

Unelte de modelare/simulare

Proiectarea, îmbunătățirea și diversificarea echipamentelor radio definite prin program sau virtuale are un puternic și eficient suport în multitudinea de unelte *hard* și *soft* special concepute și larg accesibile, în bună parte în mod gratuit. Inventarul acestor unelte este în continuă diversificare, o prezentare exhaustivă ne fiind ușor de făcut și mai ales de actualizat. Vor fi prezentate în continuare numai câteva exemple care să acopere prin utilitate mare parte a problemelor necesar a fi analizate și soluționate în funcționalitatea blocurilor aflate în structurile emițătoarelor și receptoarelor numerice.

*

Fiind procesul cel mai important din funcționarea echipamentelor radio definite prin program sau virtuale, conversia analog-numerică și numeric-analogică a fost de al început în atenția producătorii de asemenea dispozitive convertoare, aceștia oferind un susținut și puternic suport dezvoltatorilor de echipamente.

Astfel, Analog Devices (www.analog.com) oferă acces on-line la o variantă a programului **ADIsimADC™** ([4]), exemplificată prin figurile 1 - 3, care dispune de biblioteca modelelor comportamentale ale convertoarelor analog-numerice de mare viteză pe care le produce (8 – 16 biți, 40 – 250 MHz) permițând analiza comportării acestora la unu sau două semnale sinusoidale aplicate la intrare cu diverse nivele și frecvențe (SFRD, IP2 și IP3, amplitudinile celor mai semnificative componente parazite) după precizarea frecvenței și fluctuațiilor tactului de eșantionare.

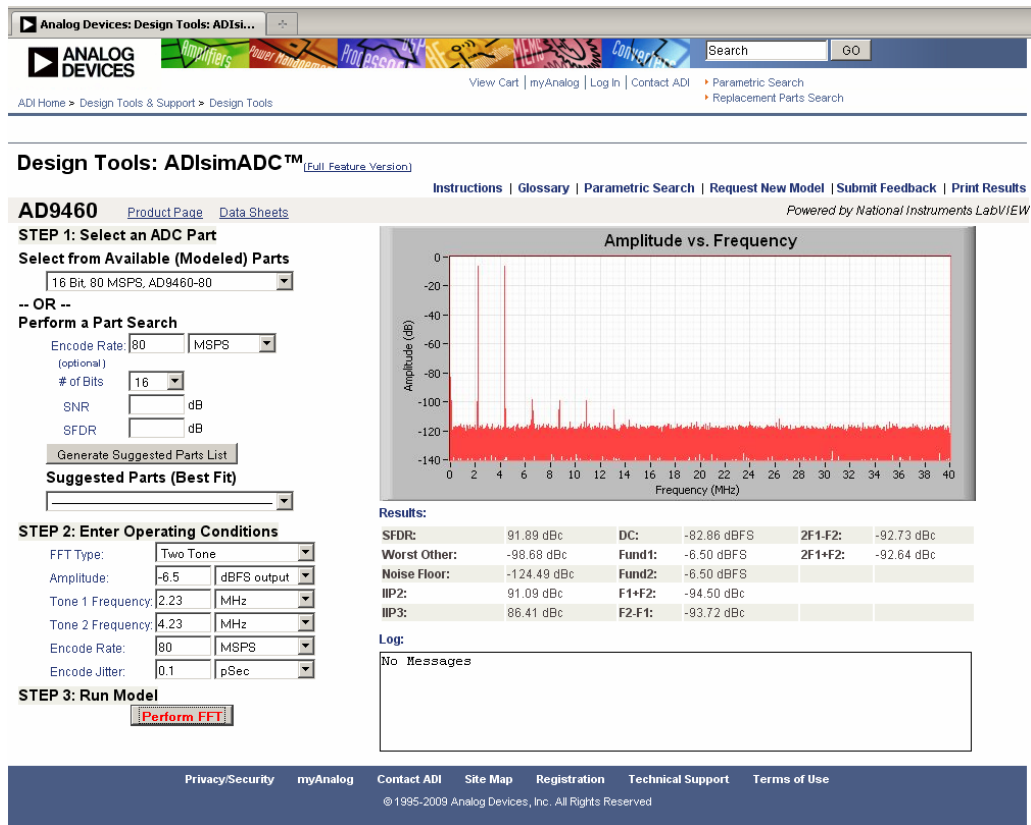


Figura 1

Curs: Echipamente Radio Definite prin Program și Virtuale

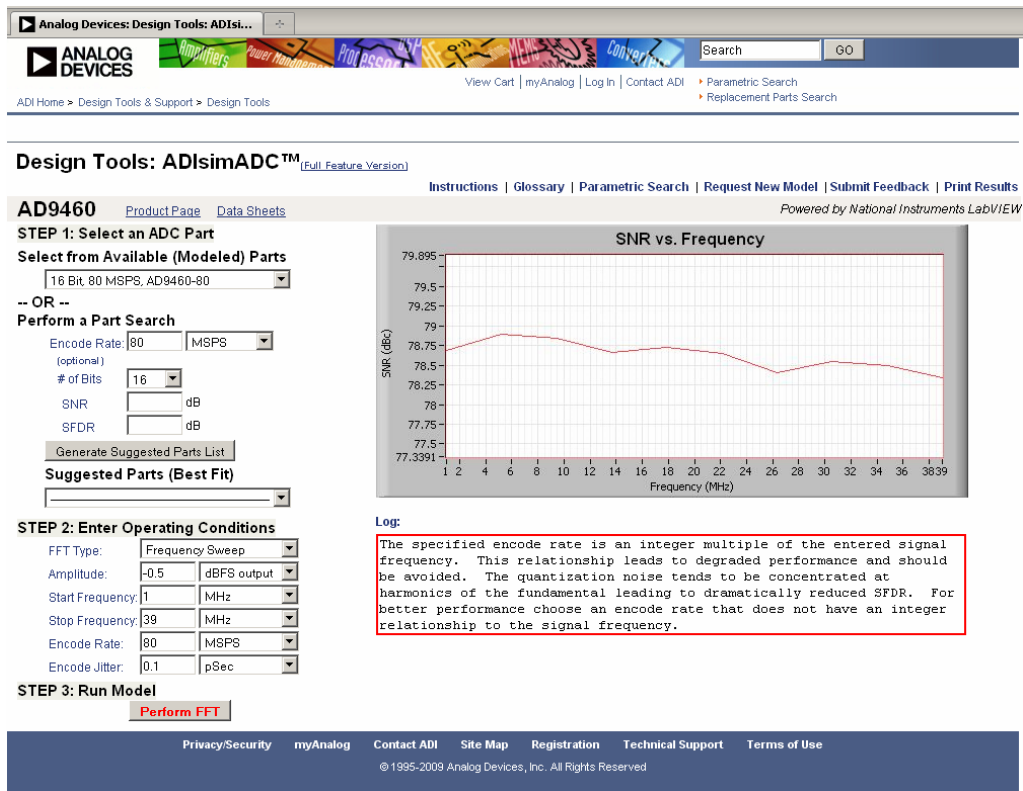


Figura 2

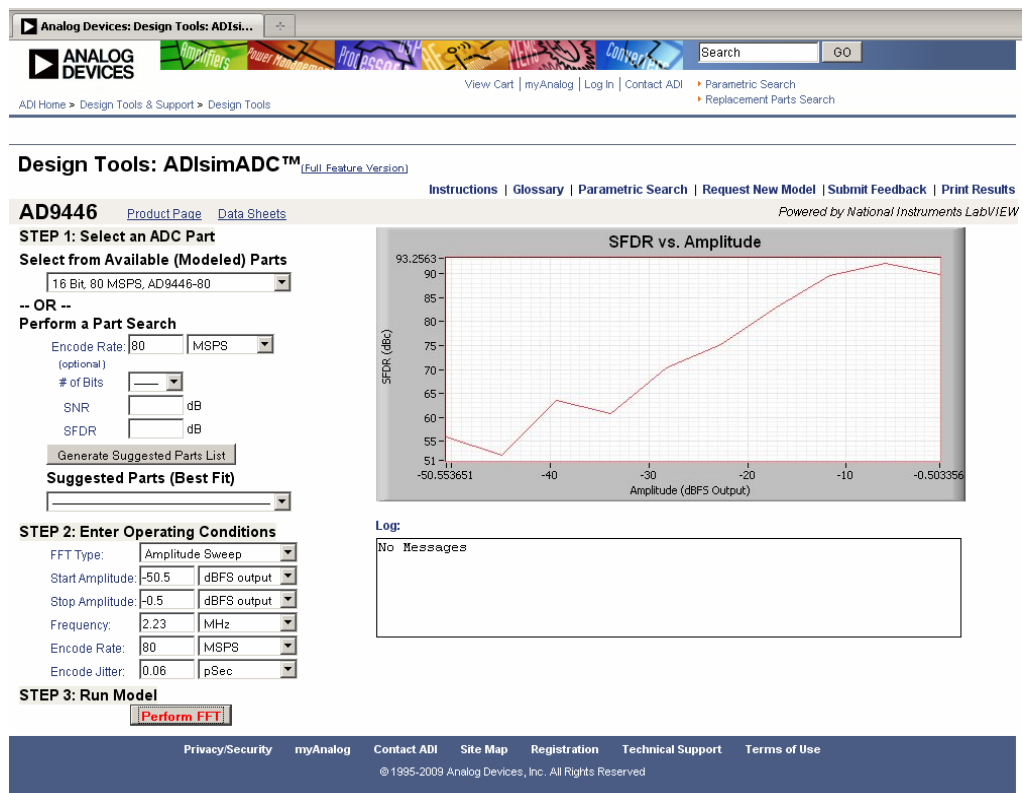


Figura 3

În mod asemănător, pentru analiza conversiei numeric-analogice este disponibilă ([5]) o aplicație care arată imaginile armonice și alte semnale parazite la ieșirea unui dispozitiv ideal, ca în figura 4, sau la unul concret ca în figura 4 (circuitul AD9777).

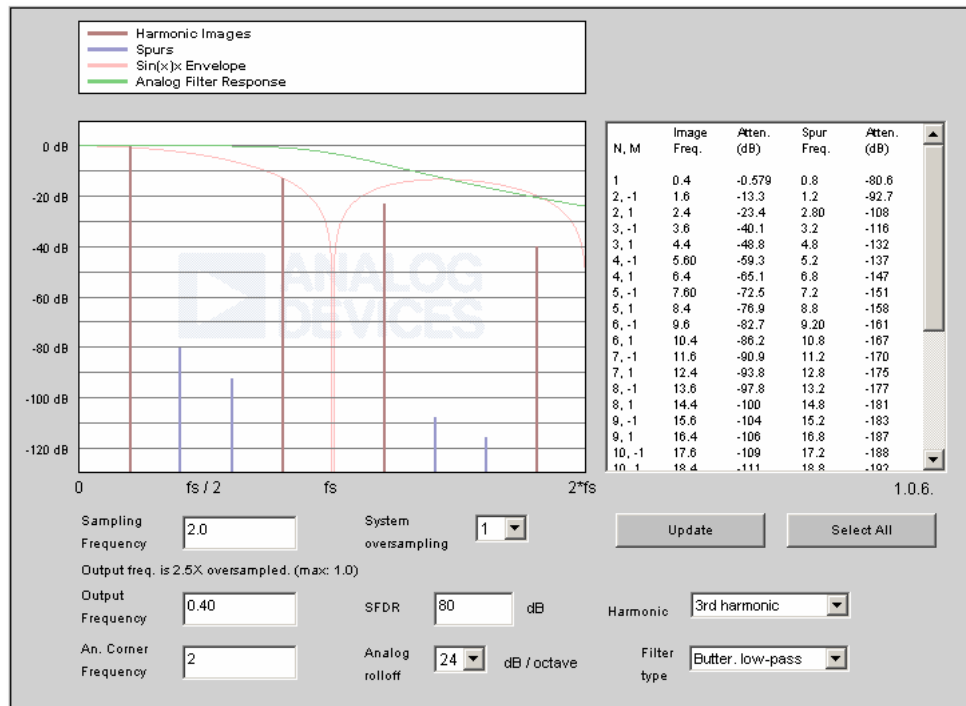


Figura 4

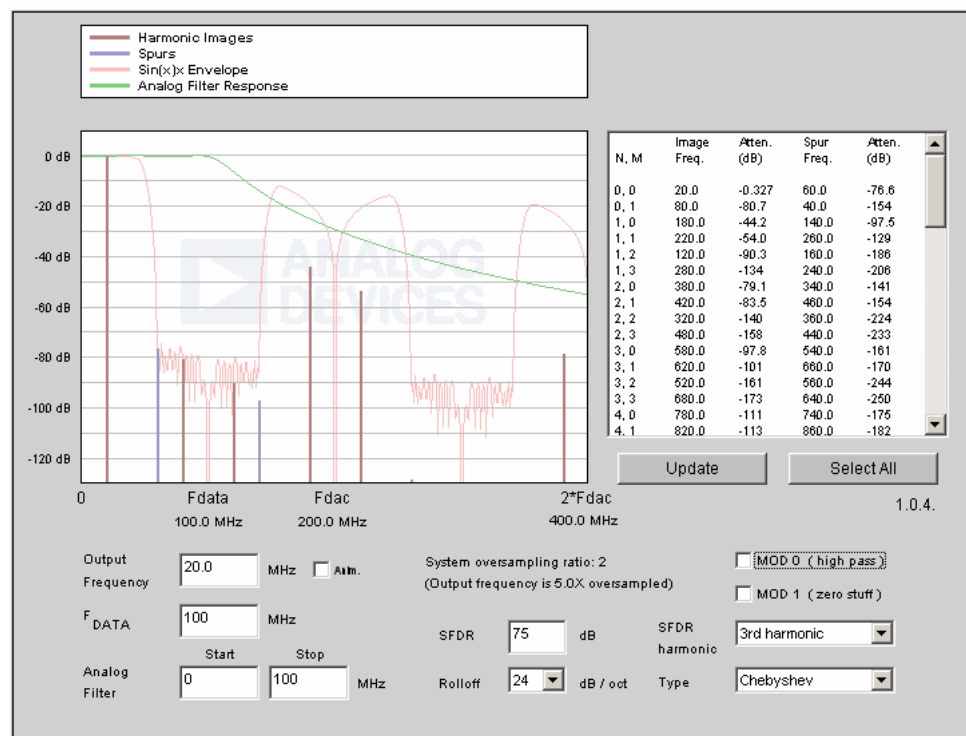


Figura 5

Același pachet de simulatoare interactive on-line conține și aplicația **ADIsimDDS™** ([6]) destinată asistării în selectarea și evaluarea circuitelor sintetizoare numerice directe (DDS) produse de Analog Devices. Se permite utilizatorului să selecteze un anumit dispozitiv, să-i definească condițiile de lucru dorite (tactul de referință și factorul său de multiplicare, frecvența de ieșire nominală, configurația filtrului de ieșire opțional) și să-i evalueze performanțele generale pe baza ecuațiilor matematice care-i descriu funcționarea relativ la structura sa internă. Rezultatele analizei (de exemplu pentru circuitul DDS AD9858), obținute sub forma grafică din figura 6 sau cea tabelară ca în figura 7, sînt utile numai în faza de proiectare, neputînd să suplinească integral testarea dispozitivelor prin intermediul plăcilor de evaluare oferite de fabricant (pe baza cărora se întocmesc de fapt specificațiile din foile de catalog).

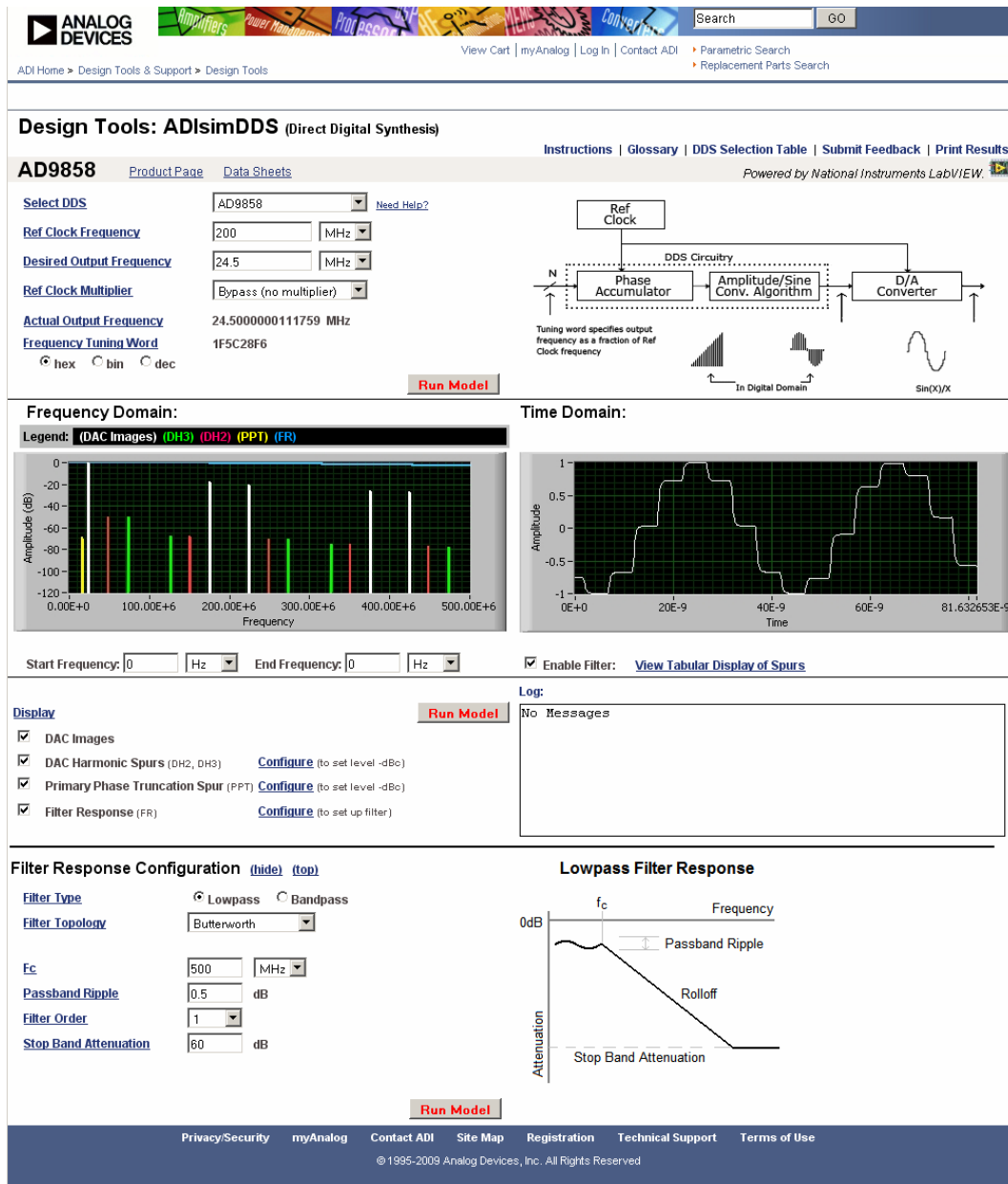


Figura 6

Tabular Display of Spurs (hide) (top) KEY: f1 = Desired Output Frequency, f2 = Ref Clock, PPT Spur = Primary Phase Truncation Spur											
DAC Images			DAC Harmonics 2nd Order			DAC Harmonics 3rd Order			Other Spurs		
Formula	Frequency (Hz)	Attenuation (dBc)	Formula	Frequency (Hz)	Attenuation (dBc)	Formula	Frequency (Hz)	Attenuation (dBc)	Name	Frequency (Hz)	Attenuation (dBc)
f1	2.450000E+7	-0.215474	2f1	4.900000E+7	-50.215474	3f1	7.350000E+7	-50.215474	PPT Spur	1.600037E+7	-70.003326
f2-f1	1.755000E+8	-17.317694	f2-2f1	1.510000E+8	-67.317694	f2-3f1	1.265000E+8	-67.317694			
f2+f1	2.245000E+8	-19.456479	f2+2f1	2.490000E+8	-69.456479	f2+3f1	2.735000E+8	-69.456479			
2f2-f1	3.755000E+8	-23.924351	2f2-2f1	3.510000E+8	-73.924351	2f2-3f1	3.265000E+8	-73.924351			
2f2+f1	4.245000E+8	-24.989706	2f2+2f1	4.490000E+8	-74.989706	2f2+3f1	4.735000E+8	-74.989706			
3f2-f1	5.755000E+8	-27.633058	3f2-2f1	5.510000E+8	-77.633058	3f2-3f1	5.265000E+8	-77.633058			
3f2+f1	6.245000E+8	-28.342801	3f2+2f1	6.490000E+8	-78.342801	3f2+3f1	6.735000E+8	-78.342801			
4f2-f1	7.755000E+8	-30.223788	4f2-2f1	7.510000E+8	-80.223788	4f2-3f1	7.265000E+8	-80.223788			
4f2+f1	8.245000E+8	-30.755965	4f2+2f1	8.490000E+8	-80.755965	4f2+3f1	8.735000E+8	-80.755965			
5f2-f1	9.755000E+8	-32.216697	5f2-2f1	9.510000E+8	-82.216697	5f2-3f1	9.265000E+8	-82.216697			
5f2+f1	1.024500E+9	-32.642391	5f2+2f1	1.049000E+9	-82.642391	5f2+3f1	1.073500E+9	-82.642391			
6f2-f1	1.175500E+9	-33.836605	6f2-2f1	1.151000E+9	-83.836605	6f2-3f1	1.126500E+9	-83.836605			
6f2+f1	1.224500E+9	-34.191328	6f2+2f1	1.249000E+9	-84.191328	6f2+3f1	1.273500E+9	-84.191328			
7f2-f1	1.375500E+9	-35.201364	7f2-2f1	1.351000E+9	-85.201364	7f2-3f1	1.326500E+9	-85.201364			
7f2+f1	1.424500E+9	-35.505401	7f2+2f1	1.449000E+9	-85.505401	7f2+3f1	1.473500E+9	-85.505401			
8f2-f1	1.575500E+9	-36.380520	8f2-2f1	1.551000E+9	-86.380520	8f2-3f1	1.526500E+9	-86.380520			
8f2+f1	1.624500E+9	-36.646546	8f2+2f1	1.649000E+9	-86.646546	8f2+3f1	1.673500E+9	-86.646546			
9f2-f1	1.775500E+9	-37.418565	9f2-2f1	1.751000E+9	-87.418565	9f2-3f1	1.726500E+9	-87.418565			
9f2+f1	1.824500E+9	-37.655029	9f2+2f1	1.849000E+9	-87.655029	9f2+3f1	1.873500E+9	-87.655029			
10f2-f1	1.975500E+9	-38.345693	10f2-2f1	1.951000E+9	-88.345693	10f2-3f1	1.926500E+9	-88.345693			
10f2+f1	2.024500E+9	-38.558508	10f2+2f1	2.049000E+9	-88.558508	10f2+3f1	2.073500E+9	-88.558508			
11f2-f1	2.175500E+9	-39.183334	11f2-2f1	2.151000E+9	-89.183334	11f2-3f1	2.126500E+9	-89.183334			
11f2+f1	2.224500E+9	-39.376800	11f2+2f1	2.249000E+9	-89.376800	11f2+3f1	2.273500E+9	-89.376800			
12f2-f1	2.375500E+9	-39.947253	12f2-2f1	2.351000E+9	-89.947253	12f2-3f1	2.326500E+9	-89.947253			
12f2+f1	2.424500E+9	-40.124596	12f2+2f1	2.449000E+9	-90.124596	12f2+3f1	2.473500E+9	-90.124596			
13f2-f1	2.575500E+9	-40.649383	13f2-2f1	2.551000E+9	-90.649383	13f2-3f1	2.526500E+9	-90.649383			
13f2+f1	2.624500E+9	-40.813083	13f2+2f1	2.649000E+9	-90.813083	13f2+3f1	2.673500E+9	-90.813083			
14f2-f1	2.775500E+9	-41.298977	14f2-2f1	2.751000E+9	-91.298977	14f2-3f1	2.726500E+9	-91.298977			
14f2+f1	2.824500E+9	-41.450983	14f2+2f1	2.849000E+9	-91.450983	14f2+3f1	2.873500E+9	-91.450983			
15f2-f1	2.975500E+9	-41.903351	15f2-2f1	2.951000E+9	-91.903351	15f2-3f1	2.926500E+9	-91.903351			
15f2+f1	3.024500E+9	-42.045224	15f2+2f1	3.049000E+9	-92.045224	15f2+3f1	3.073500E+9	-92.045224			
16f2-f1	3.175500E+9	-42.468394	16f2-2f1	3.151000E+9	-92.468394	16f2-3f1	3.126500E+9	-92.468394			
16f2+f1	3.224500E+9	-42.601400	16f2+2f1	3.249000E+9	-92.601400	16f2+3f1	3.273500E+9	-92.601400			
Privacy/Security myAnalog Contact ADI Site Map Registration Technical Support Terms of Use © 1995-2009 Analog Devices, Inc. All Rights Reserved											

Figura 7

Tot pentru abordarea oscilatoarelor din echipamente radio este dedicat și programul **ADIsimPLL**TM ([7]) care simplifică și îmbunătățește procesul tradițional de proiectare aplicabil sintetizoarelor de frecvență cu buclă PLL. Utilizatorul noului mediu de proiectare este îndrumat să specifice caracteristicile generale de frecvență ale proiectului, natura întregă sau fracționară a factorului de multiplicare al frecvenței de referință, circuitul integrat PLL și oscilatorul variabil (din biblioteca aplicației), precum și topologia filtrului buclei, ca în exemplul din figura 8. Programul determină valorile elementelor filtrului buclei și afișează parametrii cheie din domeniile frecvență și timp: zgomotul de fază, timpul și dinamica mecanismului de prindere, componentele parazite injectate de semnalul de referință.

*

Ultimul exemplu se referă la o aplicație (gratuită), **WinFilter** ([8]), utilă în proiectarea filtrelor numerice FIR (figura 11) și IIR (figura 12) trece-jos, trece-sus, trece-bandă și oprește bandă avînd caracteristică de tip Butterworth, Chebișev, Bessel, Rectangulară sau Cosinus ridicat. Execuția programului oferă sub formă grafică răspunsul filtrului în frecvență (amplitudine, fază, timp de întârziere) pentru numărul de prize sau ordinul adoptat iar sub formă text valorile coeficienților filtrului și chiar codul C prin care aceștia pot fi folosiți, așa cum arată Listele 1 și 2.

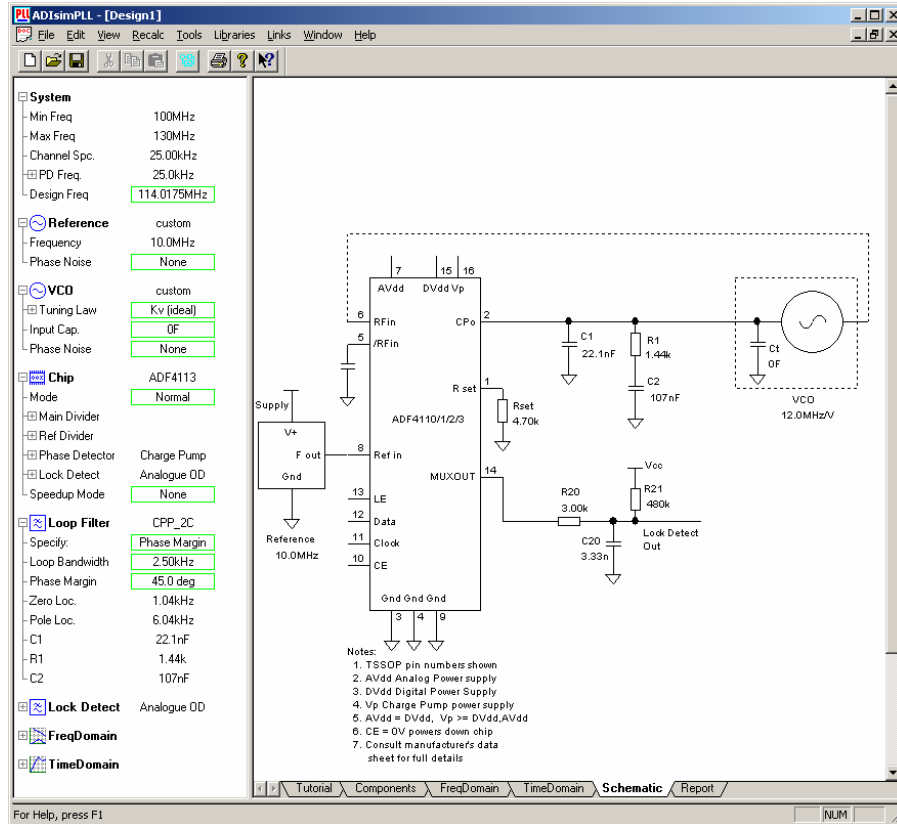


Figura 8

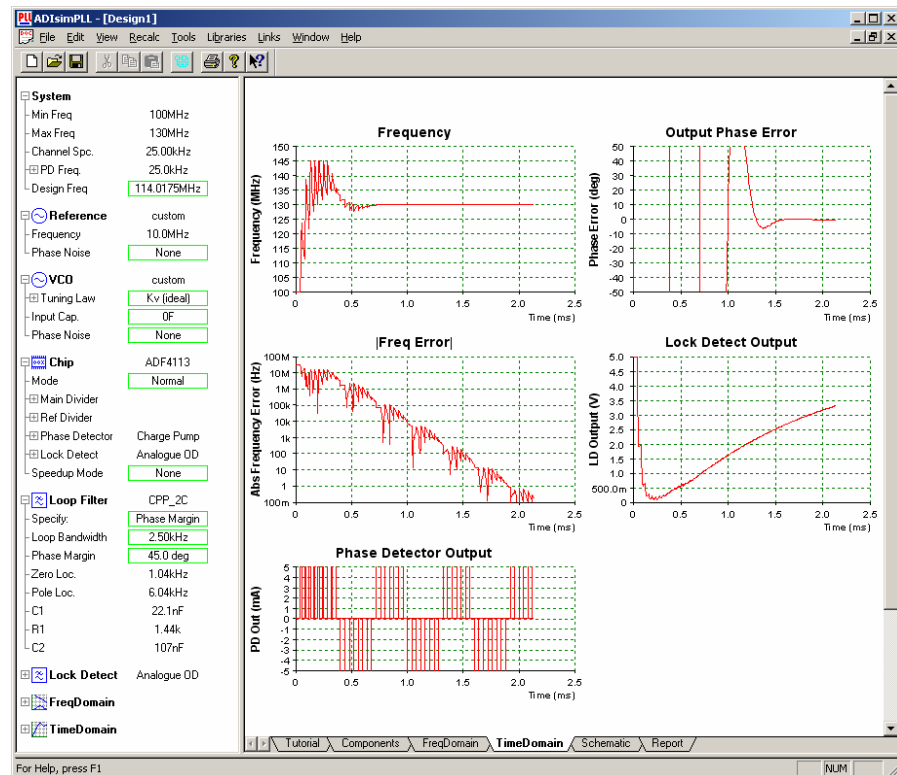


Figura 9

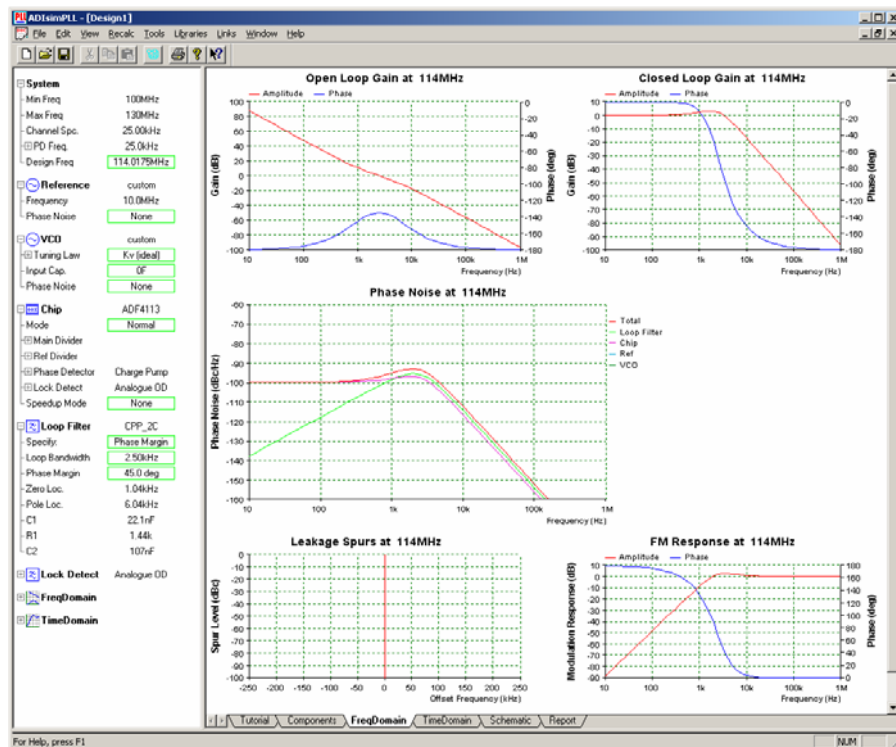


Figura 10

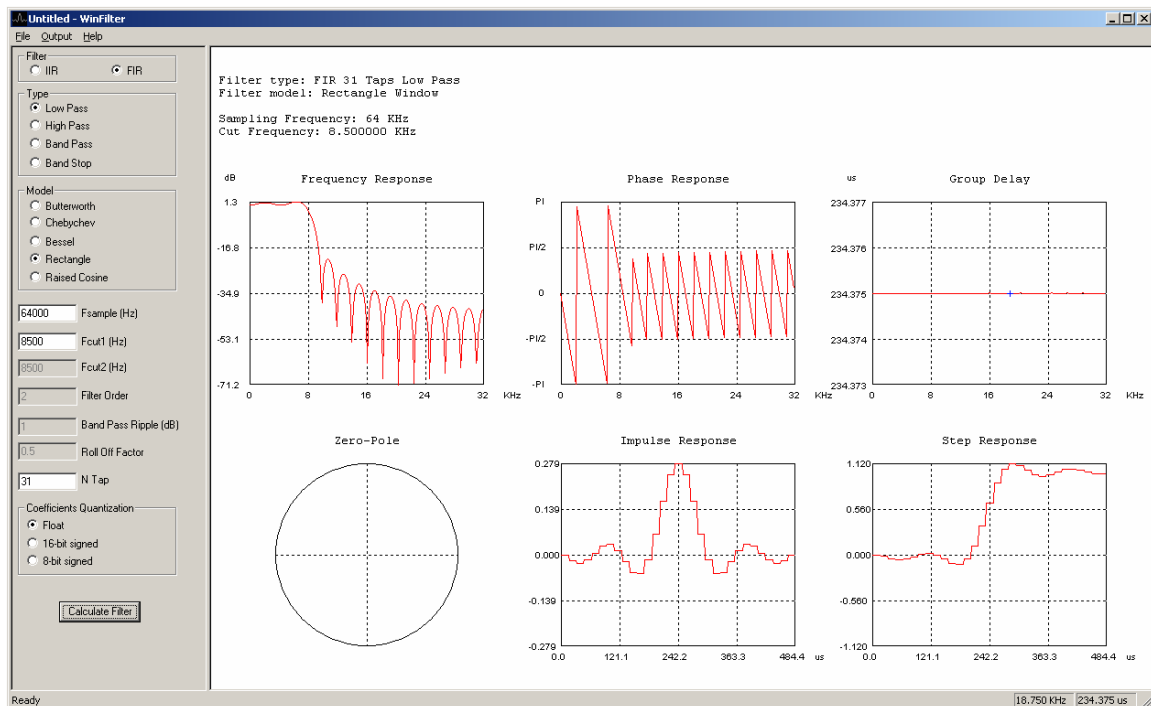


Figura 11

/*****

WinFilter version 0.8
<http://www.winfilter.20m.com>
akundert@hotmail.com


```
Filter type: Low Pass
Filter model: Rectangular Window
Sampling Frequency: 64 KHz
Cut Frequency: 8.500000 KHz
Coefficients Quantization: float
*****/

#define Ntap 31

float fir(float NewSample) {
    float FIRCoef[Ntap] = {
        -0.00109055374174134990,
        -0.01841445349941429900,
        -0.02538514086667637000,
        -0.01545053392026551400,
        0.00737376993843938140,
        0.02944800736845468500,
        0.03494040711295549900,
        0.01597983414694742800,
        -0.02040790738110919200,
        -0.05329775888344757300,
        -0.05733439250682011200,
        -0.01630272219436561900,
        0.06637864652619016000,
        0.16635027217423592000,
        0.24771694931613455000,
        0.27899115282096515000,
        0.24771694931613455000,
        0.16635027217423592000,
        0.06637864652619016000,
        -0.01630272219436561900,
        -0.05733439250682011200,
        -0.05329775888344757300,
        -0.02040790738110919200,
        0.01597983414694742800,
        0.03494040711295549900,
        0.02944800736845468500,
        0.00737376993843938140,
        -0.01545053392026551400,
        -0.02538514086667637000,
        -0.01841445349941429900,
        -0.00109055374174134990
    };

    static float x[Ntap]; //input samples
    float y=0;           //output sample
    int n;

    //shift the old samples
    for(n=Ntap-1; n>0; n--)
        x[n] = x[n-1];

    //Calculate the new output
    x[0] = NewSample;
    for(n=0; n<Ntap; n++)
```



```

y += FIRCoeff[n] * x[n];

return y;
}

```

Lista 1

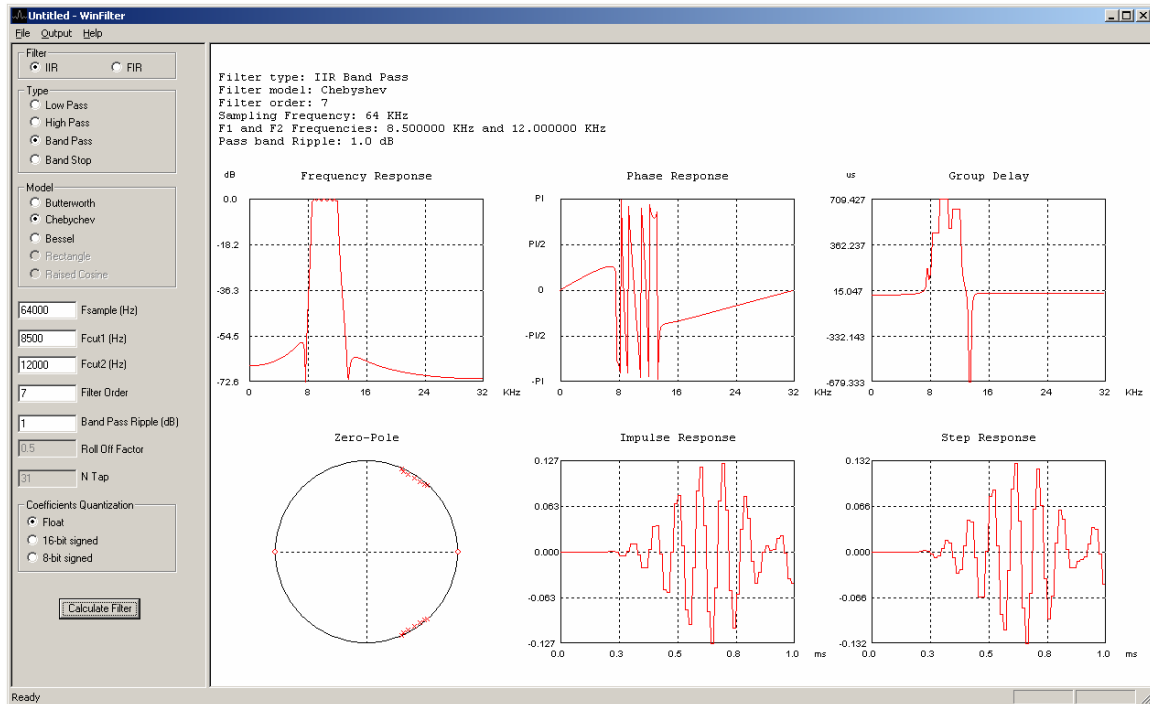


Figura 12

/******

WinFilter version 0.8
<http://www.winfilter.20m.com>
 akundert@hotmail.com

Filter type: Band Pass
 Filter model: Chebyshev
 Filter order: 7
 Sampling Frequency: 64 KHz
 Fc1 and Fc2 Frequencies: 8.500000 KHz and 12.000000 KHz
 Pass band Ripple: 1.000000 dB
 Coefficients Quantization: float

Z domain Zeros
 z = -1.000000 + j 0.000000
 z = -1.000000 + j 0.000000
 z = -1.000000 + j 0.000000
 z = -1.000000 + j 0.000000
 z = -1.000000 + j 0.000000
 z = -1.000000 + j 0.000000
 z = -1.000000 + j 0.000000
 z = 1.000000 + j 0.000000
 z = 1.000000 + j 0.000000
 z = 1.000000 + j 0.000000

```
z = 1.000000 + j 0.000000
z = 1.000000 + j 0.000000
z = 1.000000 + j 0.000000
z = 1.000000 + j 0.000000
```

Z domain Poles

```
z = 0.524303 + j -0.810107
z = 0.524303 + j 0.810107
z = 0.459913 + j -0.850671
z = 0.459913 + j 0.850671
z = 0.586480 + j -0.772750
z = 0.586480 + j 0.772750
z = 0.407408 + j -0.887500
z = 0.407408 + j 0.887500
z = 0.635817 + j -0.746161
z = 0.635817 + j 0.746161
z = 0.380242 + j -0.915682
z = 0.380242 + j 0.915682
z = 0.666468 + j -0.736339
z = 0.666468 + j 0.736339
```

```
*****/
```

```
#define NCoef 14
```

```
float iir(float NewSample) {
```

```
    float ACoef[NCoef+1] = {
        0.00000015424544888636,
        0.00000000000000000000,
        -0.00000107971814220450,
        0.00000000000000000000,
        0.00000323915442661351,
        0.00000000000000000000,
        -0.00000539859071102251,
        0.00000000000000000000,
        0.00000539859071102251,
        0.00000000000000000000,
        -0.00000323915442661351,
        0.00000000000000000000,
        0.00000107971814220450,
        0.00000000000000000000,
        -0.00000015424544888636
    };
```

```
    float BCoef[NCoef+1] = {
        1.00000000000000000000,
        -7.32126220381071310000,
        29.51319876018394300000,
        -81.24667058730766200000,
        168.49993934612681000000,
        -275.00767230718401000000,
        362.73551900187016000000,
        -391.21390031068916000000,
        346.67221337603667000000,
        -251.18663388192914000000,
        147.08371244031179000000,
        -67.77499120678625400000,
        23.52751505093606700000,
    };
```

```
-5.57751770565678750000,  
0.72826180863378542000  
};  
  
static float y[NCoef+1]; //output samples  
static float x[NCoef+1]; //input samples  
int n;  
  
//shift the old samples  
for(n=NCoef; n>0; n--) {  
    x[n] = x[n-1];  
    y[n] = y[n-1];  
}  
  
//Calculate the new output  
x[0] = NewSample;  
y[0] = ACoef[0] * x[0];  
for(n=1; n<=NCoef; n++)  
    y[0] += ACoef[n] * x[n] - BCoef[n] * y[n];  
  
return y[0];  
}
```

Lista 2

BIBLIOGRAFIE

- [1] – Luong Nguyen, Vincent Janicot, “*Simulation Method to Extract Characteristics for Digital Wireless Communication Systems*”, Mentor Graphics Corporation, 2001
- [2] – David Leiss, Greg Jue, “*Simulation and Design of SoftwareDefined Radio*”, Agilent Technologies, 2007
- [3] - ----, “AN-905 *VisualAnalogTM Converter Evaluation Tool Version 1.0 User Manual*”, Analog Devices
- [4] - <http://www.analog.com/en/design-center/>
- [5] - <http://www.analog.com/en/design-tools/dt-dac-harmonic-analyzer/design-center/list.html>
- [6] - <http://designtools.analog.com/dtDDSWeb/dtDDSMMain.aspx>
- [7] - http://forms.analog.com/form_pages/rfcomms/adisimpll.asp
- [8] - <http://www.winfilter.20m.com/>