

RF Special Edition:

Complete Simulation of a 137MHz Weather Satellite Converter using LTspice IV



Tutorial Part 2

**For the free full SPICE version, available at Linear Technologies
(www.linear.com)**

Version 1.1

Copyright by
Guntherd Kraus, Elektronikschule Tettnang, Germany,
Guest Lecturer at the DHBW (Duale Hochschule Baden-Wuerttemberg),
Friedrichshafen, Germany

Email: krausg@elektronikschule.de or Guntherd.Kraus@gmx.de
Homepage: www.elektronikschule.de/~krausg

May 1st, 2013

Note:

This edition is a shortened version of an article with 3 parts, published in the German „UKW-Berichte“ journal and named “Simulations in the RF range using LTspice”.

Herein is shown that SPICE can be used to simulate nearly every circuit property in the time and in the frequency domain and also in the RF range, including nonlinear simulations.

This issue shall be a teaching for „Do it Yourself“ in form of a cooperation between the author and the reader. So start your PC and begin -- but you should already have studied part 1 of this tutorial to know how to handle LTspice.

Contents	Page
1. Why SPICE for the RF range?	3
2. The object: a 137MHz converter for the reception of NOAA weather satellite signals	3
3. Low noise preamplifier using the Dual Gate MOSFET BF998	4
3.1. Creating a symbol for the BF998	4
3.2. SPICE simulation of the S Parameters for a drain current of 10mA	6
3.2.1. Setting this operating point	6
3.2.2. Simulating the S Parameters at this operating point	7
3.3. LTspice Simulation of the complete 137MHz LNA	9
3.3.1. Choosing and setting the operating point	9
3.3.2. S Parameter Simulation of the LNA	10
3.3.3. Noise figure simulation	12
3.3.4. Stability control	14
3.3.5. Simulation in the time domain using a 137MHz input signal	15
3.3.6. IP3 point simulation	16
4. The Double Balanced Mixer (= ring modulator)	20
4.1. Fundamentals	20
4.2. Simulation of the DBM using LTspice	21
4.2.1. Simulation schematic	21
4.2.2. Simulation of the DBM properties	22
5. Band pass filter at the DBM IF output	25
5.1. Notes	25
5.2. Filter design using the Ansoft Designer SV	25
6. Simulation of the complete converter circuit	29
6.1. Preparations	29
6.2. Gain at centre frequency	30
6.3. Pass band curve simulation	30
6.3.1. Simulation using separate spectral lines	30
6.3.2. Simulation using a Dirac pulse	32
6.4. IP3 simulation	35
6.5. Noise Figure of the complete converter	37
Literature	38

1. Why SPICE for the RF range?

That is what you ask yourself when you have looked at or worked with good, but cheap or free S Parameter Simulation Programs (like PUFF or Ansoft Designer SV). But the answer is very simple:

If a SPICE simulation of a RF circuit delivers the same results as an S Parameter simulation at the same operating point then there are unbelievable possibilities:

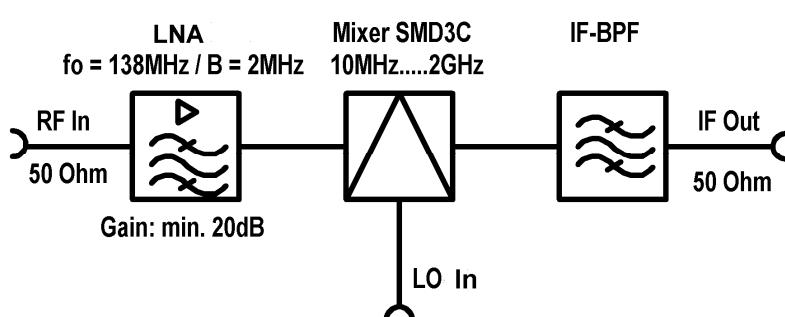
The SPICE model now represents exactly the part and its measured S Parameters. So you are allowed to vary currents or voltages in your circuit without an S Parameter file from the manufacturer, you can use other signal forms at the input, you can overdrive the stage – in any case will SPICE produce the correct solution in the time domain or in the frequency domain, including all details of a new DC operating point. And you can easily search for the optimum by parameter sweeps.

But be aware, that even by this fact an S Parameter simulation will never obsolete. This form of simulation is based on a collection of EXACTLY MEASURED VALUES during production and this will always be the reference! SPICE is always working with a model which shall represent the properties of the part (and sometimes this representation is not perfect at all). And for some special RF parts (like microstrip transmission lines) there don't exist any special models and you have to modify and to adapt the general model for a transmission line with losses for your purpose – as well as possible.

So the special procedures to simulate the properties of RF circuits are demonstrated at an interesting example. Let's go.

2. The object: a 137MHz converter for the reception of NOAA weather satellite signals

A part of this project was published in an earlier edition of the "UKW-Berichte" journal [1] („PUFF-Einsteigerprojekt. Rauscharmer Vorverstärker für 137MHz bzw. 145MHz“. Edition 4/1998, page 245) and described a low noise preamplifier using the Dual Gate MOSFET BF998 in the 136...138MHz frequency range. Published are the simulation and the measured prototype results, so SPICE can now demonstrate its strength in comparison.

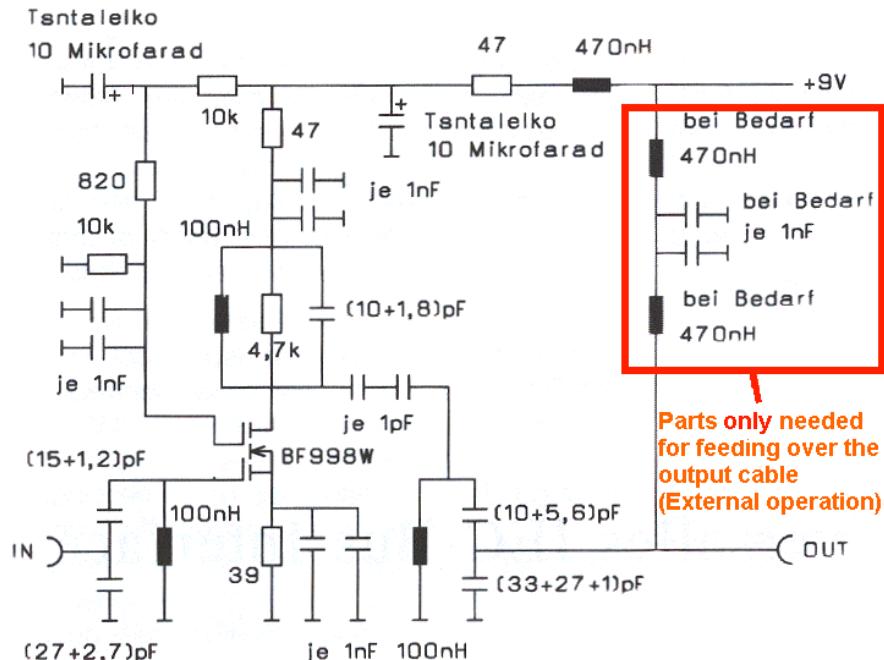


The rest of the converter is not too complicated: the LNA is followed by a Double Balanced Mixer SMD3C (in a milled alumina case) and a 100 MHz IF band pass filter, also in a separated milled alumina case. Frequency conversion down to 88...102MHz is realized by an extern 37MHz signal generator in the radio shack. Amplification and demodulation are done by a high class car radio with

perfect shielding.

3. Low noise preamplifier using the Dual Gate MOSFET BF998

This is the original schematic as published.



Noise matching at the input is realized by a capacitive voltage divider. At the output a second order band pass filter is used for selection and matching to 50Ω . The coupling element between the two resonant circuits at the output is an interdigital capacitor with 0.5pF .

The voltage gain of the stage is approx. 20dB and the noise figure NF below 1dB

3.1. Creating a symbol for the BF998

```

*   BF998 SPICE MODEL OCTOBER 1993 PHILIPS SEMICONDUCTORS
*   ENVELOPE SOT143 (R)
*   1.: SOURCE; 2.: DRAIN; 3.: GATE 2; 4.: GATE 1;
.SUBCKT BF998 1 2 3 4
L10      1 10    L=0.12N
L20      2 20    L=0.12N
L30      3 30    L=0.12N
L40      4 40    L=0.12N
L11      10 11   L=1.20N
L21      20 21   L=1.20N
L31      30 31   L=1.20N
L41      40 41   L=1.20N
C13      10 30   C=0.085P
C14      10 40   C=0.085P
C23      20 30   C=0.085P
C24      20 40   C=0.005P
D11      42 11   ZENER
D12      42 41   ZENER
D21      32 11   ZENER
D22      32 31   ZENER
RS       10 12   R=100
MOS1    61 41 11 12 GATE1 L=1.1E-6 W=1150E-6
MOS2    21 31 61 12 GATE2 L=2.0E-6 W=1150E-6

.MODEL ZENER D BV=10 CJO=1.2E-12 RS=10
.MODEL GATE1
+ NMOS LEVEL=3 UO=600 VTO=-0.250 NFS=300E9 TOX=42E-9
+ NSUB=3E15 VMAX=140E3 RS=2.0 RD=2.0 XJ=200E-9 THETA=0.11
+ ETA=0.06 KAPPA=2 LD=0.1E-6
+ CGSO=0.3E-9 CGDO=0.3E-9 CBD=0.5E-12 CBS=0.5E-12

.MODEL GATE2
+ NMOS LEVEL=3 UO=600 VTO=-0.250 NFS=300E9 TOX=42E-9
+ NSUB=3E15 VMAX=100E3 RS=2.0 RD=2.0 XJ=200E-9 THETA=0.11
+ ETA=0.06 KAPPA=2 LD=0.1E-6
+ CGSO=0.3E-9 CGDO=0.3E-9 CBD=0.5E-12 CBS=0.5E-12

.ENDS BF998

```

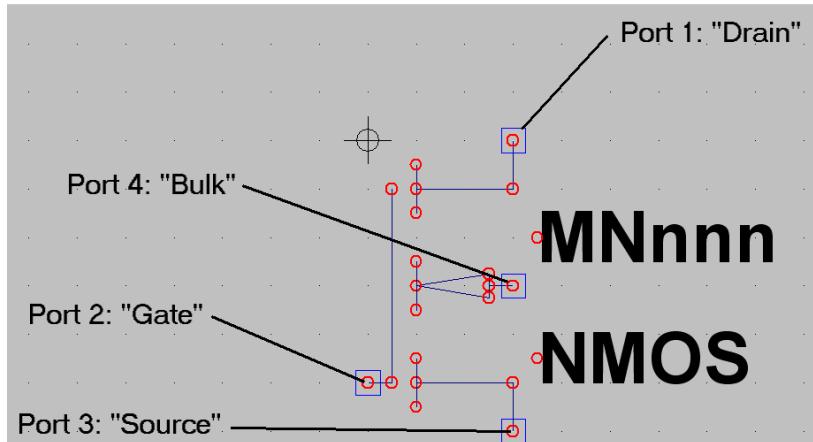
Order of part connections

Delete this!

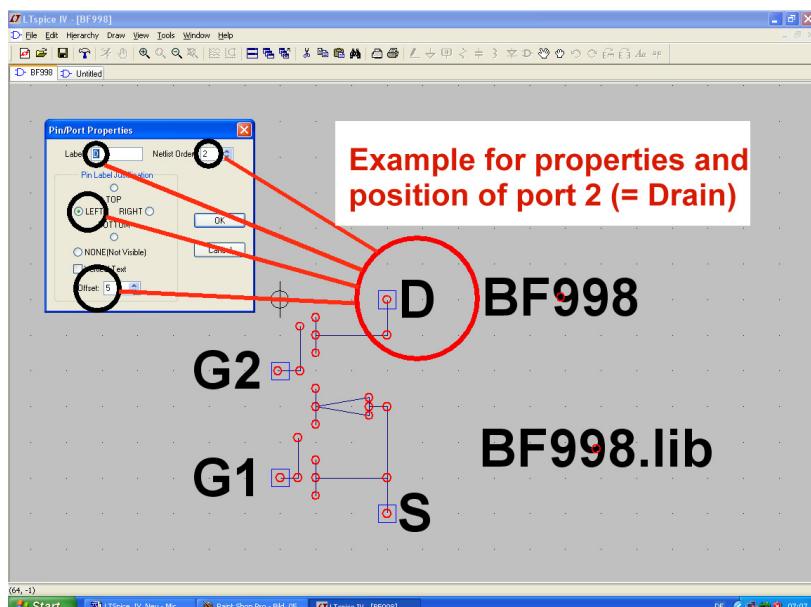
At first we need the SPICE model of the BF998 from the Internet (...enter "bf998 spice model" for the search!) and the result is shown beside. The file looks like very nice and the connections are numbered correctly, but there is a serious bug:

SPICE does not accept any additional comment or explanation when entering model part values. So please delete these and save the result as "BF998.lib" in the lib / sub folder of LTspice!

Now let us create a symbol for a Dual Gate MOSFET. Change to the "File" menu and select "New Symbol". All the tools to draw a symbol can be found under "DRAW".



Before starting pick up the symbol „**nmos4.asy**“ in the library (path: „LTspiceIV \ lib \ sym“) and put it on the screen. At once you know the dimensions if you want to draw yourself a completely new symbol. If you prefer this, press F5 and then left click on the symbol to delete it. Leave this function again by pressing “**ESC**” and then draw the complete new MOSFET.



Much easier is to modify the existing symbol, to rename it and save it as „**BF998.asy**“. But some information is necessary:

- 1) The small red circles mark the ends of lines. They are drawing helps AND catching points for connections when applying this part in a schematic.
- 2) The part connections (named “ports”) are little squares with a red circle in the centre. They cannot be drawn because there are special functions for this purpose.

So open “Edit” and search for “**Add Pin / Port**“. Left clicking opens the property menu and in the picture can be seen what you have to enter: the name of the connection, the symbol drawing (here: Left / Offset =5).

the Port / Pin number due to the SPICE model and the position in relation to the SPICE model and the position in relation to the SPICE model (here: Left / Offset =5). The pin numbering for the BF998 is: **Source = Port 1 / Drain = Port 2 / Gate2 = Port 3 / Gate 1 = Port 4**. At last close the menu for this port and connect it to the symbol by drawing a line.

3) If you want to move something on the screen, press F7 and left click on the part symbol. This will at once hang on the cursor and can be moved to the new position. Left click and then leave this function by selecting “**ESC**”.

attribute	value
Prefix	X
SpiceModel	BF998.lib
Value	BF998
Value2	BF998
SpiceLine	SpiceLine
SpiceLine2	SpiceLine2
Description	N-Channel Dual Gate MOSFET
ModelFile	ModelFile

Save at first this new created symbol at “**SpiceIV / lib / sym**“ as “**dgate_MOSFET.asy**” for tasks in the future. Then open “**EDIT**” and select “**Attributes / Edit Attributes**“ to enter the part properties in the left list. It is a little tricky because at first you have to click on the desired line and then enter the value in the **empty line ABOVE this list**. But the entry will only be saved if you afterwards click on another line in the list....

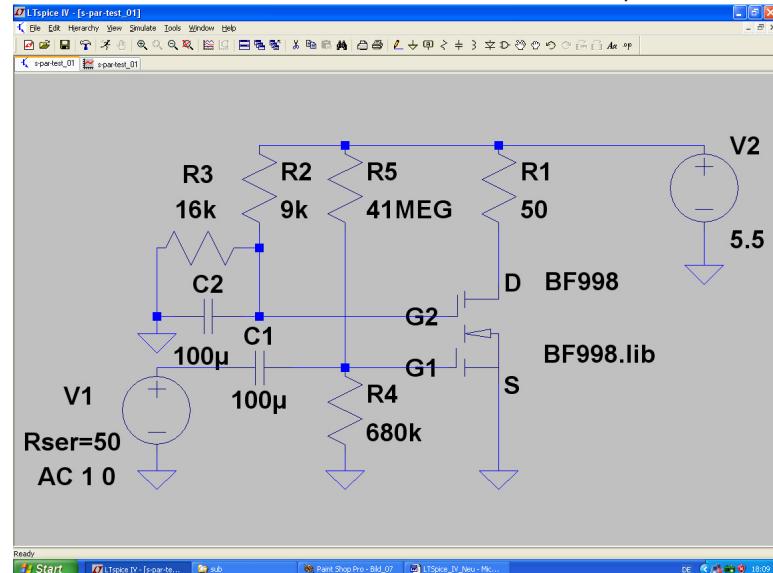
Now we are near to the end. Open once more “**EDIT**” but now “**Attribute / Attribute Window**“. So you can enter additional information which will appear together with the symbol. Another list appears and so please left click on “**VALUE**” and confirm with OK. Suddenly “**BF998**” is placed beside the symbol. Now repeat the procedure and activate the SPICE model name “**BF998.lib**“ to be visible together with the symbol. Save the complete symbol as **BF998.asy** in the **lib / sub folder of LTspice**.

All is successfully finished -- congratulations!

3.2. SPICE simulation of the S Parameters for a drain current of 10mA

3.2.1. Setting this operating point

From the Internet you can only get the S parameters for two Drain current values: 5mA and 10mA. So the first task is to simulate the S Parameters for 10mA and to compare the result with the given S parameter file.



So draw at first this schematic to check the operating point whether it is the same as given in the S parameter file ($I_D = 10\text{mA}$ / $U_{DS} = 5\text{V}$ / $U_{G2S} = 3,5\text{V}$).

Resistor, capacitor and ground symbol can be found in the menu bar. But for the voltage source f open the component library by this button (= at the right hand side of the diode symbol button, but marked by an AND gate). Then you find "voltage" in the list and you can place two of them on the screen.

The new BF998 symbol can also be found in the list and finally connect all together using the "WIRE" function (= button with pencil).

Programming of the voltage sources is very simple: click right on the symbol and then on "Advanced" to open the property menu.

For source V1 enter an **amplitude value of "1"** and a phase of "0" under "**Small Signal AC Analysis**". **Do not forget to enter 50Ω for the source resistor to prepare an S parameter simulation.**

Source V2 is the supply voltage with a DC value of 5.5V.

To change a resistor's or capacitor's value, click right on the symbol and enter the new value.

Now a little explanation is necessary: Drain resistor R1 serves as "output termination for the twoport with 50Ω ", but all other resistors are used to set the correct operating point ($I_D = 10\text{mA}$ / $U_{DS} = 5\text{V}$ / $U_{G2S} = 3,5\text{V}$). Gate 2 is grounded for AC by capacitor C2 and C1 is used to separate the source resistance of 50Ω and the voltage divider at Gate 1, consisting of R4 and R5.

--- Operating Point ---		
V(n001):	5.5	voltage
V(n003):	5.00077	voltage
V(n002):	3.52	voltage
V(n005):	0.0897312	voltage
V(n004):	4.48656e-016	voltage
I(C2):	3.52e-016	device_current
I(C1):	8.97312e-018	device_current
I(R5):	1.31958e-007	device_current
I(R4):	1.31958e-007	device_current
I(R3):	0.00022	device_current
I(R2):	0.00022	device_current
I(V1):	0.00998461	device_current
I(V2):	-0.0102047	device_current
I(U1):	8.97312e-018	device_current
Ix(u1:1):	-0.00998461	subckt_current
Ix(u1:2):	0.00998461	subckt_current
Ix(u1:3):	3.99684e-012	subckt_current
Ix(u1:4):	6.93895e-014	subckt_current

Now we need the simulation command for a DC analysis. Select "Simulate" and "Edit Simulation cmd", then "DC op. Pnt.". Press OK and the command .op will hang on the cursor and can be placed in the schematic.

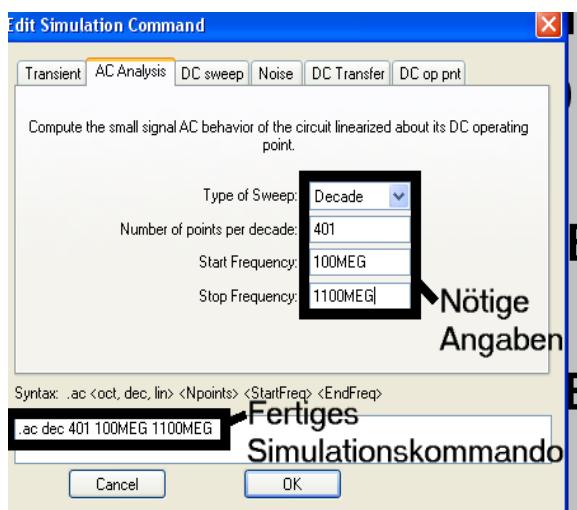
Start the simulation and you will get this list including every current and voltage of the schematic.

But there is another way with fewer efforts: close this list and move the cursor to an interesting connection in the circuit. At once you find the

current or the voltage for this point in the lower left corner of the screen.

If you now compare the marked voltage and current values with the conditions of the given S parameter file: everything is OK and we can start the S parameter simulation.

3.2.2. Simulating the S Parameters at this Operating Point



Select "Simulate", then "Edit Simulation cmd" and "AC Analysis". Use a decade sweep, a start frequency of 100MHz, a stop frequency of 1100MHz and 401 points per decade.

Add a SPICE-Directive to calculate the S parameters and select first "Edit" and "SPICE Directive". Now enter:

.net I(R1) V1

This means:
".net" invokes the network simulation.

"I(R1)" is the voltage at the output termination resistance $R1 = 50\Omega$.

"V1" relates this output voltage to the input voltage source which delivers the incident wave.

The directive ".op" of the last chapter is now automatically switched off and the simulation can be started. The diagram in the waveform viewer is still empty. So right click on it, select "ADD TRACE" and then the four S parameters S11 / S21 / S12 / S22. At last click on OK.

Now move the cursor over the scaling of the horizontal axis and left click to set a linear axis scaling **from 100MHz to 1100MHz with a tick of 100MHz**.

In the same manner the vertical axis scaling can be set as follows: **+20db to -80dB with a tick of 10dB**.

If you don't want to see the phase information, move the cursor over the scaling of the right vertical axis and click left. Then you can select "Don't plot Phase".



This is the success, but the result should be compared with other S parameter simulations.

```

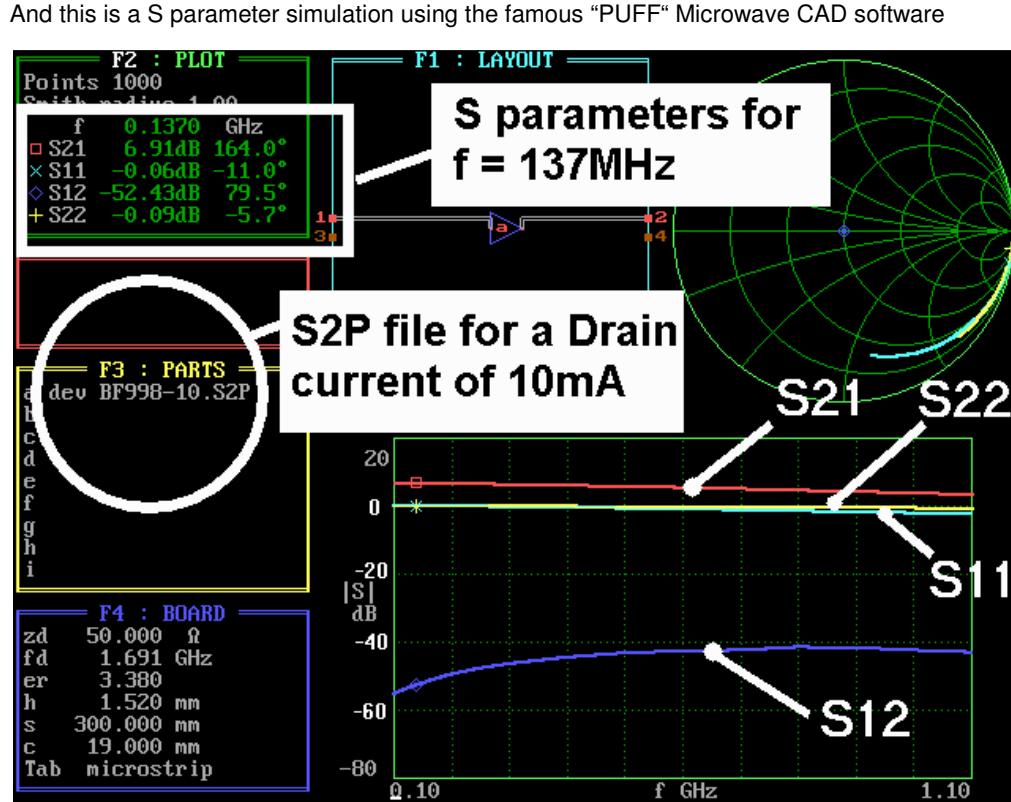
! SIEMENS small signal semiconductors
BF998
Si Dual Gate MOSFET Tetrode in SOT143
VDS = 5 V      VG2S = 3.5 V      ID = 10 mA
Common Source S-parameter S.

```

#	GHZ	S	MA	R	50	S11	S21	S12	S22
	GHZ	MAG	ANG	MAG	ANG	MAG	ANG	MAG	ANG
0.050	0.9992	-4.0	2.230	174.5	0.0009	89.0	0.9923	-2.1	
0.060	0.9991	-4.8	2.225	173.2	0.0011	86.1	0.9915	-2.5	
0.070	0.9988	-5.6	2.218	172.0	0.0013	85.9	0.9917	-3.0	
0.080	0.9974	-6.4	2.221	170.8	0.0015	84.8	0.9917	-3.4	
0.090	0.9969	-7.2	2.228	169.6	0.0016	83.6	0.9914	-3.7	
0.100	0.9968	-8.0	2.232	168.4	0.0018	83.1	0.9908	-4.1	
0.150	0.9918	-12.0	2.212	162.4	0.0026	78.6	0.9895	-6.2	
0.200	0.9849	-15.9	2.211	156.5	0.0035	74.5	0.9865	-8.4	
0.250	0.9769	-19.8	2.190	150.5	0.0043	71.2	0.9848	-10.4	
0.300	0.9680	-23.6	2.159	144.6	0.0050	67.6	0.9806	-12.4	
0.400	0.9451	-31.2	2.095	133.1	0.0062	60.8	0.9749	-16.4	
0.500	0.9199	-38.4	2.021	122.1	0.0072	54.4	0.9681	-20.3	
0.600	0.8945	-45.4	1.945	111.6	0.0077	48.8	0.9608	-24.0	
0.700	0.8686	-52.2	1.861	101.1	0.0079	45.5	0.9527	-27.7	
0.800	0.8382	-59.0	1.769	91.2	0.0088	39.9	0.9403	-31.4	
0.900	0.8085	-65.2	1.694	81.7	0.0086	30.6	0.9315	-35.2	
1.000	0.7824	-71.5	1.622	72.3	0.0079	26.6	0.9230	-38.9	
1.200	0.7256	-84.0	1.467	53.0	0.0069	21.1	0.9071	-46.9	
1.400	0.6697	-95.2	1.309	35.4	0.0061	4.9	0.8853	-54.1	
1.500	0.6464	-100.2	1.243	26.9	0.0043	-5.0	0.8781	-57.8	
1.600	0.6265	-105.1	1.187	18.5	0.0034	-19.2	0.8724	-61.6	
1.800	0.5887	-115.0	1.086	1.2	0.0029	-78.7	0.8643	-69.0	
2.000	0.5586	-124.4	1.001	-16.9	0.0064	-122.3	0.8494	-77.0	

f	Fmin	Gammaopt	rn/50
GHz	dB	MAG	ANG
0.010	3.00	0.80	1 3.40
0.050	1.30	0.80	3 2.00
0.100	0.80	0.80	6 1.40
0.200	0.70	0.80	12 1.10
0.450	1.00	0.78	27 1.00
0.800	1.75	0.73	48 0.96
0.900	2.00	0.71	53 0.94
1.100	2.35	0.63	65 0.84

! SIEMENS AG Semiconductor Group, Munich



Compared to the LTspice result there are only few and not very serious differences (especially when regarding S12). Congratulations to the author and creator of the SPICE model!

DC Operating Point

February 1991

At first a look at the S Parameter file as found in the Internet and saved in a folder on your PC (for a Drain current of 10mA):

Let us now compare the exact simulation results for a frequency of 137MHz and so compare a Time Domain simulation versus a S parameter simulation in the frequency domain in a table. (To get the exact results a cursor was used):

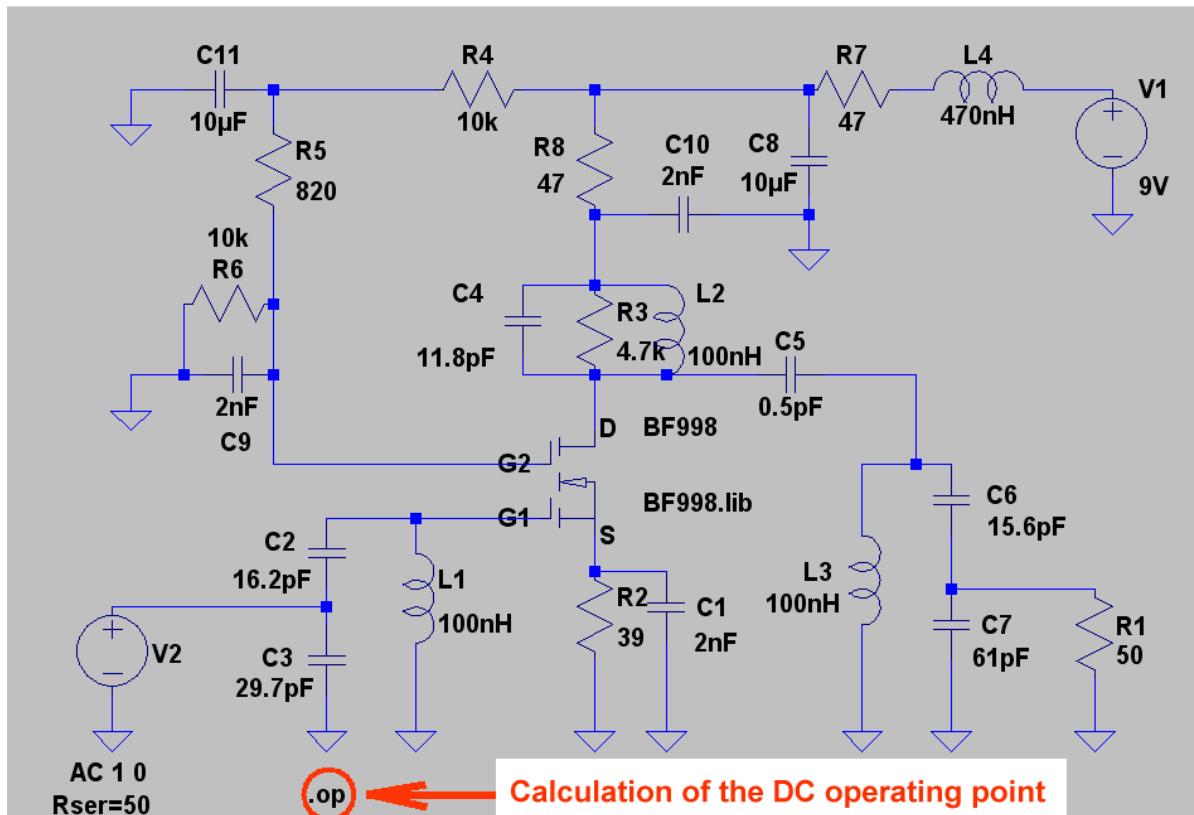
S parameter	LTS spice	PUFF
S11	-0,085dB	-0,06dB
S21	+7,95dB	+6,91dB
S12	-58dB	-52dB
S22	-0,05dB	-0,09dB

Who now starts to worry about the differences should remember: the S parameter file comes from Siemens, the SPICE model comes from PHILIPS. And we did not yet discuss any influence of tolerances of the part properties. This means that the LTspice S parameter simulation is absolutely believable.

3.3. LTspice Simulation of the complete 137MHz LNA

3.3.1. Choosing and setting the operating point

At first a really hard work: drawing the schematic for the simulation by using the circuit diagram as found in the publication.



You must not forget to add a series resistance of 1Ω at every coil with $100nH$ to represent a quality factor $Q = 60$ at 137MHz. einzutragen (therefore left click on the inductor symbol). Then we simulate the DC operating point and get these result values:

$$UDS = 8,1V \quad / \quad UG2S = 4V \quad / \quad IS = ID = 5mA$$

But these are exactly the conditions given in the S2p file for a Drain current of 5mA in the BF998 when you search in the Internet!

3.3.2. S Parameter Simulation of the LNA

This is easy:

- Select “Edit Simulation Command“ and write it for a decade sweep from 120 to 160MHz with **4001 points per decade**:

```
.ac dec 4001 120MEG 160MEG
```

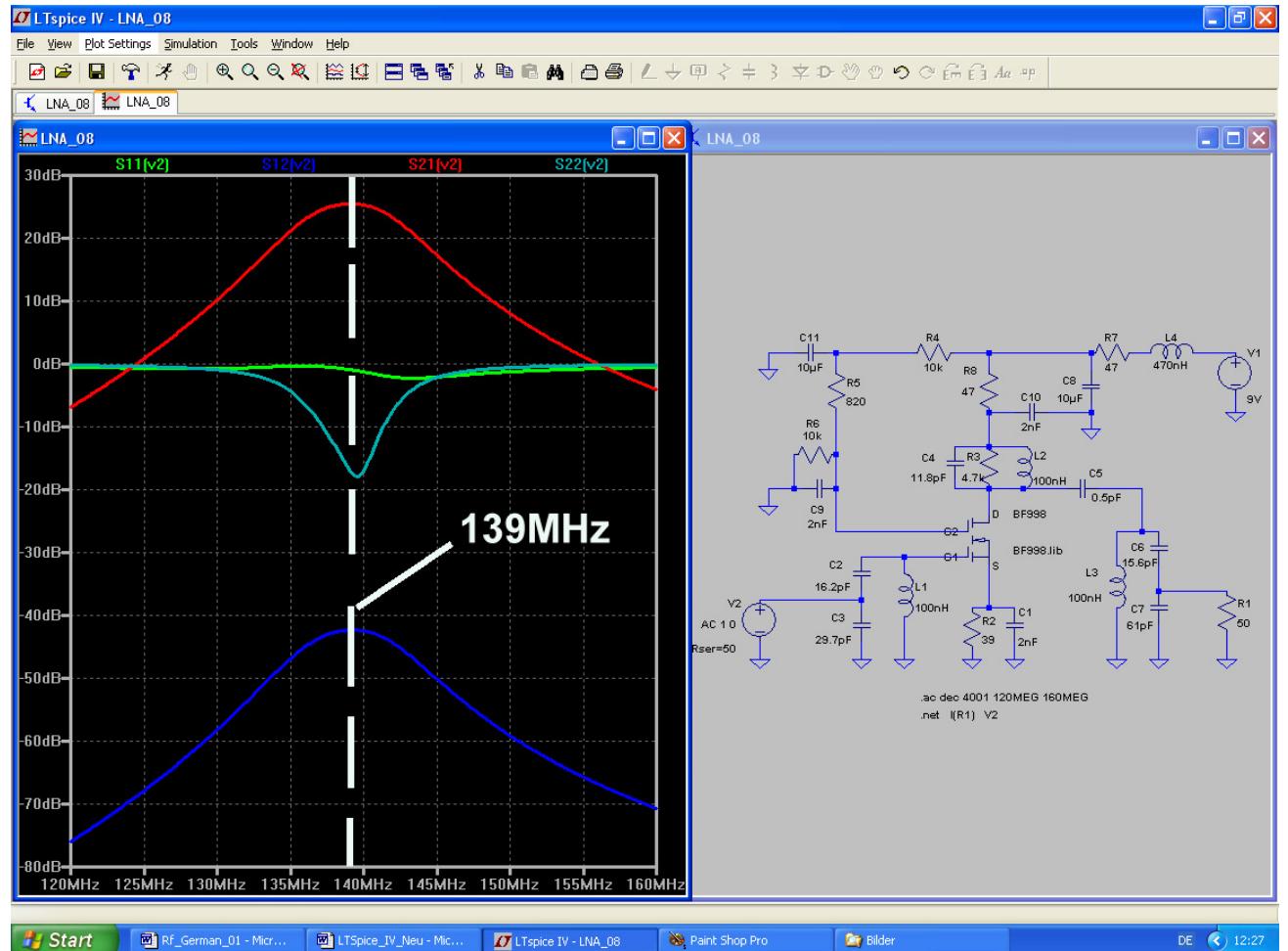
- Write a SPICE directive for an S parameter simulation:

```
.net I(R1) V2
```

R1 = 50Ω is the output termination of the stage. V2 is the input voltage source with the properties “AC Amplitude AC = 1“, “AC-Phase = 0“, source resistance = 50Ω.

Now simulate and don't show the phase. Use a linear frequency axis from 120MHz to 160MHz with a tick of 4MHz.

The amplitude range at the vertical axis should be +30dB to -80dB with a tick of 10dB.



This is the result and on the following page you find the original result publication from the article in the UKW Berichte journal (issue 1 / 1999).

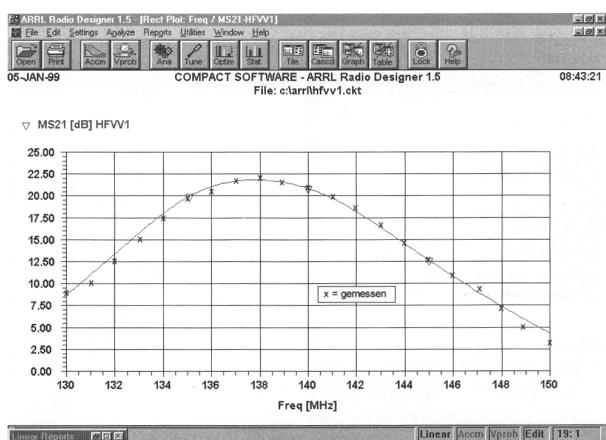


Bild 21: Vergleich zwischen Simulation und Messung

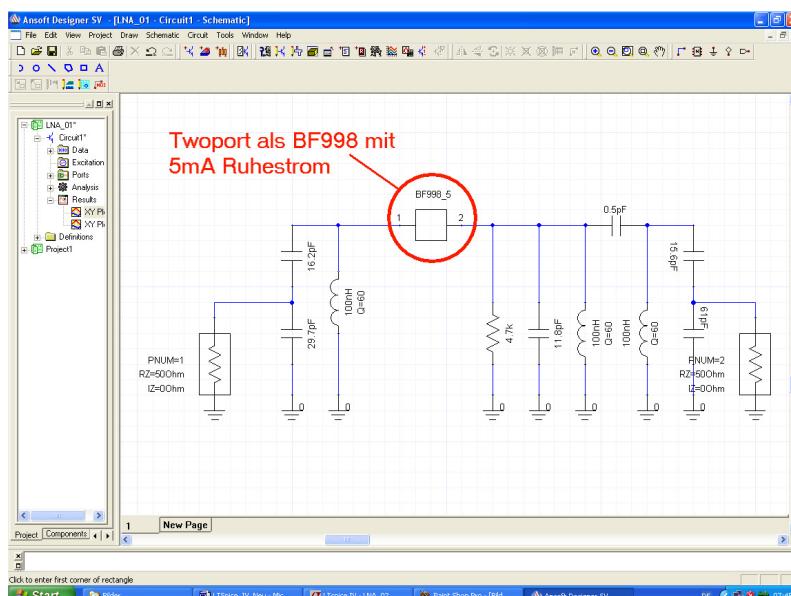
For this simulation the ARRL Radio Designer was used. The simulated S21 value was

“22.5dB at 137MHz”

and this is 2.5dB less but the LTspice simulation.

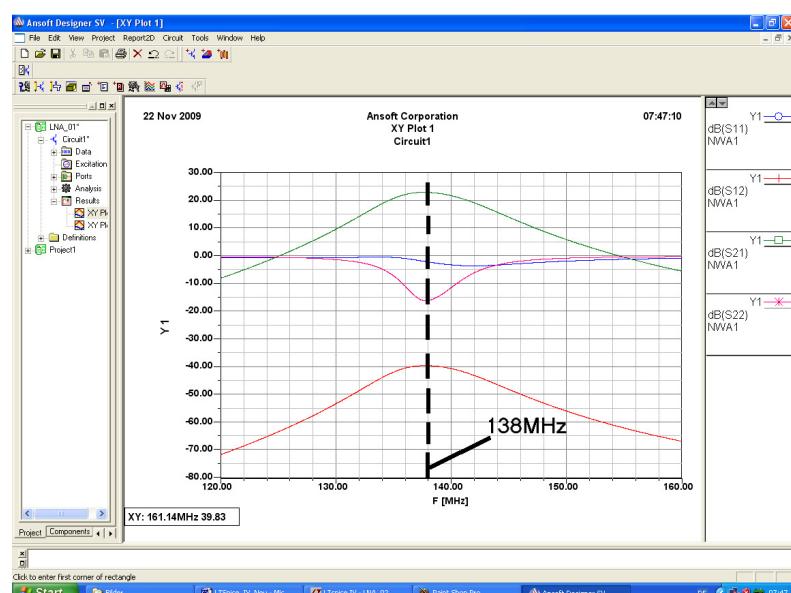
(The value of 22.5dB was confirmed by measurements).

The measured centre frequency was 138MHz
(SPICE says: 139MHz.)



So let us test the result with one of the best (and free!) modern S parameter simulation programs: the Ansoft Designer SV.

This is the drawn schematic with an S parameter file for $I_D = 5\text{mA}$



...and this is the result of the S parameter simulation.

Now it is really interesting to compare all:

Ansoft calculates a gain of 23dB at 138MHz (= 1MHz less than SPICE).

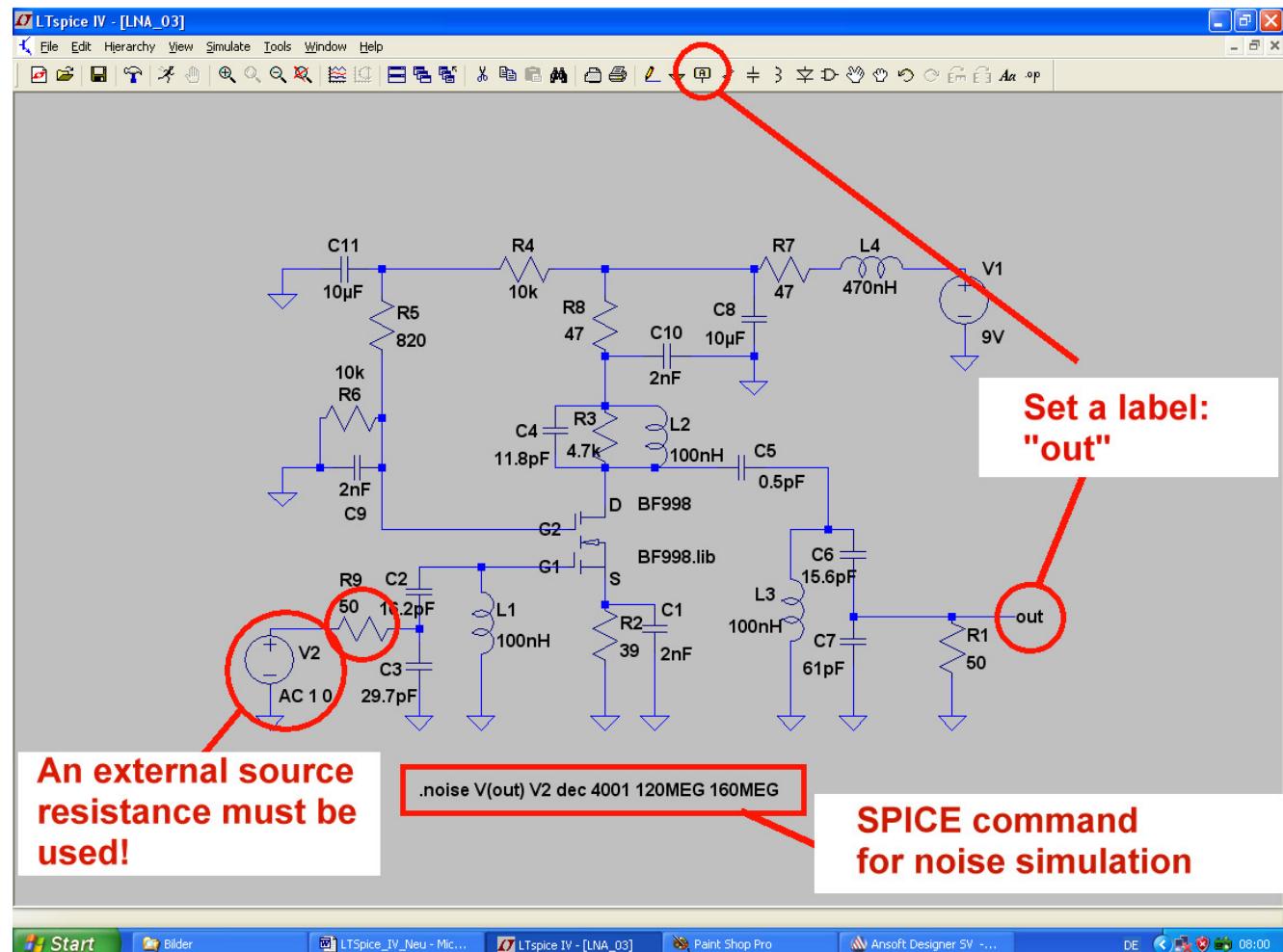
The little difference of the centre frequency is not a serious problem: for this purpose RF coils have cores which can be turned.

3.3.3. Noise figure simulation

To get a feeling for the values which can be expected, study the different datasheets for the noise figure value at a Drain current of 10mA (you will find published values of NF in dB only for this current).

The results are: **0.6dB (Siemens)**, **0.7dB (Philips)** and finally **1dB (Vishay)**. This seems to become an interesting story!

But at first we must modify the schematic and write a SPICE command:



The internal resistance of source V2 must be set to a value of **zero** and replaced by the **external resistor R9 = 50Ω**.

At the output termination R1 a label „**out**“ must be used (the symbol can be found beside the ground symbol in the menu bar).

Select “**Simulate**“ and “**Edit Simulation Command**“. Then write the following command for a noise simulation with a decade sweep from 120MHz to 160MHz (4001 points per decade):

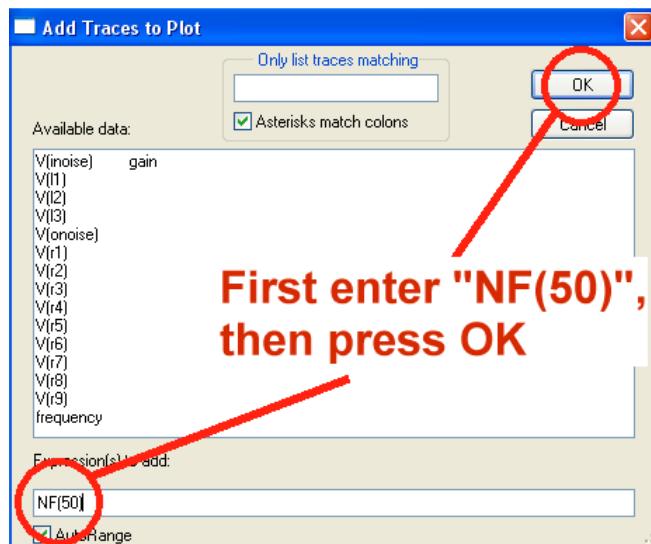
```
.noise V(out) V2 dec 4001 120MEG 160MEG
```

Now simulate and measure at R1 by left clicking on the connection. What you now get to see is the “**Spectral Noise Density**” at the output. If you are interested in the value of the Spectral Noise Density at the input: click on the connection between V2 and R9.

But for the calculation of the **Noise Figure NF** more effort is necessary. Please invoke a text editor and write the following line on an empty sheet:

```
.func NF(50) 10*log10(V(inoise)*V(inoise)/(4*k*300*50))
```

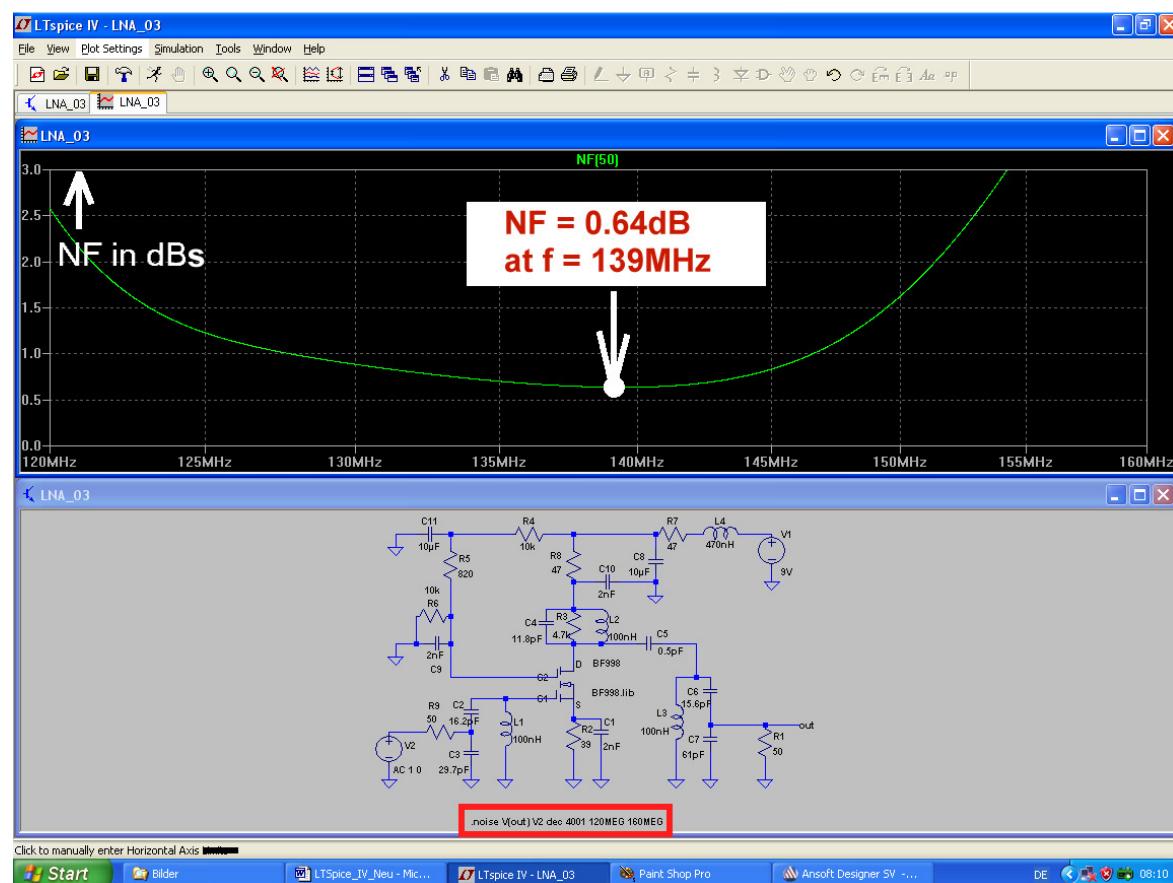
Save this file as **plot.defs** in the **Ltspice** folder and restart Ltspice
By this way the program knows this new function “**NF(50)**” to plot the noise figure of a 50Ω system at a temperature of 300K using the unit “dB”.



And now:

- Simulate.
- After the simulation the wave form viewer presents an empty diagram. Right click on it and open the “Add Trace” menu. Enter “**NF(50)**” in the empty line and press OK. The result is the desired noise figure NF in dB.

But it is a pity that in this case no units can be added to the vertical axis..



Please compare the result to the NF values given in the different datasheets at the top of this chapter. A very nice result, indeed...

3.3.4. Stability Control

Good RF simulation software must have this option -- e. g. by calculating the stability factor "k" which is based on the S parameters.

The S parameter simulation for the LNA is already done and so we have to write a function for this purpose.
In the literature [2] we find the following formula:

$$k = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |S_{11} \cdot S_{22} - S_{12} \cdot S_{21}|^2}{2 \cdot |S_{12}| \cdot |S_{21}|}$$

So let us open the file ".plot.defs" (...do you rememeber: we had to create it to calculate the noise figure...) and transfer the formula to a form which can be understood by LTspice.

Caution:

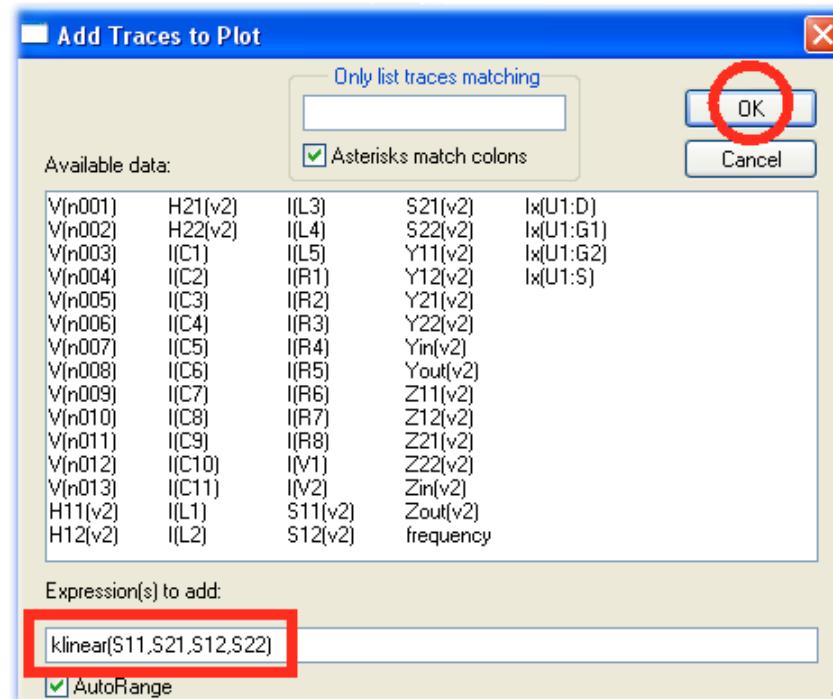
The letter "k" is already reserved by SPICE for other purposes! So we use "klinear".

This is the formula transfer result and it must be saved in the file "plot.defs", which is located in the LTspice folder:

```
.func klinear(S11,S21,S12,S22)
{(1-abs(S11(v2))*abs(S11(v2))-abs(S22(v2))*(S22(v2))+abs(S11(v2)*S22(v2)-
S12(v2)*S21(v2))*abs(S11(v2)*S22(v2)-S12(v2)*S21(v2)))/(2*abs(S12(v2))*abs(S21(v2)))}
```

But do not forget the program restart after saving this file.

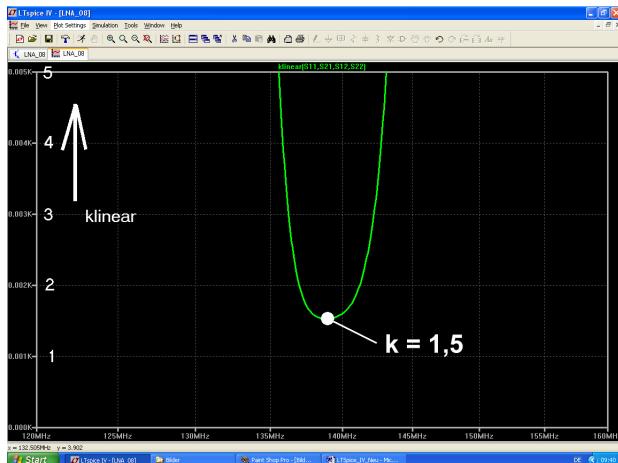
Then simulate again the S parameters and right click on the wave form viewer. Select „Add Plot Pane“ to get a new empty diagram.



Right click once more and select
“ADD TRACE”. Enter

„klinear(S11,S21,S12,S22)“

in the empty field and press OK.



Change the vertical axis settings **from k = 0 to k = 5** and you get a **minimum value of k = 1.5 at 139MHz**.

(Use the option “plot curves with thick lines”. Therefore select the “Control Panel” = the button with the hammer. Go to “Waveforms” and you’ll find what you want).

Using the Ansoft Designer SV for this calculation you the nearly the similar curve but with a minimum value of k = 1.0 at 139MHz.

(For practical application: in this case add a little series inductor of 1.5...2nH to the Source to improve the stability factor to 1.5...2).

3.3.5. Simulation in the time domain using a 137MHz input signal

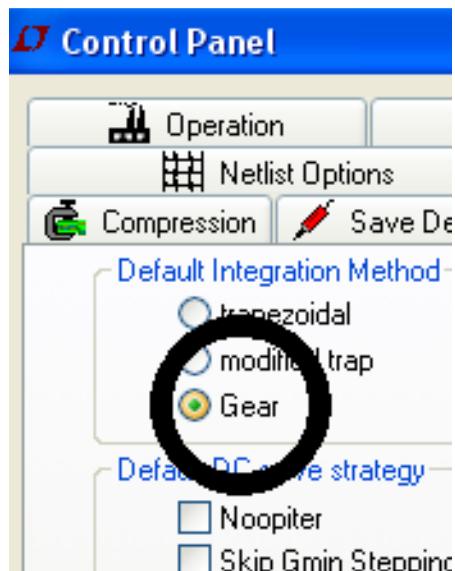
All our simulations were done in the frequency domain and S parameters could be easier calculated by the mentioned special RF CAD program.

But the most important feature of SPICE is the time domain calculation. So let us test the power by applying a Sine signal ($f = 137\text{MHz}$, peak value = 30mV) to the input and show the RF voltages in the circuit.

You need a sine voltage source ($f = 137\text{MHz}$, peak value = 30mV) and the following simulation command for the first 200ns using a maximum time step of 0.1ns

```
.tran 0 200ns 0 0.1n
```

Delete all other directives and commands from the screen.



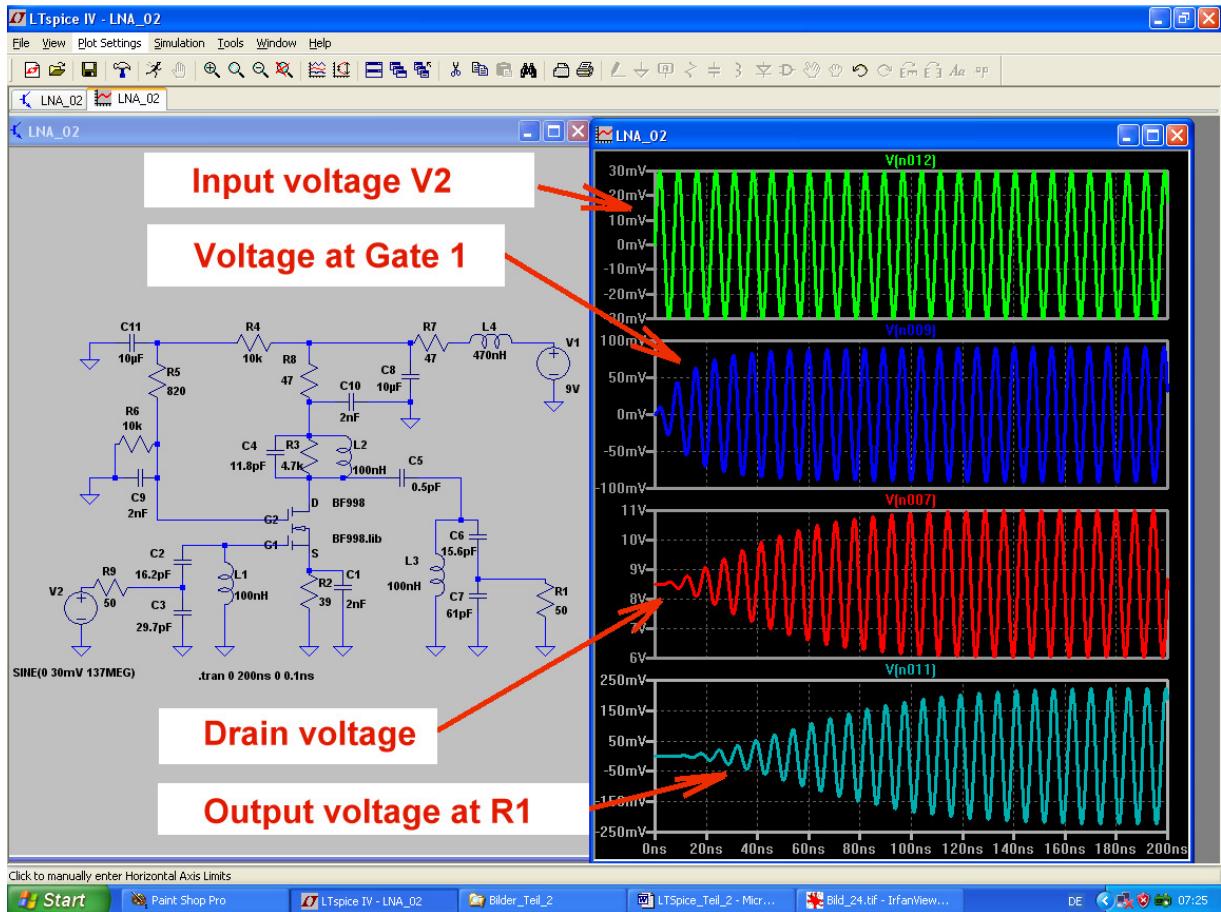
Caution:

Press the “Control Panel Button” (= button with the hammer) and select “SPICE”. Always use “Gear” as Integration Method -- otherwise you get serious problems when simulating RF circuits!

But that is not the end:

Check this setting as often as possible, because LTspice changes sometimes without any announcement when

modifying the circuit or part values or settings or simulation commands...



These are the four RF signals which can be measured in the circuit. But be aware that the input voltage on top of the diagram is not the incident wave. Shown is the source voltage with twice the amplitude of the incident wave.

3.3.6. IP3 point simulation

With the simulation of the “**Third Order Intercept Point**“ we get informations about the distortions when increasing the input voltage level. At this IP3 point the distortions would theoretically have the same amplitude as the applied input signal (...which is not possible in practice, because ca. 10dB earlier you run into “compression” and amplitude limiting).

The IP3 point determination is done by applying two “inband input signals with same amplitudes”. There is only a little frequency difference between them and they are located at the centre of the passband.

The level must be high enough to produce measurable intermodulation but no compression – in practice between -30dBm and -10dBm (RMS values between 7mV and 70mV).

To simulate this intermodulation we connect two voltage sources at the input of the LNA in series and enter the following properties:

**Source V2 is a Sine wave with a peak value of 20mV and a frequency of f = 137MHz.
Source V3 is a Sine wave with a peak value of 20mV and a frequency of f = 137.5MHz.**

Caution:

There are fundamental laws of the FFT (= Fast Fourier Transform) when calculating a frequency spectrum which never must be injured or respected:

Simulation time in the Time Domain and frequency resolution of the calculated spectrum is bound by the relationship

$$\text{Frequency resolution} = 1 / (\text{Simulation Time})$$

If you ask for a frequency resolution of 50 kHz, then the necessary simulation time can be calculated to

$$t_{\text{Simulation}} = \frac{1}{50\text{kHz}} = 20\mu\text{s}$$

The “**Maximum Time Step**“ is the time difference between two simulation points in the diagram and determines the minimum sample frequency to

$$f_{\text{sample}} = 1 / \text{Maximum Time Step}$$

This frequency value must always be higher than twice the maximum spectral component in the information (= Shannon law). Otherwise you get “Aliasing distortions” which cannot be filtered out or eliminated. Always avoid this terrible case!

The highest frequency in the calculated and displayed spectrum is determined by this sampling frequency. But to get a correct solution another law must not be injured, because the FFT uses a definite number of samples for the necessary calculations. This “number of samples” must be entered by the user in the menu before the FFT calculation is started (...and must always be a multiple of 2...).

But if you enter more samples for a FFT than calculated by LTspice during the Time Domain simulation, you get problems (= inaccuracy, a huge noise floor, endless calculation time, spectral lines which don't exist in reality....).

So always be aware that the

$$\text{Number of simulated samples} = \frac{\text{Simulation Time}}{\text{Maximum Time Step}}$$

is always higher than the number of samples used for the FFT calculation.

The default sample number value when starting the FFT option is always 65 536. We use 131 072 samples -- this will give a greater dynamic range and a higher maximum displayed frequency at the result.

So let us produce a collection of 200 000 samples due to a

$$\text{Maximum Time Step} = \text{Simulation Time} / 200\,000 = 20\mu\text{s} / 200\,000 = 100\text{ns}$$

At last, add a SPICE directive to your schematic to switch off the data compression which will automatically be applied by LTSpice to the Time Domain simulation results. Otherwise the number of 200 000 calculated samples is dramatically reduced and you get the mentioned problems (= number of samples to calculate the FFT would be higher than the amount of “realistic” samples).

Select “Edit” and “Spice directive” to write :

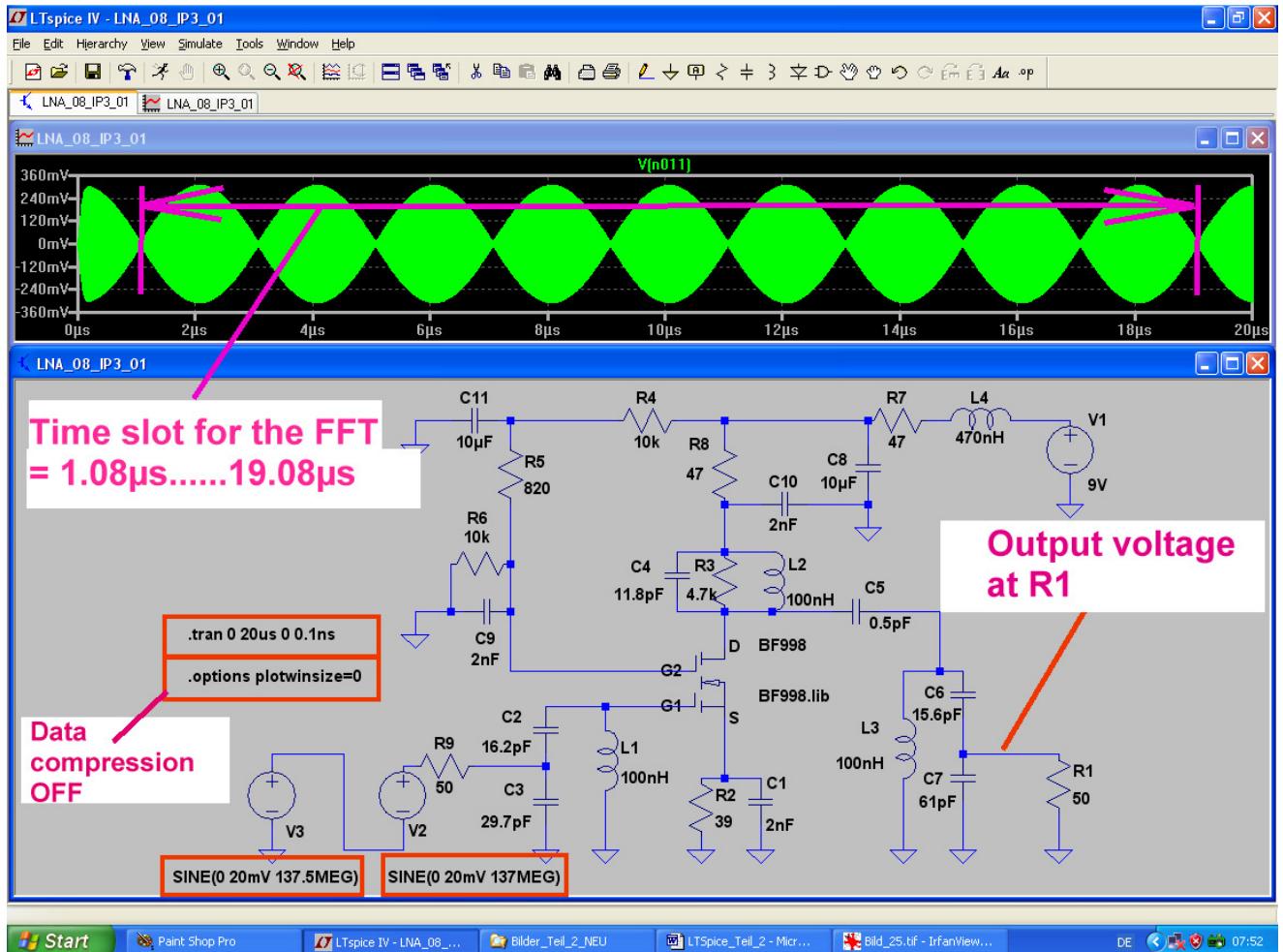
.options plotwinsize=0

and please check again whether “GEAR” is still active as integration method.

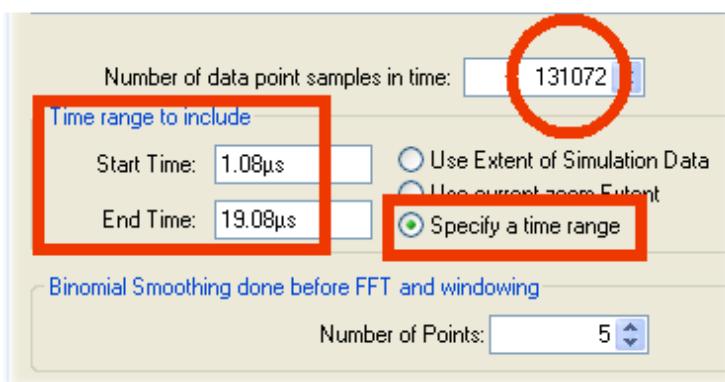
On the next page comes the schematic with commands and directives and the Time Domain simulation result. In a simulation it is easy to connect two voltage sources in series for addition of the two input signals, but in practice...see the information about the problems and procedures in [3].

In the diagram with the output voltage a “**time slot for the FFT**” is marked, because you must avoid anything that could have negative influence to a correct FFT result.

So the “**switch on and run up time**” of the circuit must be eliminated to get only “**steady state informations**” and the time slot is arranged between two “zero crossing points” of the curve.

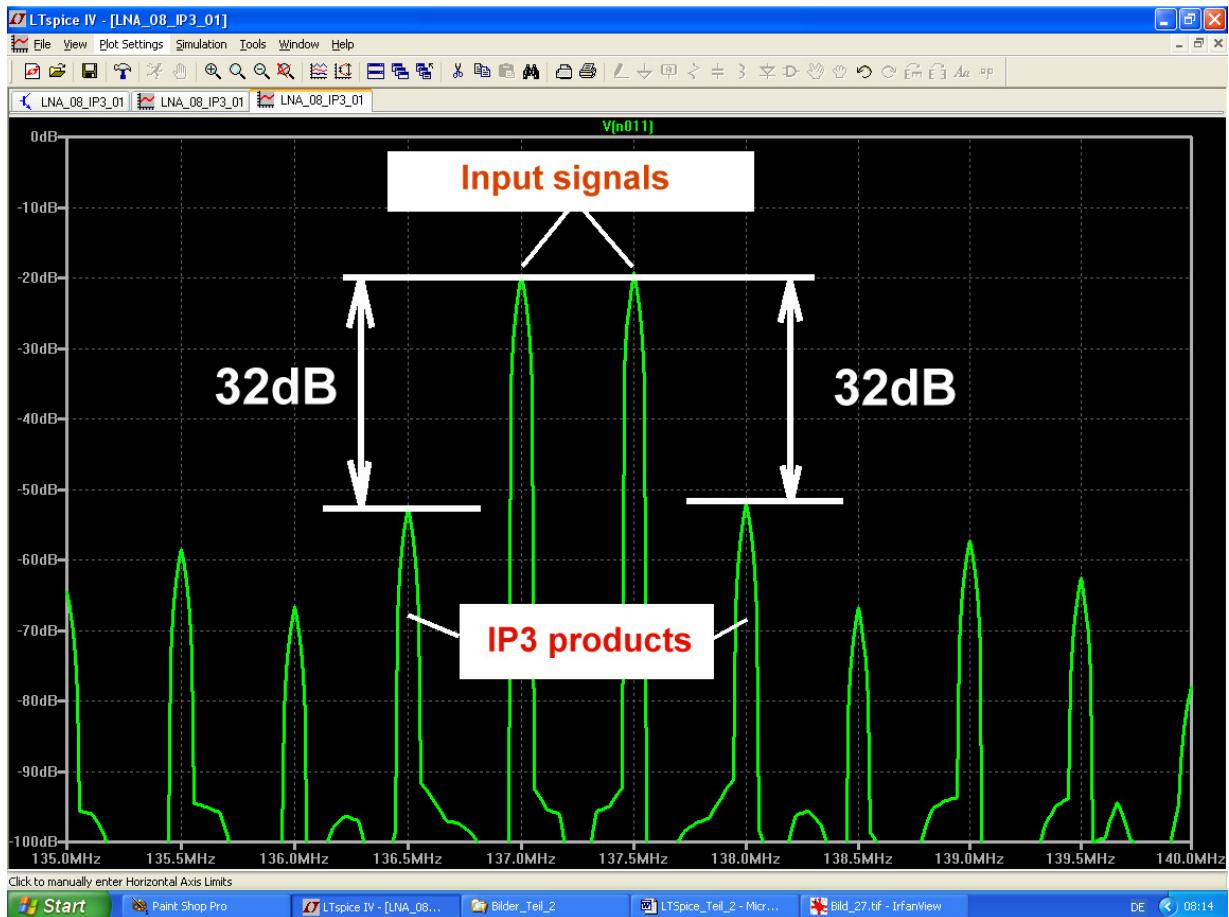


And now the FFT: right click on the diagram and select "VIEW", then "FFT"



These are the necessary entries. But at first enter the sample number of 131072 and afterwards specify the time range. Otherwise the program does not accept your wishes.

In the result diagram zoom the frequency range between 135...140MHz. You'll find the two input signals and the Third Order Intermodulation Products which are 32dB down.



Now calculate the IP3 value by the relation

$$\text{IP3 in dBm} = (\text{Input signal level}) + 0.5(\text{Level difference to IP3 products})$$

But you have the choice between the IP3 point referred to the input (**IIP3**) or referred to the output (**OIP3**). The difference is the gain of the stage.

For the input side:

With a source voltage of 20mV (peak value) we get an RMS value of 14.14mV. So the incident wave has an amplitude of 7.07mV and this gives a power level of

$$20 \cdot \log\left(\frac{7,07\text{mV}}{224\text{mV}}\right) = -30\text{dBm}$$

The rest of the procedure is simple because the IIP3 will be:

$$\text{IIP3} = -30\text{dBm} + 0.5(32\text{db}) = -30\text{dBm} + 16\text{dB} = -14\text{dBm}$$

With a voltage gain of ca. 23...24dB (See the simulation in the preceding chapters) we get

$$\text{OIP3} = \text{IIP3} + (23\ldots24\text{dB}) = -14\text{dBm} + (23\ldots24\text{db}) = +9\text{dBm}\ldots+10\text{dBm}$$

Such values can also be found in downloaded Application Notes for LNA's using Dual Gate MOSFETs.

That is it with the LNA – I think it was a lot of work, but interesting. Let us now continue with the mixer stage.

4. The Double Balanced Mixer (= ring modulator)

4.1. Fundamentals

This well known circuit is used to change the frequency of a signal without touching its information content – you can realize a frequency conversion.

There are two inputs and one output:

- The **information** which shall be converted is applied to the **RF Input** (= radio frequency input) with **small amplitude**.
 - The frequency conversion is caused by applying a signal with a **large amplitude** to the **LO Input** (= Local Oscillator Input).
 - At the **IF output** (= intermediate frequency output) we get the result of the frequency conversion.
- Wichtig:

Note:

The two applied signals are **multiplied** in the circuit and mathematic delivers the formula for this process if cosine signals are used:

$$\cos(\alpha) \cdot \cos(\beta) = \frac{1}{2} [\cos(\alpha + \beta) + \cos(\alpha - \beta)]$$

Regarding the right hand side of the formula we get information about the form of the result::

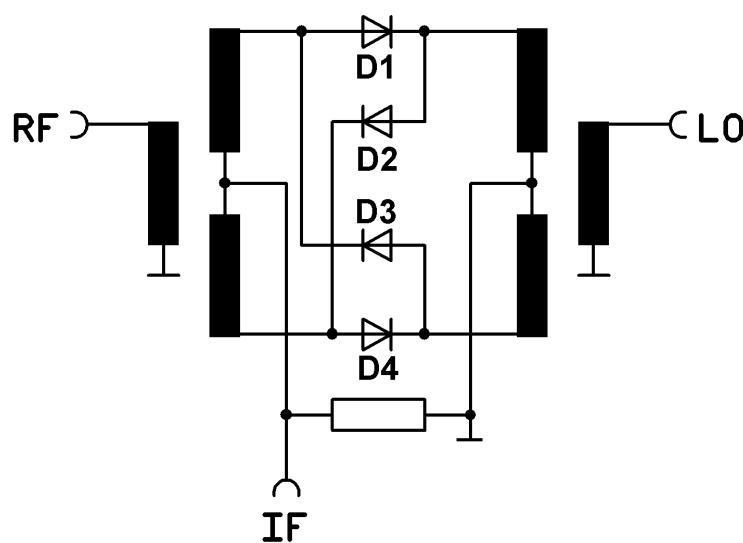
At the IF output the two input signals have disappeared. Instead of them we find the sum frequency and the difference frequency!

If a symmetric square wave signal is used as LO signal (or if the peak value of a sine wave as LO signal is very large -- let's say greater than 1..2V), then the following happens:

A symmetric square wave consists itself of the fundamental sine wave and "odd Harmonics" (= sine signals with frequencies which are odd multiples of the fundamental frequency).

Now not only the fundamental wave but also every harmonic is multiplied by the input RF signal and every multiplication process produces a signal pair – consisting of sum frequency and difference frequency.

The user will then use filters to suppress the undesired and to work only with the desired signals.



A DBM can be realized as an active or as a passive circuit. On the left the passive version can be seen. It uses four fast Schottky diodes, connected to a ring, and two transformers. Every transformer has three windings to deliver two output signals with the same amplitude but a phase shift of 180 degrees.

And so the multiplication process is realized:

The positive half of the LO signal switches on the two diodes D2 and D4 which are connected in series. Otherwise D1 and D3 will be switched on during the negative half of the LO signal.

By this way the IF output signal comes either from the lower secondary winding of the RF transformer (when D2 and D4 are conducting) or from the upper winding (when D1 and D3 are conducting). This

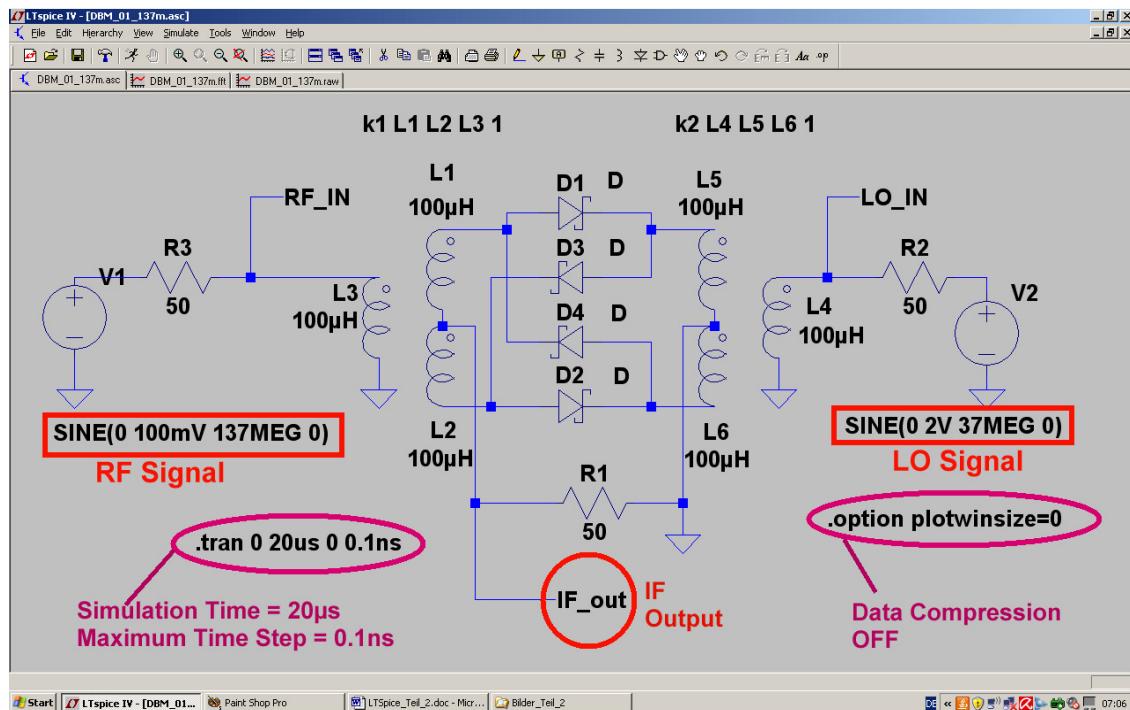
gives an alternating phase shift of 180 degrees at the IF port – caused by the LO signal.

Advantages: every low distortion when fast switching. Due to the construction of the transformers and the quality of the Schottky diodes usable to very high frequencies (...in special microstrip versions up to 50GHz.) Produced in high quantities – this gives low price. Needs no supply voltage but correct 50Ω termination at every port. Available with nearly every popular RF plug.

Disadvantages: the complete power for the diode switching must be delivered by the LO source. Very fast ON and OFF switching speed necessary for low distortions at the IF port. No gain – but attenuation of ca. 6dB and as a passive device a Noise Figure of this value (6dB).

4.2. Simulation of the DBM using LTspice

4.2.1. Simulation schematic



All needed parts can be found in the LTspice library. The name for the Schottky diode symbol is "schottky". But some information is necessary for the transformers:

We use three separated inductors which are "magnetically coupled". Select the symbol "**ind2**" with a little circle to mark the polarity of the inductor's voltage. A value of 100μH for the inductance will do the job down to frequencies near 1MHz.

Note:

You must write a SPICE directive for the magnetic coupling factor of the three windings for each transformer. For the left this would be the line

k1 L1 L2 L3 1

for a coupling of 100% and the command for the right transformer is similar.

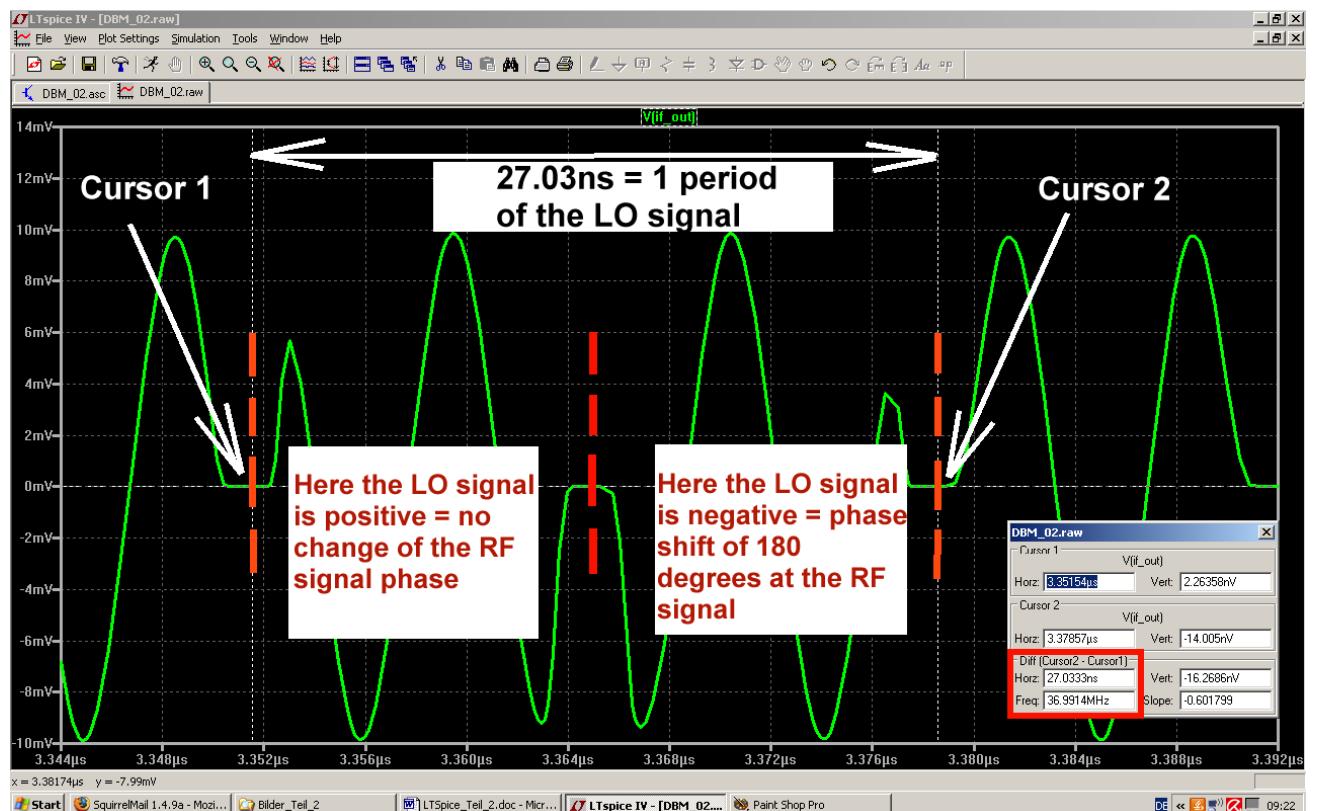
A sine signal with a peak value of 100mV and a frequency of 137MHz is applied to the RF input (like in practice when the antenna signal comes already amplified by more but 20dB from the LNA). A sine signal with a peak value of 2V is used as LO signal for fast switching of the diodes at a frequency of 37MHz. This will give a frequency difference of 100MHz which later must be filtered out as IF signal. Please use a label "**IF-out**" for the IF output and terminate it by 50Ω.

4.2.2. Simulation of the DBM properties

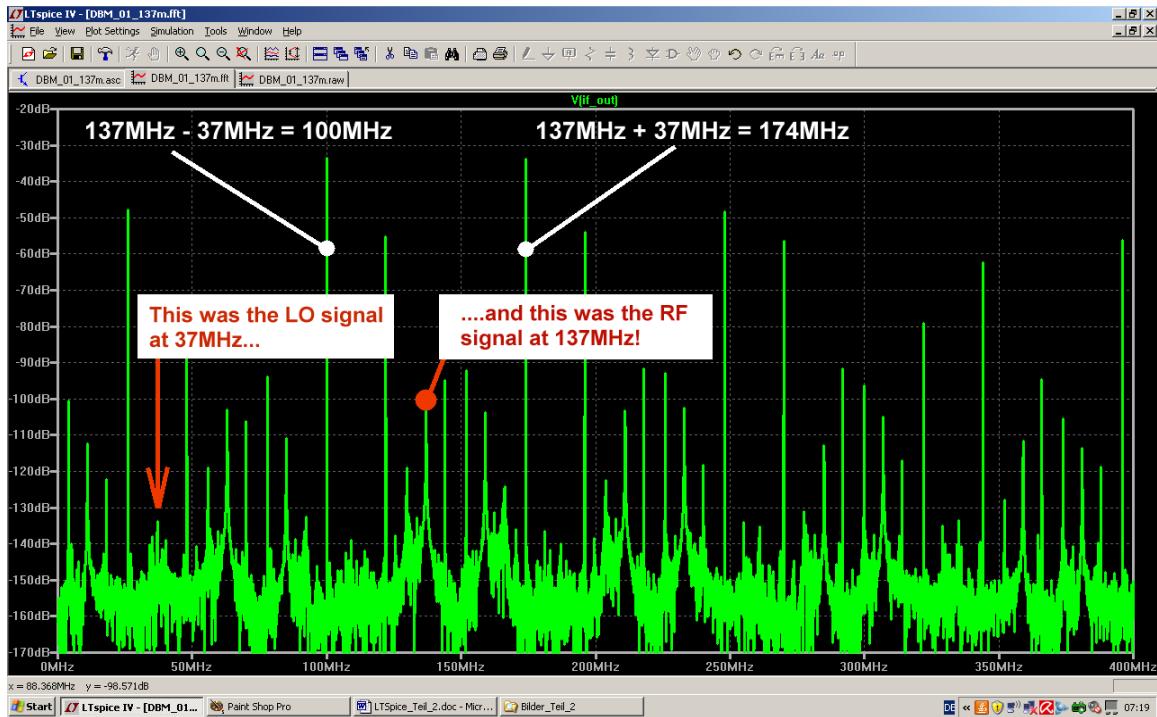
Simulation time is $20\mu s$ and this gives a frequency resolution of $1 / 20\mu s = 50\text{kHz}$ in the FFT spectrum. With a maximum time step of 0.1 ns we get a minimum of $20\mu s / 0.1\text{ns} = 200\,000$ calculated samples and so it is possible to choose the default value of 131 072 values for the FFT.

(For specialists: increasing the number of samples gives a higher maximum frequency in the simulated spectrum and a higher dynamic range. But the simulation time and the volume of the result file will also increase. Decrease will the noise floor which is quantization noise due to the Analog to Digital Conversion principle and another part which comes from LTspice itself. We have entered the maximum time step and this is equal to the minimum sampling frequency. But at “difficult corners of the curve” LTspice will increase the sampling rate to get a better result – and that is Frequency Modulation! If this happens regularly during the simulation of the curve it causes additional lines in the spectrum. And if it does not happen regularly, then we find additional noise FM).

The data compression must be switched off again by **.option plotwinspace=0** and the simulation can be started.

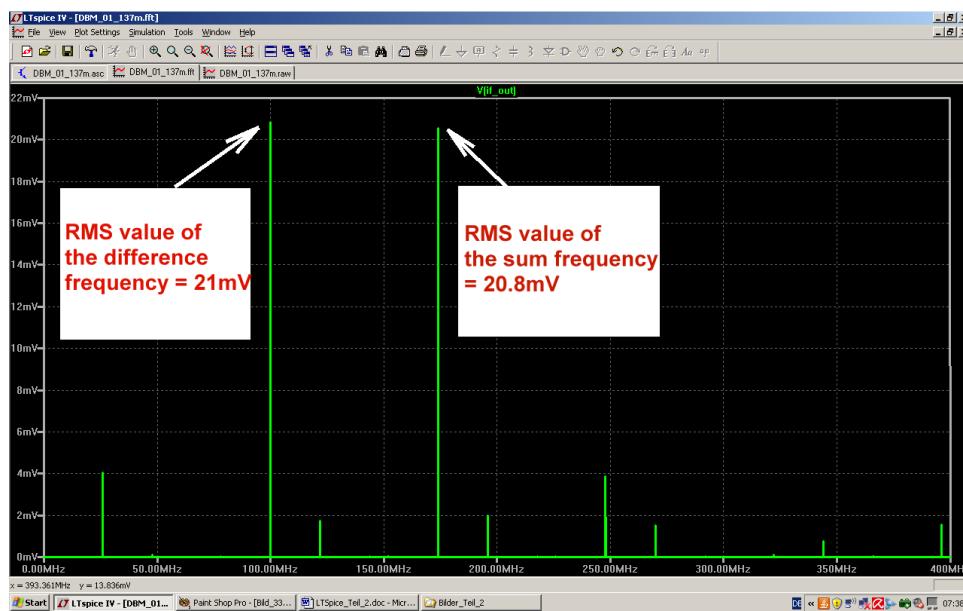


At this screenshot the principle of the DBM can clearly be seen. But do not forget to check the “GEAR” adjustment for the integration method before starting....



This is the result for a FFT with 131072 samples and a frequency range from 0....400MHz. Sum and difference frequency can easily be identified.

LO signal and RF signal are attenuated but should be completely suppressed. There exist also many new and unexpected lines, caused by the nonlinear region of the diodes when switching OFF and ON. At the zero crossing points of the LO signal the IF output signal will completely be missing for a short time, due to the threshold voltage of 0.3...0.4V for a Schottky diode – and this causes distortion.



With a linear presentation it is easy to determine the conversion loss.

For the sum frequency we find an RMS value of 20.8mV, for the difference frequency an RMS value of 21mV.

Now it is possible to calculate the conversion loss for the 100MHz signal:

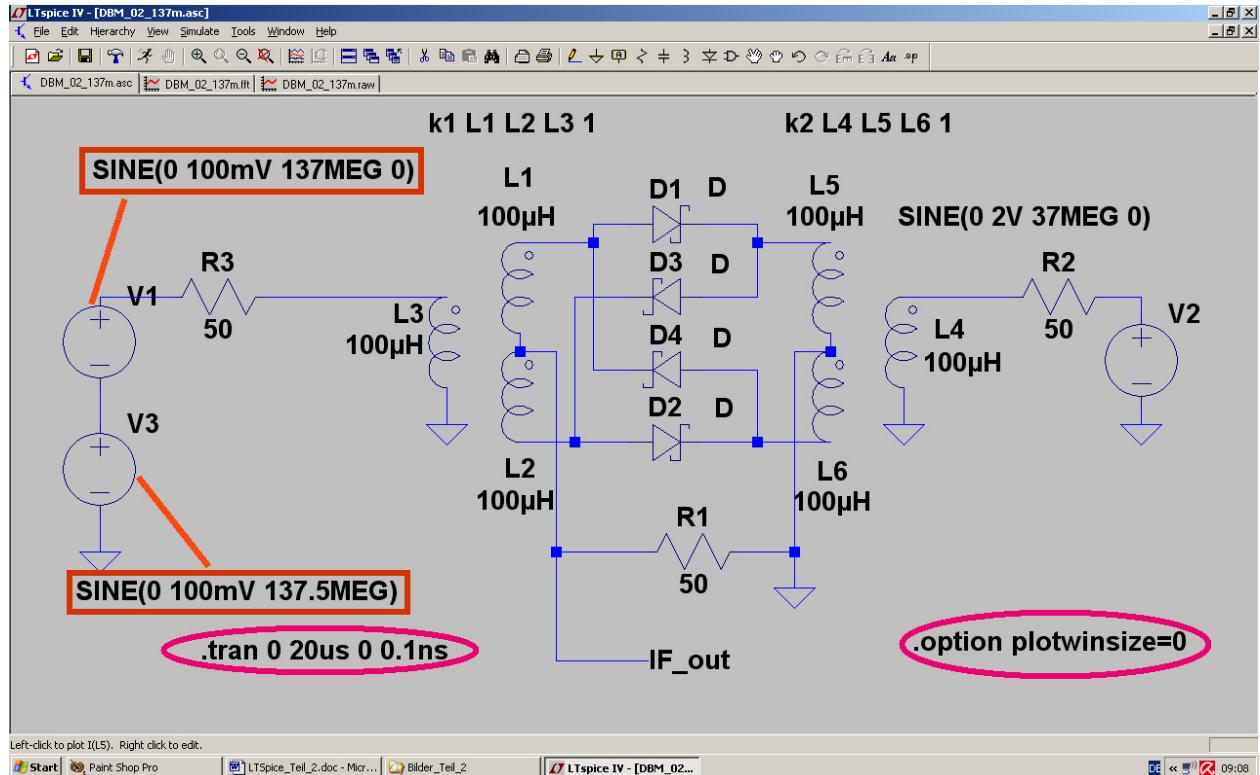
The peak value of the source voltage in the simulation was 100mV; this gives an RMS value of 70.7mV. The incident wave is half of this value = 35.35mV and appears at 100MHz with an RMS value of 21mV. So the conversion loss must be

$$a = 20 \cdot \log \frac{35.35\text{mV}}{21\text{mV}} = 4.5\text{dB}$$

In practice this value is higher due to transformer losses and we find very often 5.5....6dB.

And the amplitude of the sum frequency decreases with increasing frequency values because also the Schottky diodes show weakness at high frequencies.

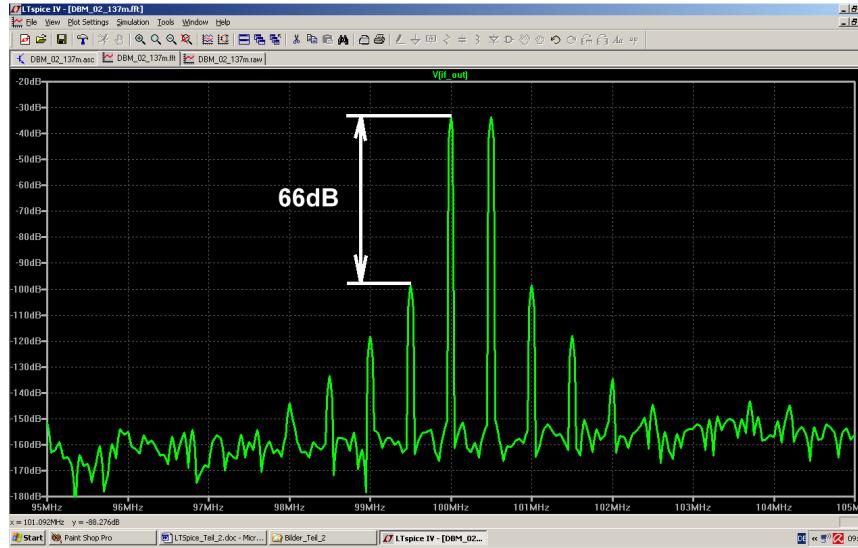
Now let us have a look at the Intermodulation. We feed the input with the well known "In Band signals" with a small frequency difference of 0.5MHz. The two signals are 137MHz and 137.5MHz, each with a peak value of 100mV.



Left-click to plot I(L5). Right click to edit.

Start Paint Shop Pro LTspice_Teil_2.doc - Micr... Bilder_Teil_2 LTspice IV - [DBM_02...]

09:08



At the output the IP3 products are 66dB down, but at first the amplitude of the incident wave must be calculated.

A peak value of 100mV gives an RMS value of 70.7mV and so the incident wave (with a half of this value) is 35.35mV. This gives a Power Level of

$$20 \cdot \log \frac{35.35\text{mV}}{224\text{mV}} = -16\text{dBm}$$

and an IP3 point of

$$\text{IP3} = (\text{Input Level}) + 0.5 \cdot (\text{IM Level difference}) = -16\text{dBm} + 0.5 \cdot 66\text{dB} = +17\text{dBm}$$

The Output IP3 value of the LNA was only +10dBm and so the LNA will earlier start with compression and distortion than the mixer.

5. Band pass filter at the DBM IF output

5.1. Notes

If we regard the simulated output spectrum of the DBM then a high quality band pass filter must be used to filter out the desired IF and to suppress all undesired signals. The bandwidth (at a centre frequency of 100MHz) should not exceed 5MHz and so a high quality factor of the used parts is necessary. A very good choice for this purpose is the "Narrow Band Pass" (..Ansoft calls it "Coupled Resonator Filter") with the following characteristics and properties:

- a) All used inductors have the same value (100nH at 100MHz is a good choice!).
- b) The coupling capacitors are realized as "Microstrip Interdigital Version" and so no discrete parts are necessary and must be soldered.
- c) The bandwidth should be 5MHz but 10MHz beside the centre frequency a stop band attenuation of 70dB should be possible.
- d) The attenuation up to 1 GHz should be better than 70dB.
- e) System resistance is 50Ω .

The design of such a filter is today very easy when using the free Microwave CAD software "Ansoft Designer SV". And these are specification for the filter to be designed:

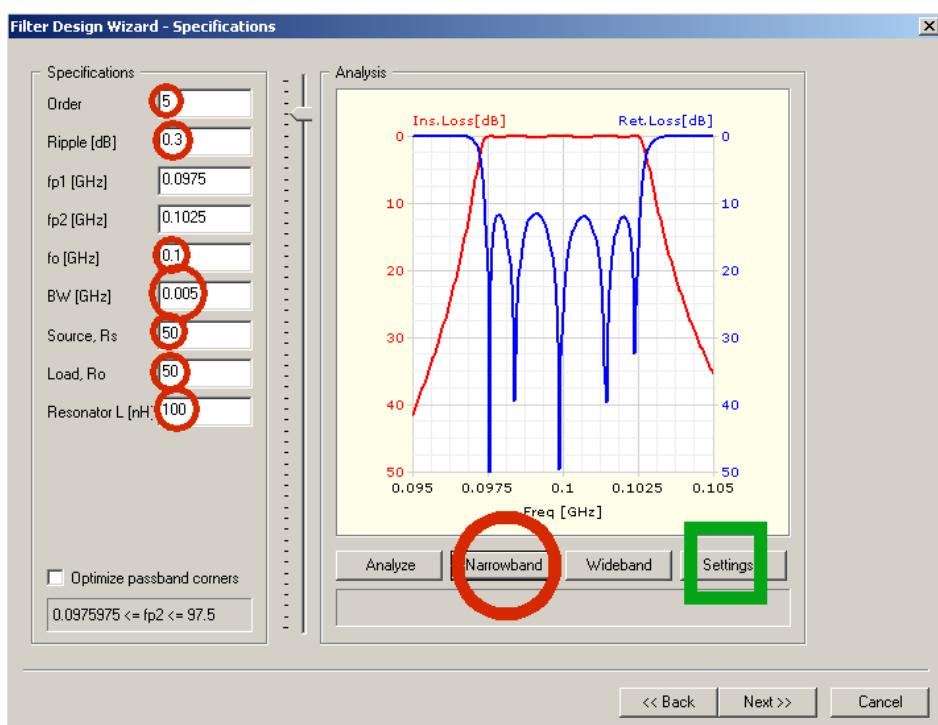
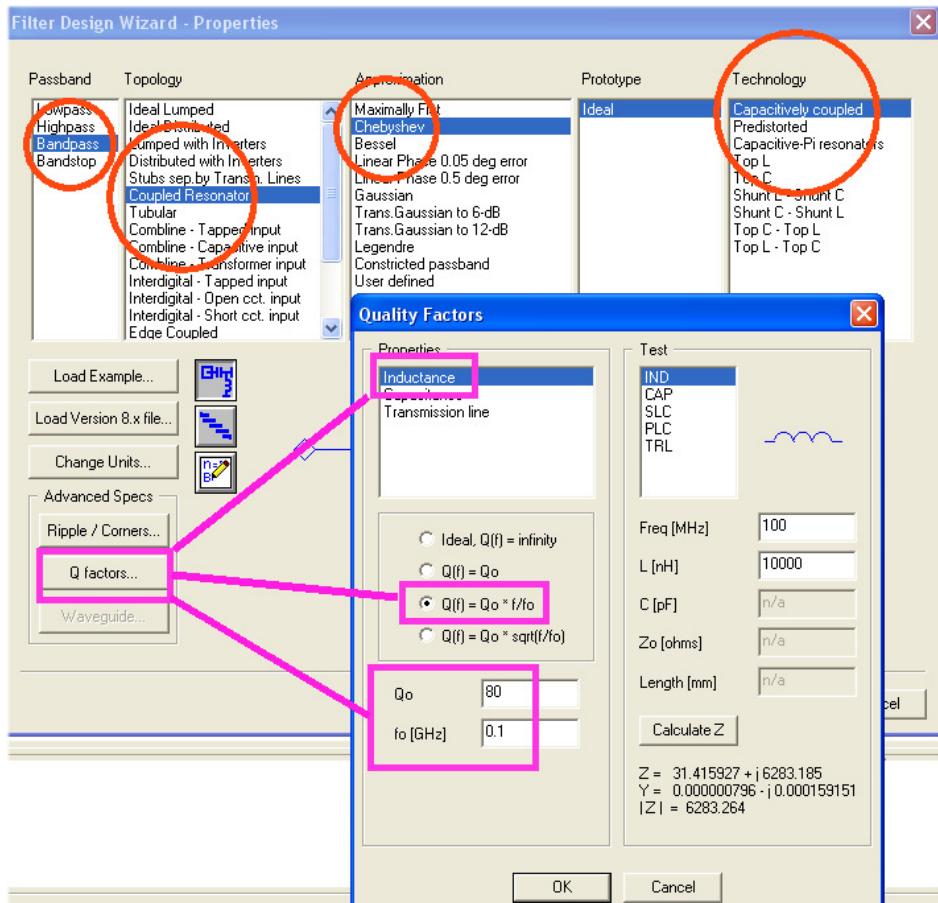
Type of filter	Chebychev BPF, Coupled Resonator Type
System resistance	$Z = 50\Omega$
Filter degree	n = 5
Centre frequency	100MHz
Pass band ripple	0.3dB
Ripple cut-off frequency	5MHz
Capacitors	SMD 0804 made from NP0
Inductors	Neosid helix filter in a silver coated Alumina case. $L = 100nH$, $Q = 80$ at 100MHz)
Coupling capacitors	Printed Interdigital version
PCB material	Rogers RO4003, thickness = 32MIL = 0.813mm, copper coating of $35\mu m$, $\epsilon_r = 3,38$

5.2. Filter design using the Ansoft Designer SV

At first congratulations to Ansoft for this excellent and free software. Not only the complete S parameter simulation including hundreds of models is free – it exists also an excellent free filter calculator. The Interdigital capacitors can also be found in the library.

Now start the program, select "Project" and then "Insert Filter Design".

Enter the desired filter type (**Band pass , coupled resonator , Chebychev , capacitively coupled**) and press the "Q Factor" button (**See next page**). Enter a Q factor of $Q = 80$ at 100MHz with linear relation between Q and frequency (**See next page**).



frequency and the bandwidth. Click “Narrowband” to see the S parameters S11 and S21 for a lossless circuit.

Now continue and you have to enter all filter properties

Filter degree: 5

Ripple: 0.3dB

Centre frequency f_o = 0.1GHz

Band width BW = 0.005GHz

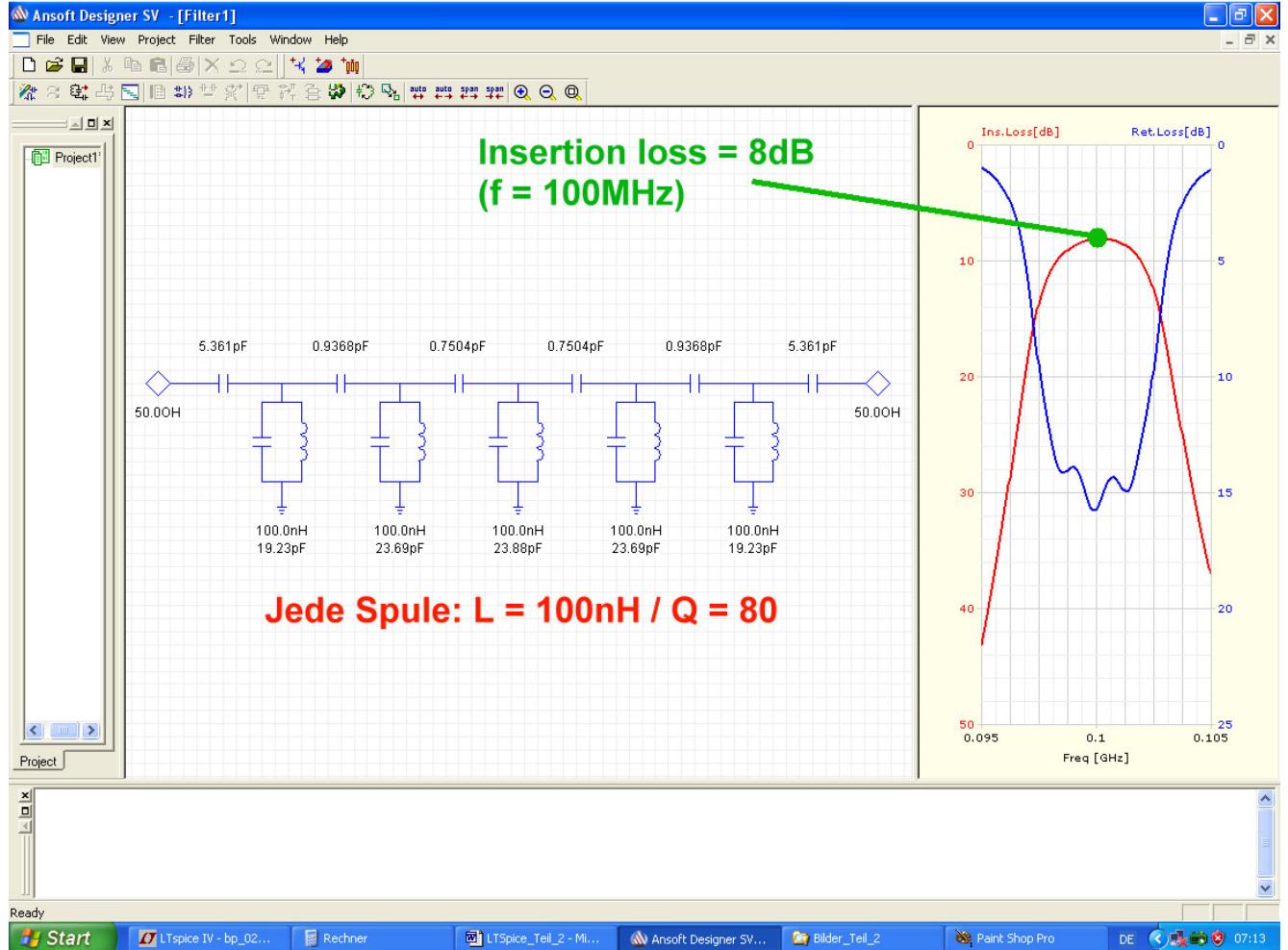
Source resistor R_s = 50Ω

Load resistor R_o = 50Ω

Resonator L = 100nH

The both corner frequencies are automatically calculated from the centre

To include the losses in the calculation select “Settings” and “Include Q factor losses in the response” terminate the calculation procedure. Now you get the S11 and S22 curves for the filter with losses. These inductor losses cause an Insertion loss of 8dB.



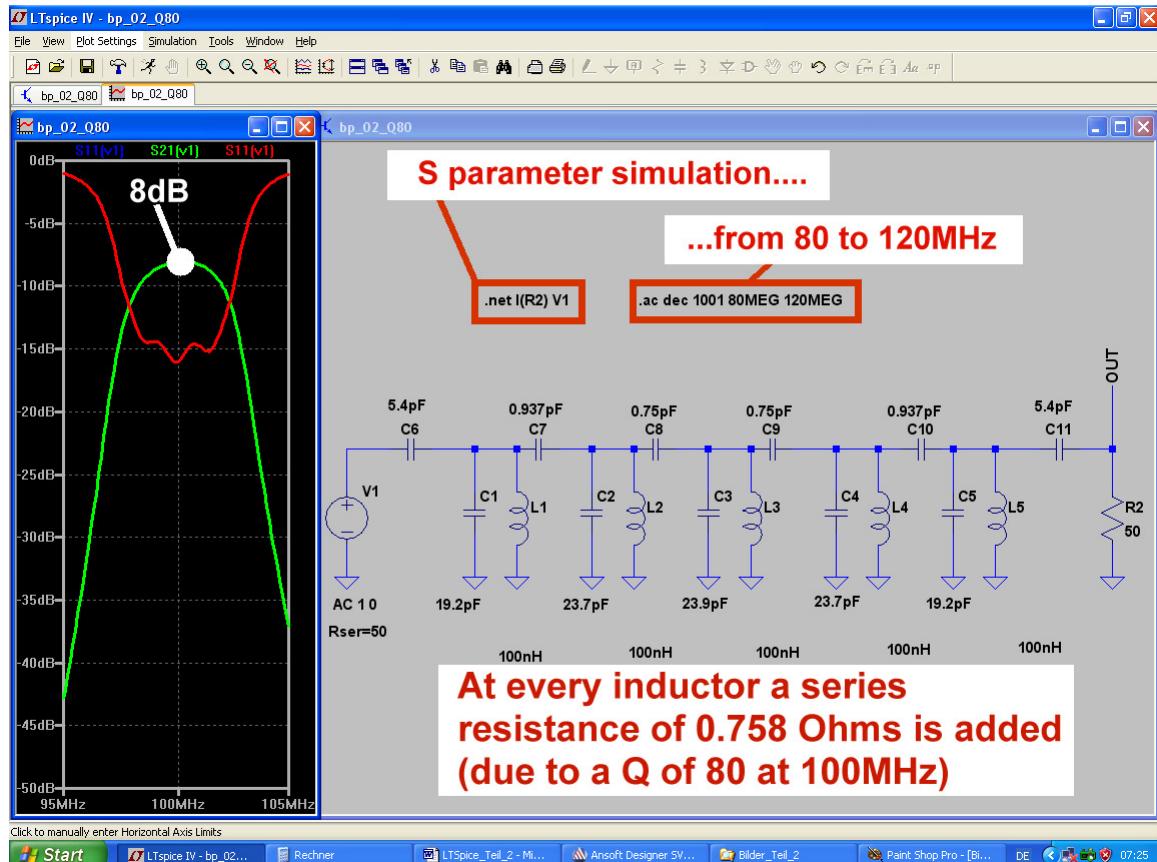
Now let us test the opinion of LTspice to this lossy filter.

Draw the schematic and start an S parameter simulation in the same frequency range. Include the losses as a series resistance of

$$R_{\text{series}} = \frac{2\pi \cdot f \cdot L}{Q} = \frac{2\pi \cdot 100\text{MHz} \cdot 100\text{nH}}{80} = 0.758\Omega$$

at every coil.

The schematic and the simulation result can be found on the next page.



Compared to the Ansoft calculation this result is really nice and OK.
The capacitance values have been rounded that (if necessary and possible) several SMD capacitors can be used in parallel for the practical realization of the filter.



Here comes a look at the finished filter in a milled alumina case.

The Interdigital capacitors can be well identified.

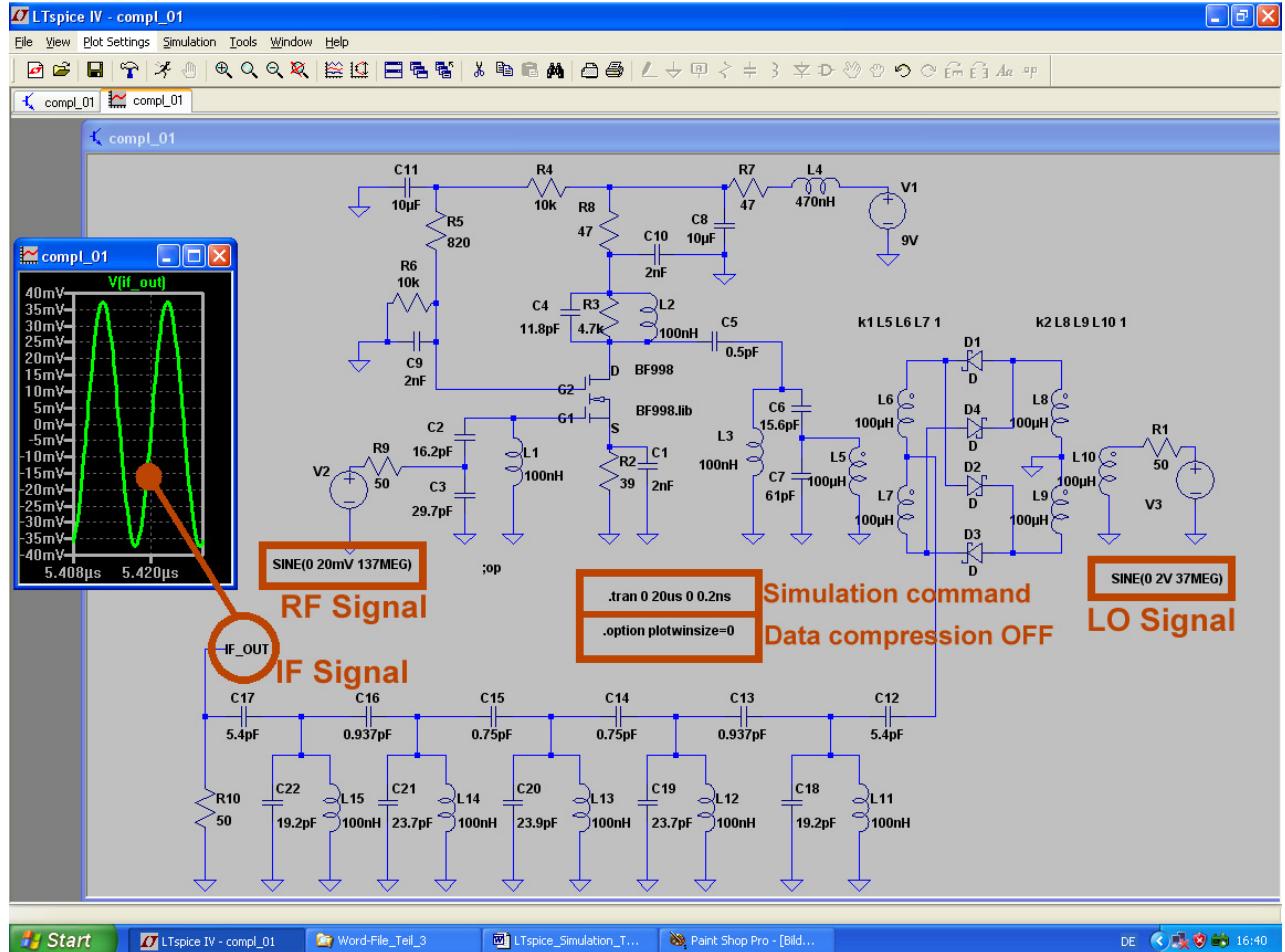
Every resonant circuit consisting of the shielded coil and 2 SMD capacitors is arranged on a small “ground island” which is connected to the ground plane on the under side of the PCB by a lot of vias.

6. Simulation of the complete converter circuit

6.1. Preparations

This is a lot of work, because the complete schematic (consisting of LNA, mixer and band pass filter) must be drawn with the LTspice editor. It is the best to start with the LNA project, to save it under a new name and then to draw the rest.

For the output the label "IF_OUT" is used. To the RF input a sine wave with a peak value of 20mV and a frequency of 137MHz is applied. As a LO signal also a sine wave (peak value = 2V, frequency = 37MHz) is used. The supply voltage is +9V.



Simulation time is 20 μ s with a maximum time step of 0.1ns and this gives 200 μ s / 0.1ns = 200 000 real samples, if the data compression is switched OFF. But remember:

Do not forget to press the button with the hammer (= Control Panel) and to check whether GEAR as integration method is activated. Otherwise the simulation will fail at these high frequencies, causing an infinite simulation time, inaccurate curves or an abort of the simulation.

Please repeat this check as often as possible – e.g. if you modify the circuit or if you add something new! LTspice changes sometimes this option without any announcement.

6.2. Gain at centre frequency

Please look at the output voltage curve at the label "IF_out" in the schematic (preceding chapter) and you get a peak value of 37,5mV. The peak value of the Rf source voltage was 20mV corresponding to an incident wave with a peak value of 10mV.

This gives an S21 value of the complete converter of $37,5\text{mV} / 10\text{mV} = 3.75 (= 11,5\text{dB})$.

But you should also estimate this value by a simple addition:

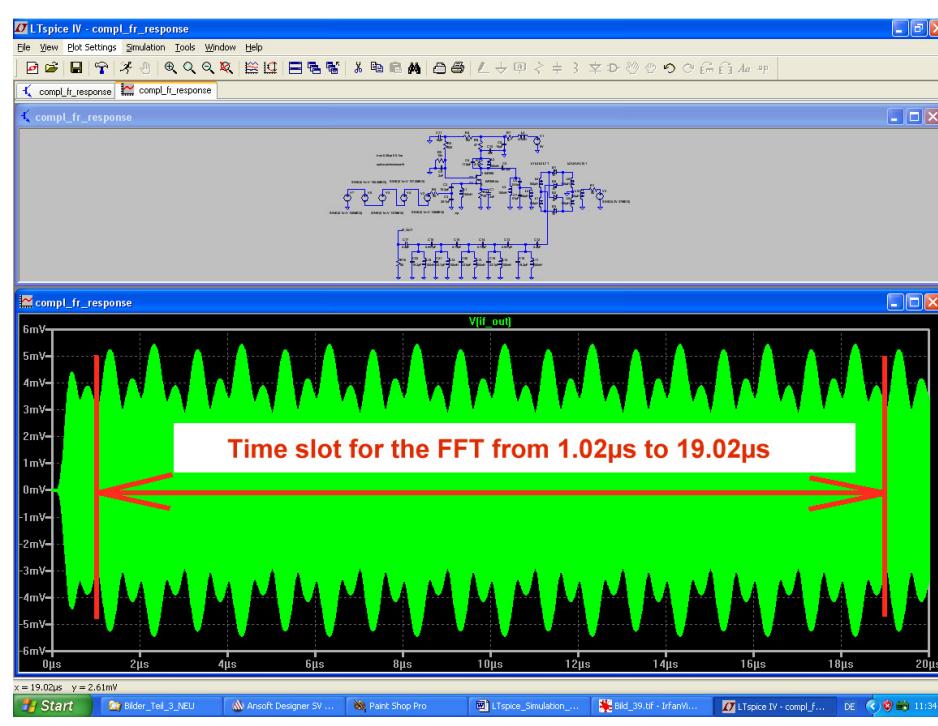
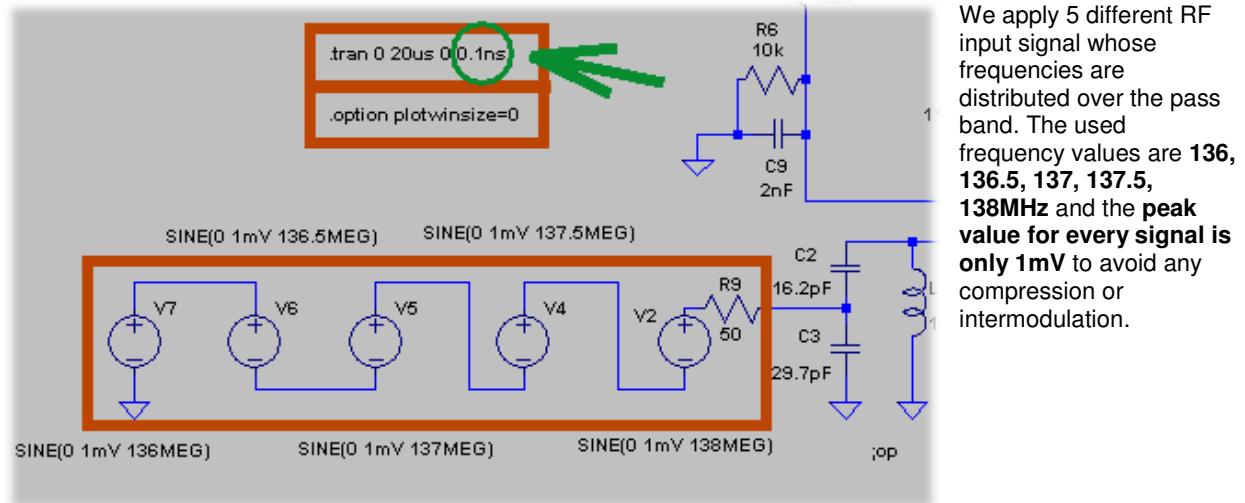
S21 in dB = LNA gain – mixer attenuation – filter attenuation = 25dB – 4.5dB – 8dB = 12.5dB.

The difference of 1dB is due to mismatching between the stages. But if you take into account the losses of the non ideal transformers this difference will still increase by ca. 2...3dB.

6.3. Pass band curve simulation

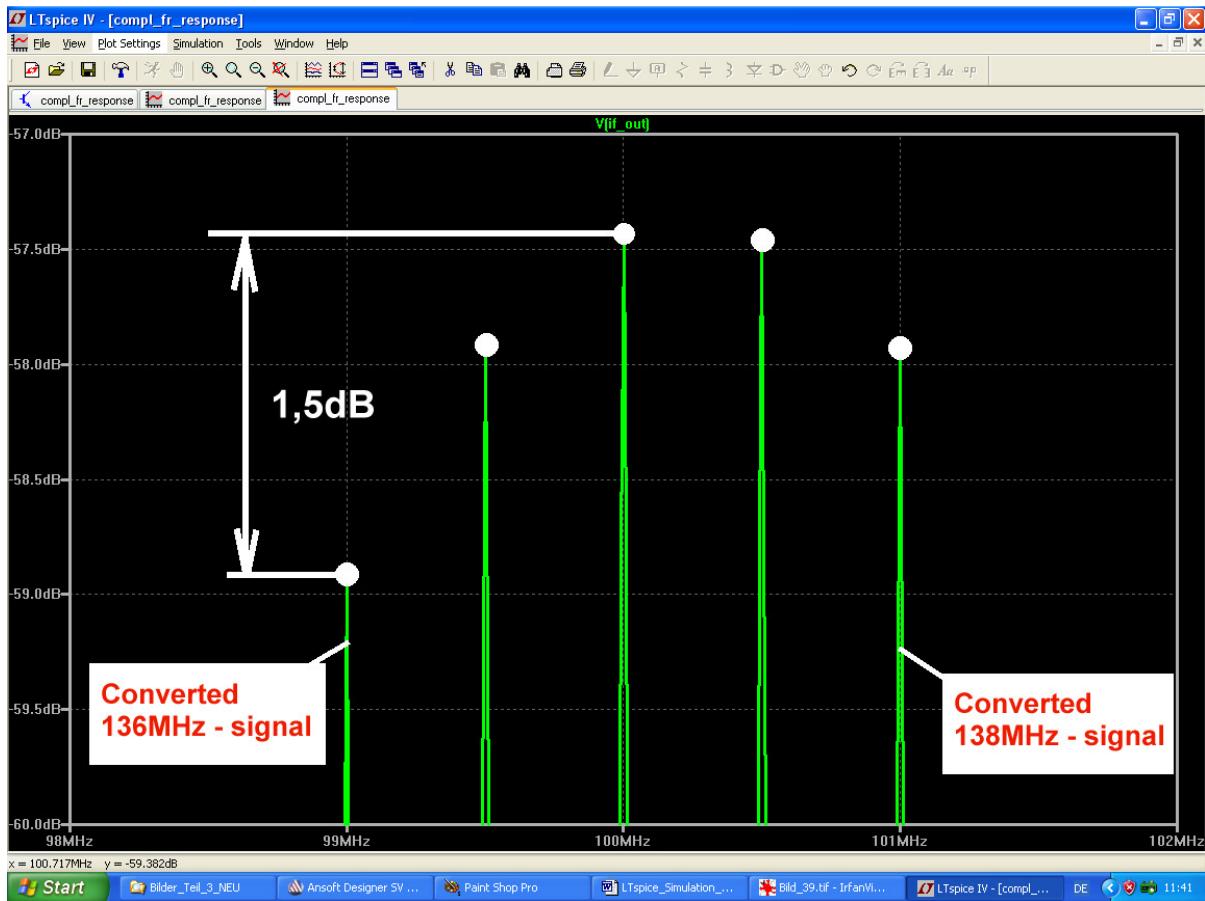
6.3.1. Simulation using separate spectral lines

Not so easy because the AC sweep fails completely – because there is a frequency conversion caused by the mixer in the circuit! So we think out another solution for the Time Domain.



Using a simulation time of $20\mu\text{s}$ and a maximum time step of 0.1ns gives $20\mu\text{s} / 0.1\text{ns} = 200\ 000$ real samples. So an FFT with 131 072 samples can be used. But now the same procedure as at every simulation: switch off the data compression and check whether GEAR is activated as Integration method.

To get a true FFT result a time slot is used again to eliminate "the time before steady state". And the time slot reaches from zero crossing to zero crossing of the IF output signal. This gives a slot from $1.02\mu\text{s}$ to $19.02\mu\text{s} = 19\mu\text{s}$.



Here we have the information about the pass band curve: the level difference is 1.5dB.

6.3.2. Simulation using a Dirac pulse

This is a very modern technique enabled by the progress in Computer-, Storage and Communication Technique. Let us apply it to our task of the pass band curve simulation and compare the results.

A Dirac pulse must be applied to the input of the converter. This is something crazy because its amplitude value is in finite but the pulse length is immeasurable short. **Important: the pulse area is "1" and constant** (e. g. = 1 Volt x 1 second or 1 MegaVolt x 1 μs etc.). Such a pulse has a constant spectral power density at every frequency – up to infinite.

Applying such a pulse to the input corresponds to the usage of a sweep generator with constant output voltage starting at Null and sweeping up to infinite.

In practice it is allowed to use an “approximated Dirac pulse” to get the same results in a system or a circuit. This means that a pulse with a maximum pulse length of

$$t_{\text{pulse}} = 0,01 \cdot \frac{1}{\text{cutoff frequency of the system}}$$

must be applied because its frequency spectrum is (nearly) constant over this frequency range.

And in a LTI system (= Linear and Time Invariant) it is allowed to reduce the amplitude down to a value which does not damage the circuit (...it is not a good idea to use a pulse with 1 μs of pulse length and an amplitude of 1 MegaVolt to feed the input of a transistor amplifier...). The form of the result curve (= **impulse response**) at the output will stay the same at reduced amplitude values. *And by applying a mathematical method named „Convolution“ to this curve we get the unknown frequency response of the circuit or the system!*

Practical example = the converter:

Let us test the properties of the converter up to **200 MHz**. Then the Dirac pulse length must be less than 1% of 1 / 200MHz = 1% of 5ns = **50 picoseconds**.

Rise time and fall time are equal and are chosen to 1...2% of the pulse length. Let us say: e.g. **1 picosecond**.

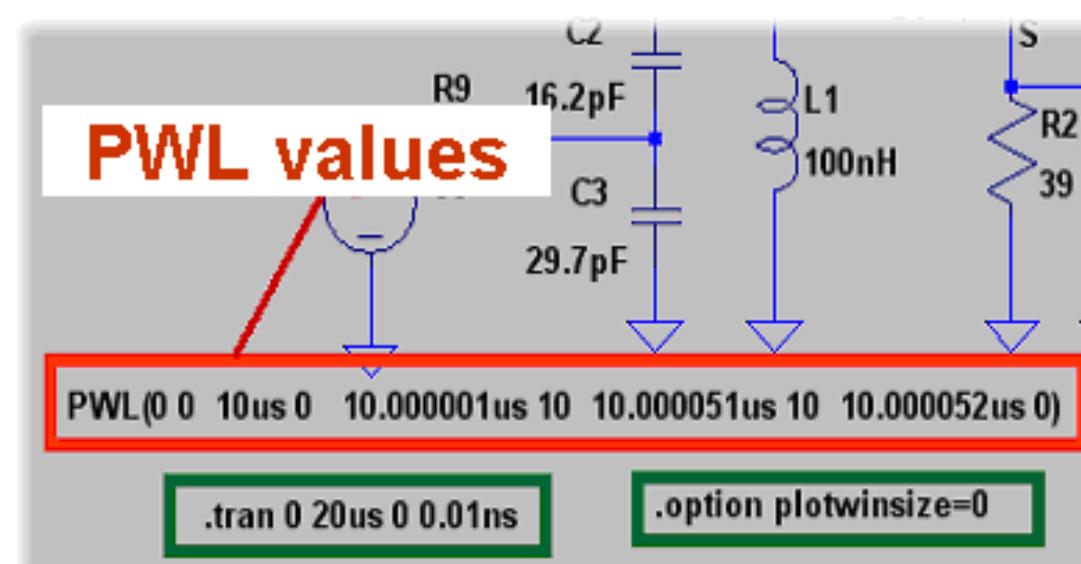
A pulse amplitude value of 10V will not damage the input of the LNA but a start delay should be programmed. So the pulse is applied when the complete converter has left its starting procedure and is in “steady state”.

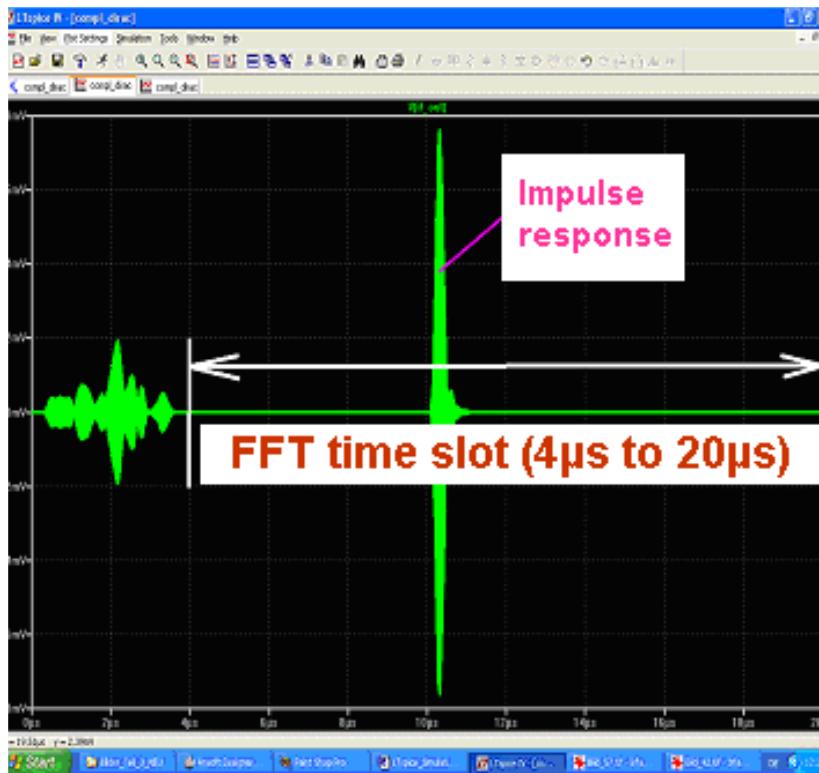
Simulation for **20 μs** gives a frequency resolution of **1 / 20μs = 50kHz** in the calculated spectrum and after a **start delay of 10μs** starts the pulse with a length of 50ps. With a maximum time step of **10 picoseconds** to avoid that the pulse is “overseen” by LTspice) the rest is simple: switch off the data compression, check the “GEAR” adjustment – and run the simulation.

Caution:

The simulation will take a long time (= **20 minutes** on the PC with 2GHz at the author) and produce a **very big result file with 1 Gigabyte**! This is clear when reflecting that now 2 000 000 samples and at every sample lots of currents and voltages in the circuit must be calculated.

The Dirac pulse is realized by programming a PWL source (see the figure below). Data compression is switched OFF and GEAR must be active as integration method.





Now start the simulation and wait to see this diagram.

The time slot for the FFT is chosen to 4μs...20μs to ignore the “starting procedure”.

And a “Hamming window function” is used to smooth the abrupt changes at the start and the end of the time slot. So simulation errors due to these “steps” are reduced.

Number of data point samples in time: 131072

Time range to include

- Start Time: 4μs
- End Time: 20μs
- Specify a time range
- Use Extent of Simulation Data
- Use current zoom Extent

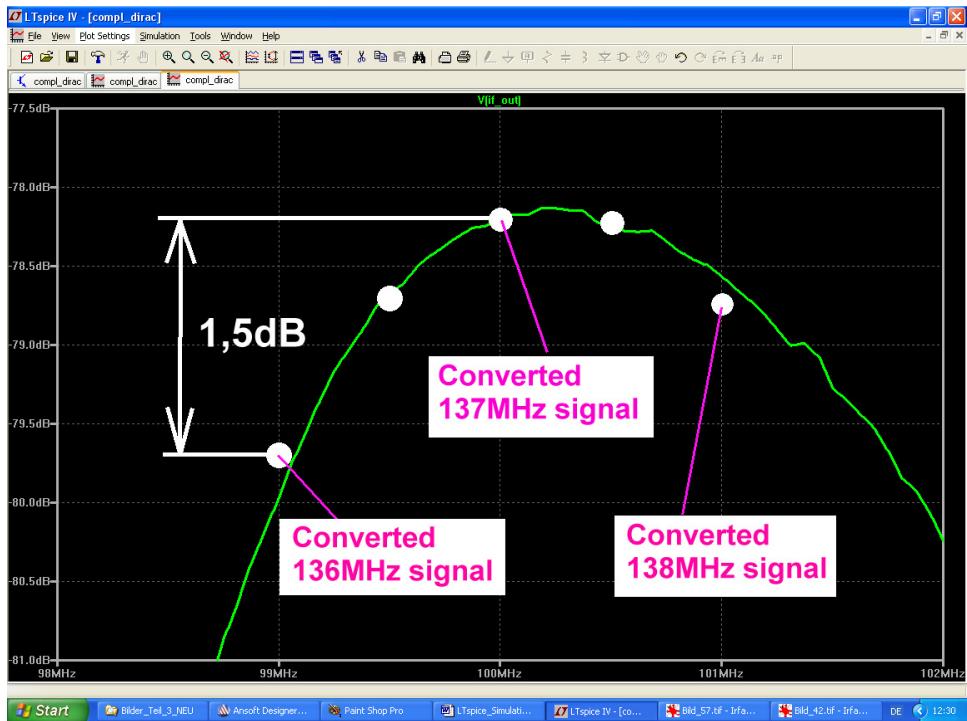
Binomial Smoothing done before FFT and windowing

Number of Points: 5

Windowing [Periodic and normalized to unit area]

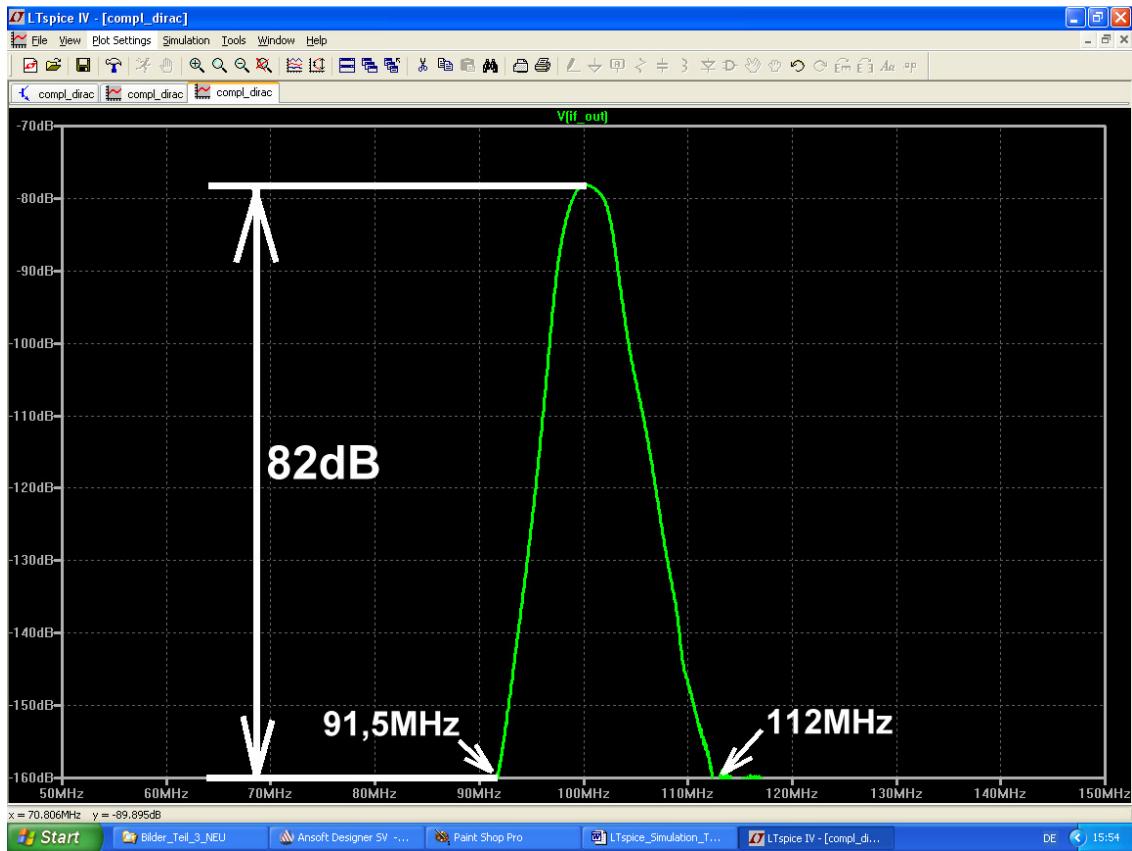
Windowing Function: Hamming

Kaiser-Bessel Parameter Beta: 2



Now the FFT with 131 072 samples can be applied and gives an interesting result.

It is exactly the same as with the separated spectral lines in the last chapter!



Also the wide band attenuation is a pure joy. A really fascinating technique!

6.4. IP3 simulation

This is a little difficult because in the converter there are now two stages (= LNA and DBM) producing intermodulation. A new publication [4] shows the different and complicated effects in this case caused by different compression points and different distortion properties.

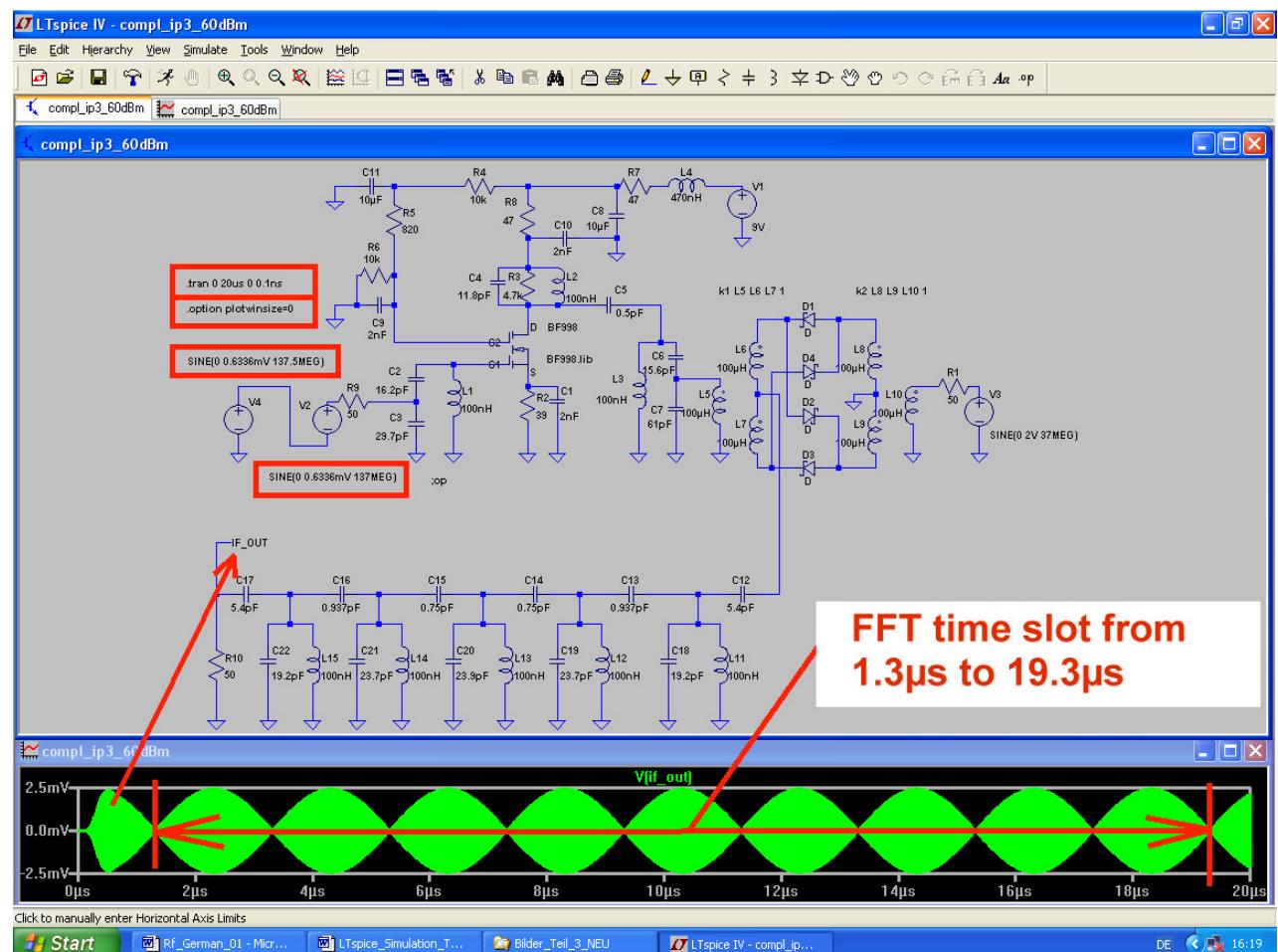
To get an overview to the distortion properties of the complete converter this way was used:

The IP3 simulation is started with very low input levels to get a level difference more but 60dB between the input signals and the IP3 products. Then the input levels are increased in steps of 10dB and checked at the output how long the output level difference decreases in steps of 20dB. Compression will start anywhere in the circuit as soon as this law is injured.

Start with an input level of **-60dBm** – (this gives an RMS value of 0.224mV and a peak value of 0.3168mV for the incident wave and a peak value of $2 \times 0.3169\text{mV} = 0.6336\text{mV}$ for the source voltage).

The used signal frequencies are **137MHz and 137,5MHz** with a frequency difference of 500kHz.

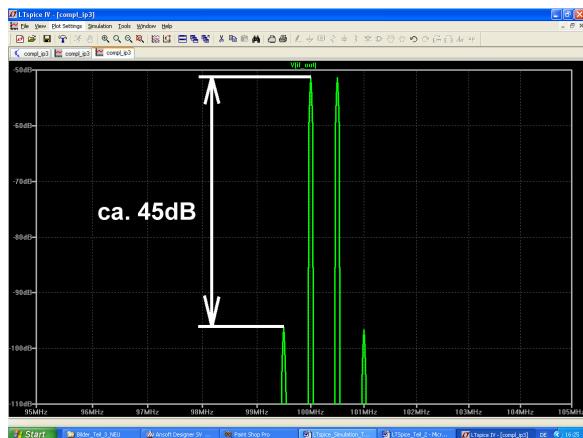
Simulation time is 20 μs and a maximum time step of 0.1ns is used. Do not forget to switch OFF the data compression and to check the GEAR adjustment.



This is the well known IP3 simulation schematic using two voltage sources (connected in series) at the input. Peak voltage values are 0.6336mV to give an incident wave of -60dBm .Signal frequencies are 137MHz and 137.5MHz, a time slot of 18 μs (= from 1.3 μs to 19.3 μs) is used.

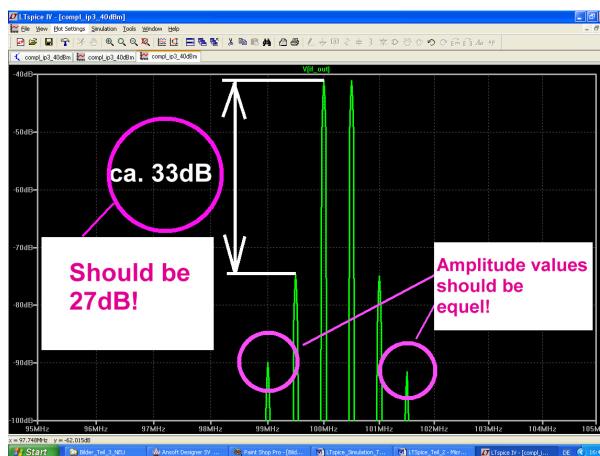


This is the result for input levels of **-60dBm**. A level difference of 67dB between input signal and IP3 product can be observed.

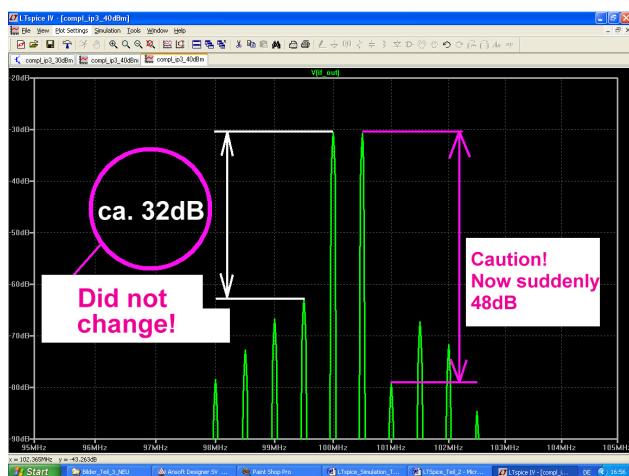


Now the simulation is repeated with input levels of **-50dBm**.

The level difference decreases to only 45db and this seems to be the border to the range of unexpected behaviour (because the theoretical value should be 47dB)



But when using **-40dBm** at the input, curious affects occur. The level difference falls to 33dB instead of 27dB but the IP3 products with higher order show different attenuation at the left hand side compared to the right hand side of the input signals.



And at an input level of **-30dBm** everything gets crazy: the level difference suddenly remains constant on the left hand side but increases on the right hand side.

But now we know a little bit more about the allowable maximum input voltages.

6.5. Noise Figure of the complete converter

This is impossible to simulate with LTspice due to the frequency conversion by the mixer. SPICE can only handle spectral noise density in the frequency domain. So we do not worry, have look at the existing formulae for this purpose and calculate (like our fathers) the “**Cascaded Noise Figure**“:

$$F_{\text{total}} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 \cdot G_2} + \dots$$

But this is heavy work because we must not use the dB values!
Everything must be re-calculated to ratios and values.....

Let us start with the LNA.

The simulated noise figure NF was 0.64dB. This gives $F_1 = 10^{\frac{0.64}{10}} = 1.16$

And a gain of 25dB gives

$$G_1 = 10^{\frac{25}{10}} = 316$$

Now the calculation for the DBM.

The conversion loss for an ideal version would be 4.5dB.

So the power gain as ratio is:

$$G_2 = 10^{\frac{-4.5}{10}} = 0.355$$

And the noise factor = the inverse of the power gain:

$$F_2 = \frac{1}{G_2} = \frac{1}{0.355} = 2.82$$

Finally the band pass filter with its loss of 8dB.

Power gain : $G_3 = 10^{\frac{-8\text{dB}}{10\text{dB}}} = 0.158$

Noise factor = the inverse of the power gain:

$$F_3 = \frac{1}{G_3} = \frac{1}{0.158} = 6.3$$

This gives:

$$F_{\text{total}} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 \cdot G_2} = 1.16 + \frac{2.82 - 1}{316} + \frac{6.3 - 1}{316 \cdot 0.355} = 1.16 + \frac{1.82}{316} + \frac{5.3}{112.18}$$

$$F_{\text{total}} = 116 + 0,0058 + 0,047 = 1,21$$

So the noise figure will be:

$$NF = 10 \cdot \log 1.21 = 0.83\text{dB},$$

And that is equal to a noise figure increase of 0.2dB compared to the LNA.

This value cannot be reached in practice due to the ignored transformer losses in the mixer stage and variations of the MOSFET properties during production.

But we have a confirmation of the well known rule: the total noise of a receiver is determined by the noise of the first stage and its gain.

Literature

[1]: UKW-Berichte 1/1999, page 43

[2]: George D. Vendelin, Anthony M. Pavio, Ulrich L. Rohde: „Microwave Circuit Design using Linear and Nonlinear Techniques“, page 48. Wiley- Interscience, New York

[3]: Agilent – Design –Tip from the Internet: “Measuring IP3” by Bob Myers

[4]: Ralf Rudersdörfer, „Funkempfänger-Kompendium“. Elektor Verlag Aachen, Germany 2010. ISBN: 978-3-89576-224-6