modul sastoji se od sledećih portova:

- clk\_i – ulazni port preko koga se dovodi taktni signal
- u\_i – ulazni port preko koga se dovodni signal koji treba filtrirati
- b\_i – ulazni port preko koga se dovode koeficijenti filtra
- mac\_i – ulazni port mac modula
- mac\_o – izlazni port mac modula

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

```

entity fixed_fir is
    Port ( clk_i : in STD_LOGIC;
          u_i : in STD_LOGIC_VECTOR (23 downto 0);
          y_o : out STD_LOGIC_VECTOR (23 downto 0));
end fixed_fir;

architecture Behavioral of fixed_fir is
    type std_2d is array (4 downto 0) of std_logic_vector(47 downto 0);
    signal mac_inter : std_2d := (others=>(others=>'0'));

    type coef_t is array (0 to 4) of std_logic_vector(23 downto 0);
    signal b : coef_t := ("000110001011111111001011",
                          "000110100000010111011001",
                          "000110100111010010110101",
                          "000110100000010111011001",
                          "000110001011111111001011");

    signal constant_signal:std_logic_vector(47 downto 0) :="000000000000000000000000000000000000000000";

    component mac
        port( clk_i : in STD_LOGIC;
              u_i : in STD_LOGIC_VECTOR (23 downto 0);
              b_i : in STD_LOGIC_VECTOR (23 downto 0);
              mac_i : in STD_LOGIC_VECTOR (47 downto 0);
              mac_o : out STD_LOGIC_VECTOR (47 downto 0));
    end component;

begin

    prvi_MAC: mac
        port map(clk_i=>clk_i,
                 u_i=>u_i,
                 b_i=>b(4),
                 mac_i=> constant_signal,
                 mac_o=>mac_inter(0));

    drugi_MAC: mac
        port map(clk_i=>clk_i,
                 u_i=>u_i,
                 b_i=>b(3),
                 mac_i=>mac_inter(0),
                 mac_o=>mac_inter(1));

    treci_MAC: mac
        port map(clk_i=>clk_i,
                 u_i=>u_i,
                 b_i=>b(2),
                 mac_i=>mac_inter(1),
                 mac_o=>mac_inter(2));

    cetvrti_MAC: mac
        port map(clk_i=>clk_i,
                 u_i=>u_i,
                 b_i=>b(1),
                 mac_i=>mac_inter(2),
                 mac_o=>mac_inter(3));

    peti_MAC: mac
        port map(clk_i=>clk_i,
                 u_i=>u_i,
                 b_i=>b(0),
                 mac_i=>mac_inter(3),
                 mac_o=>mac_inter(4));

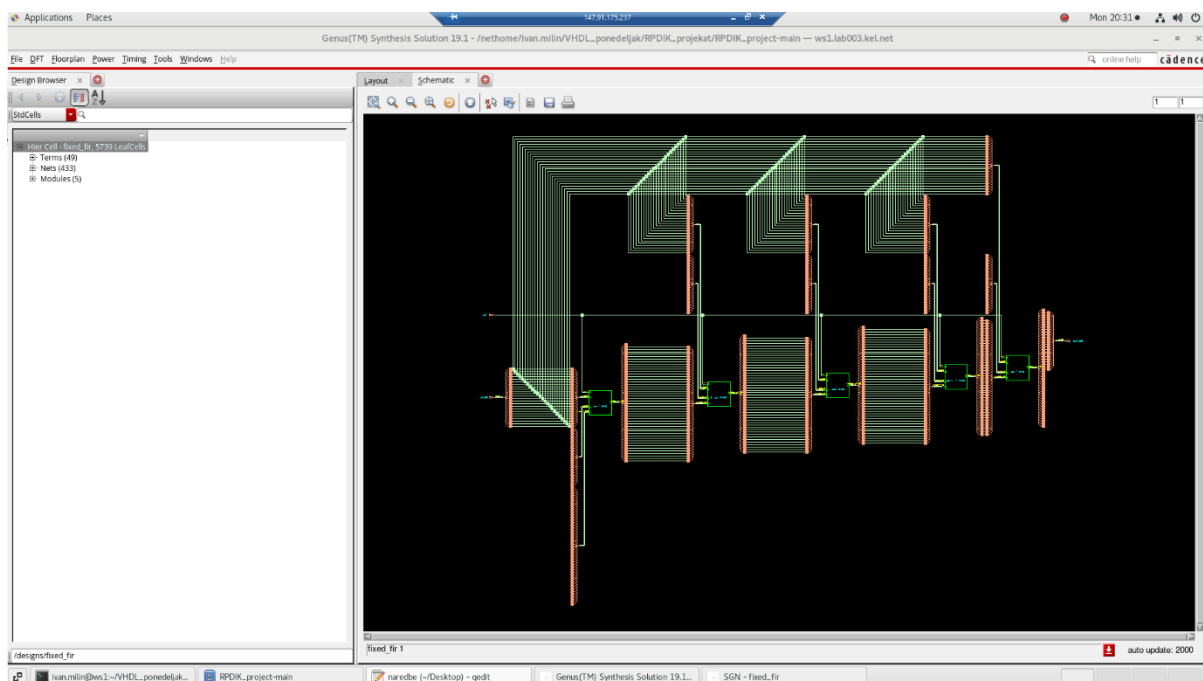
    y_o <= mac_inter(4)(46 downto 23);

end Behavioral;
```

Filtar ima sledeće portove:

- clk\_i – ulazni port preko koga se dovodi takti signal
- u\_i – ulazni port preko koga se dovodni signal koji treba filtrirati
- y\_i – izlazni port preko koga dobijamo modifikovani signal sa ulaza

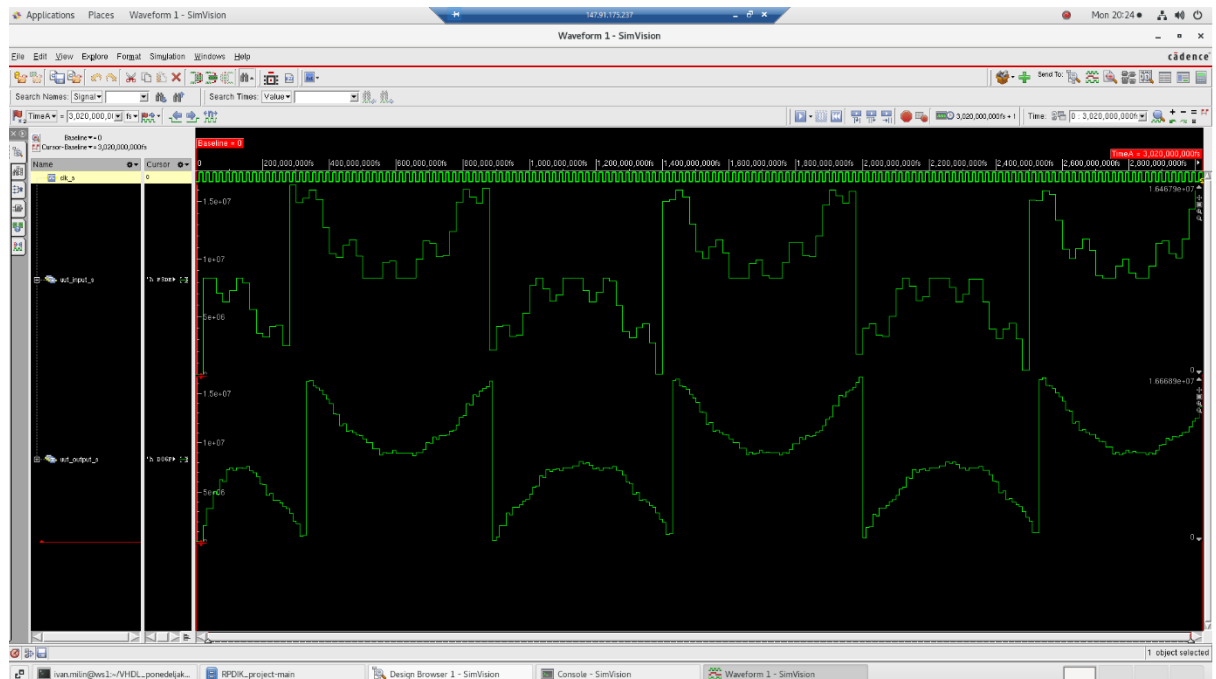
Šematik FIR filtra prikazan je na slici 2.



Slika 2. Šematik FIR filtra

### 3. Simulaciona provera rezultata

Simulacioni rezultati prikazani na slici 3.



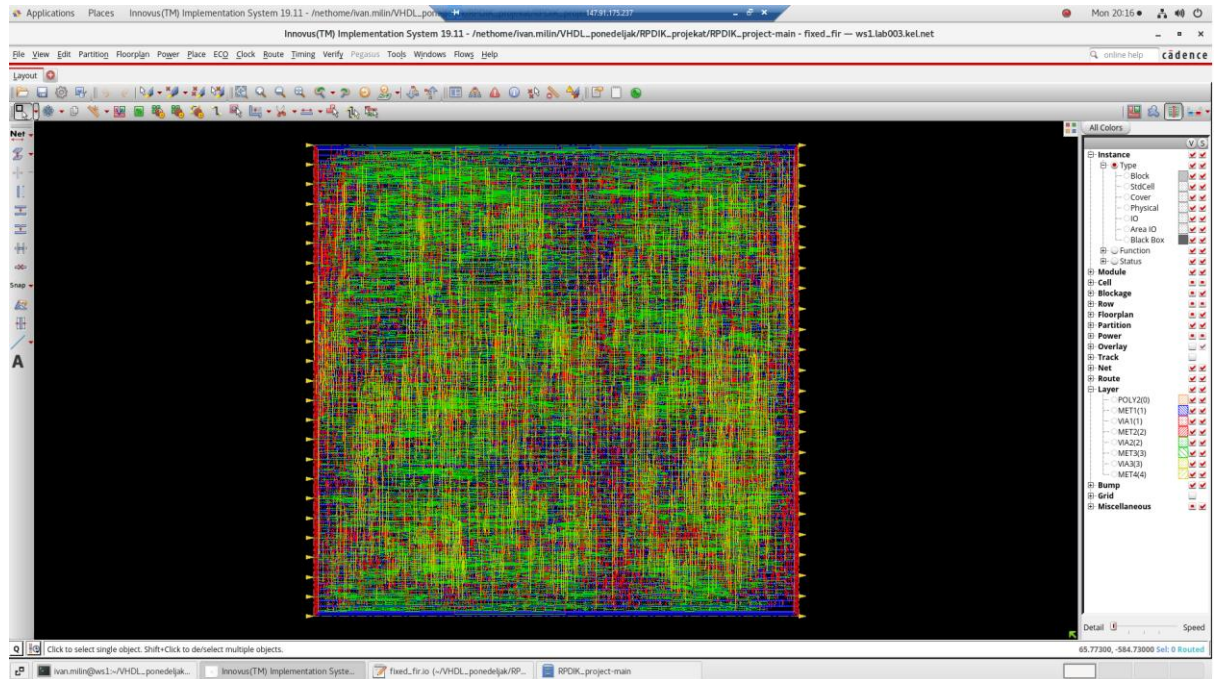
Slika 3. Simulacioni rezultati

Kao što možemo videti na slici 3. ulazni signal *uut\_input\_s* ima dve komponente, u formuli 1 „sporiju” komponentu predstavlja ona od 400 Hz, dok je „brža” ona od 4000 Hz.

Pošto je filter NP tipa, izlazni signal *uut\_output\_s* čini isključivo „spora” komponenta što je i očekivano.

#### 4. Lejaut projektovanog kola

Na slici 4. prikazan je lejaut FIR filtra.



Slika 4. Lejaut FIR filtra



## **5. Diskusija dobijenih rezultata**

Nakon vizuelne provere signala prikazanih tokom simulacije možemo zaključiti da kolo izvršava željene zahteve.

„Brža” komponenta ulaznog signala je potpuno potisnuta projektovanim FIR filtrom.