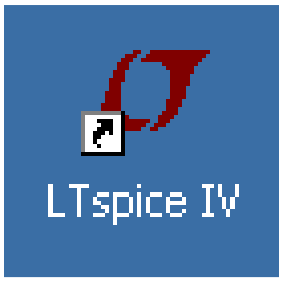
HF-Spezialausgabe: Vollständige Simulation eines 137MHz-Wettersatelliten-Empfangskonverters mit LTspice IV



Tutorial, Band 2

zur frei im Internet (www.linear.com) erhältlichen SPICE-Vollversion der Firma Linear Technologies

Version 1.1

Copyright by
Gunthard Kraus, Oberstudienrat i.R. an der Elektronikschule Tettnang
Gastdozent an der Dualen Hochschule Baden Württemberg (DHBW) in Friedrichshafen

Email: krausg@elektronikschule.de und Gunthard.Kraus@gmx.de
Homepage: www.elektronikschule.de/~krausg

28. April 2013

Anmerkung:

Dieses Manuskript ist eine leicht gekürzte Version der in der Zeitschrift „UKW-Berichte“ ab Heft 4/2009 veröffentlichten, dreiteiligen Artikelserie „Simulationen im HF-Bereich mit LTspice“.
Es soll zeigen, dass sich mit SPICE alle wichtigen HF-Schaltungseigenschaften sowohl im Zeit- wie auch im Frequenzbereich erfassen lassen -- und dazu hat man als kostenlose Dreingabe noch die nichtlineare Simulation. Das Ganze ist zum Selbst-Mitmachen am Rechner gedacht, aber der Anwender sollte sich vorher bereits mit Band 1 des Tutorials befasst haben.

Inhaltsverzeichnis — Seite

1. Wozu SPICE im HF-Bereich? 3

2. Das untersuchte Objekt: ein 137MHz-Konverter für NOAA-Wettersatelliten-Empfang 3

3. Rauscharmer Vorverstärker mit dem Dualgate-MOSFET BF998 4
- **3.1.Erstellung des Bauteils „BF998“ für die SPICE-Simulation 4**
- **3.2. SPICE-Simulation der S-Parameter bei einem Drainstrom von 10mA 6**
  - 3.2.1. Arbeitspunkteinstellung 6
  - 3.2.2. Ermittlung der S-Parameter im gewählten Arbeitspunkt 7
- **3.3. LTspice-Simulation beim 137MHz-LNA 9**
  - 3.3.1. Bestimmung des Arbeitspunktes 9
  - 3.3.2. Ermittlung der S-Parameter beim LNA 10
  - 3.3.3. Simulation der Rauschzahl (NF in dB) mit LTspice 12
  - 3.3.4. Stabilitätskontrolle mit LTspice 14
  - 3.3.5. Simulation im Zeitbereich: ein 137MHz-Signal am Eingang 15
  - 3.3.6. Bestimmung des IP3-Punktes 16

4. Der „Double Balanced Mixer“ (= Ringmodulator) 19
- **4.1. Zuerst etwas Grundlagen 19**
- **4.2. Simulation des DBM mit LTspice 20**
  - 4.2.1. Schaltung für die LTspice-Simulation 20
- **4.2.2. Simulation der Mischereigenschaften 21**

5. Der Ausgangs-Bandpass 24
- **5.1. Vorbemerkungen 24**
- **5.2. Schaltungsentwurf mit Ansoft Designer SV 24**

6. Untersuchung der Gesamtschaltung 28
- **6.1. Vorbereitung 28**
- **6.2. Bestimmung der Gesamtverstärkung bei Mittenfrequenz 29**
- **6.3. Ermittlung der Durchlaßkurve des Konverters 29**
  - 6.3.1. Simulation mit einzelnen Spektrallinien 29
  - 6.3.2. Frequenzgangbestimmung über einen Dirac-Impuls 31
- **6.4. Großsignalverhalten und IP3-Punkt 34**
- **6.5. Rauschzahl des vollständigen Konverters 36**

Literaturverzeichnis 37

1. Wozu „SPICE“ im HF-Bereich?

Diese Frage scheint berechtigt, wenn man an die guten und kostengünstigen linearen S-Parameter - Simulationsprogramme (wie PUFF oder ANSOFT DESIGNER SV) denkt, die dem „Normal-Anwender mit kleinem Geldbeutel“ zur Verfügung stehen. Nun, die Antwort ist eigentlich ganz einfach:

Wenn wir es nämlich schaffen, dass die SPICE-Simulation einer HF-Schaltung (z. B. eines LNA) im gleichen Arbeitspunkt dieselben Ergebnisse liefert wie die lineare Simulation mit den vom Hersteller bereitgestellten S-Parameter-Files, dann tun sich unglaubliche Möglichkeiten auf:

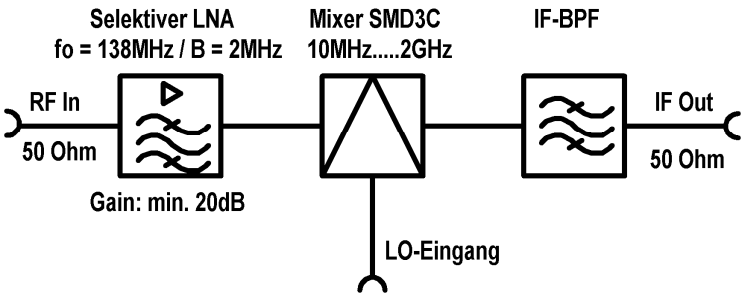
In diesem Fall repräsentiert das Modell wirklich exakt den tatsächlich erhältlichen Baustein (und dessen GEMESSENE S-Parameter). Also darf man anschließend beliebig mit der Arbeitspunkteinstellung herumspielen und ist nicht mehr auf die wenigen vom Hersteller zur Verfügung stehenden S-Parameter-Dateien bzw. auf einen bestimmten Arbeitspunkt angewiesen! Und zusätzlich kann man den Simulationsergebnissen im Zeitbereich (= bei Ansteuerung mit anderen Signalformen als Sinus oder bei Übersteuerungen) jetzt wirklich trauen.

Dann kann man nach eigenen Überlegungen Änderungen vornehmen und sieht sofort die Auswirkungen -- das SPICE-Modell wird immer die richtigen Ergebnisse im gerade eingestellten Betriebsfall liefern. Das Großsignal-Verhalten und alle Übersteuerungseffekte können im Zeitbereich beobachtet sowie die zugehörigen Spektren mittels der FFT analysiert werden und das wird dann z. B. bei Endstufen richtig interessant. Die Frequenzgänge der Verstärkung, der Rauschzahl und der S-Parameter stehen ebenso zur Verfügung wie alle Gleichspannungen und Gleichströme innerhalb der Schaltung und die Auswirkung jeder Schaltungsänderung kann deshalb blitzschnell ermittelt werden. Und mit einem „Parameter Sweep“ kann man dann nach dem Optimum suchen. Dadurch wird die S-Parameter-Simulation keineswegs überflüssig, denn sie basiert z. B. bei Halbleiterbauteilen immer auf vorher **EXAKT GEMESSENENEN Eigenschaften** (und bildet damit den entscheidenden Prüfstein bzw. die Referenz). PSPICE arbeitet dagegen mit einem Modell des Bauteils, das seine Gesamteigenschaften annähern soll und da kann es manchmal mit der Exaktheit doch etwas holpern. Und für bestimmte Bauteile (Stichwort: Microstrip-Leitungen) gibt es nichts. Man muss da improvisieren und mühsam das allgemeine SPICE-Modell (hier für die „Lossy Tline“ = verlustbehaftete Transmission Line) selbst anpassen.

An einer Baugruppe aus der HF-Technik soll nun das genaue Vorgehen gezeigt und die Verlässlichkeit der obigen Behauptungen geprüft werden.

2. Das untersuchte Objekt: ein 137MHz-Konverter für NOAA-Wettersatelliten-Empfang

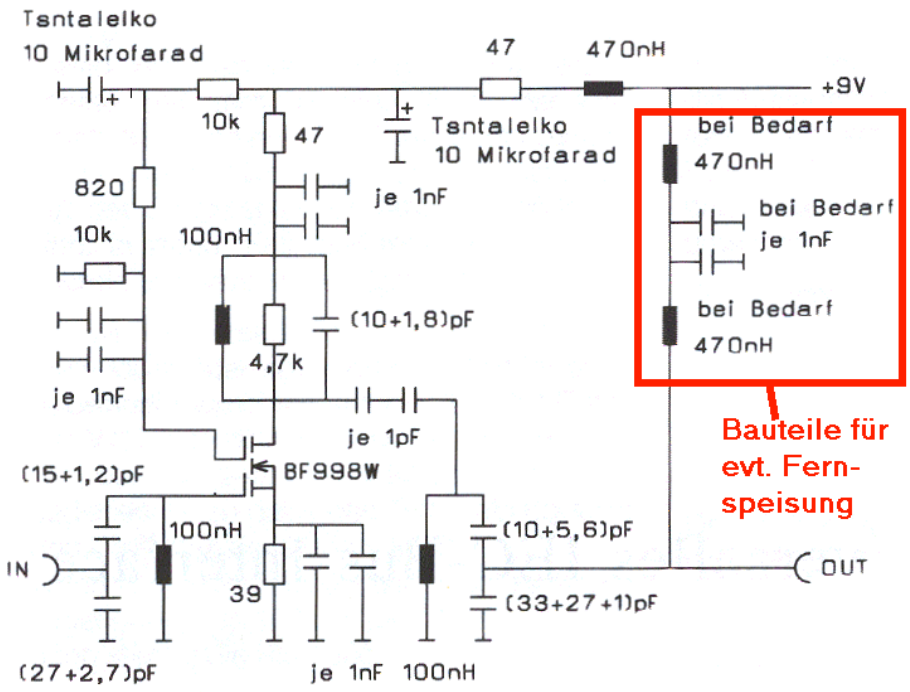
Dieses Projekt wurde gewählt, weil aus einer früheren Veröffentlichung in der Zeitschrift „UKW-Berichte“ („Puff-Einsteigerprojekt. Rauscharmer Vorverstärker für 137MHz bzw. 145MHz“. Heft 4/1998, Seite 245) ein rauscharmer Vorverstärker mit dem Dualgate-MOSFET BF998 für den Frequenzbereich von 136…138MHz vorhanden war, der immer wieder mal beim Autor seinen Dienst tun muss. Im erwähnten Artikel finden sich außer den Simulationen auch die Messergebnisse bei den verschiedenen gefertigten und untersuchten Prototypen. Da soll LTspice IV jetzt zeigen, ob es mithalten kann.



Der Rest des Konverters ist schnell erklärt: aus dem vorhandenen Vorrat an immer wieder mal ausgetüftelten Bausteinen wurde hinter den LNA erst ein Mischer (= käuflicher Double Balanced Mixer SMD3C, auf eigene Platine gesetzt und in ein gefrästes Alugehäuse eingebaut) und dann ein ZF-Bandpassfilter zur Aussiebung der Zwischenfrequenz nachgeschaltet. Als Umsetzoszillator diente beim Test ein in der Kellerwerkstatt sowieso vorhandener älterer professioneller Messsender (= auf 37MHz eingestellt). Das ZF-Signal mit 99…101MHz wurde nach der Aussiebung mit einem geeigneten Bandpass einem passenden FM-Empfänger (hier war es ein hochwertiges Autoradio) zur Auswertung und Demodulation zugeführt. Das obige Bild zeigt den Übersichtsschaltplan der Anordnung.

3. Rauscharmer Vorverstärker mit dem Dualgate-MOSFET BF998

Dies ist der komplette Stromlaufplan des Bausteins aus der Veröffentlichung.



Die Rauschanpassung wird im Eingangskreis durch einen kapazitiven Spannungsteiler vorgenommen, während beim Ausgangs ein kapazitiv gekoppeltes Zweikreis-Bandfilter eingesetzt wird. Als Koppelelement dient dort ein „Interdigitalkondensator“ mit 0,5pF und der Ausgangskreis arbeitet wieder mit einem kapazitiven Teiler zur Leistungsanpassung der 50Ω-Last. Die Stufenverstärkung liegt bei ca. 20dB, die Rauschzahl etwas unter 1dB.

3.1.Erstellung des Bauteils „BF998“ für die SPICE-Simulation

```
*   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
```

Reihenfolge der Anschlüsse

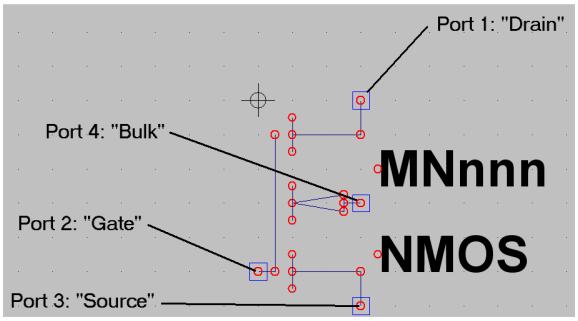
Das muss weg!

Das geht natürlich mit der Beschaffung dieses Modells aus dem Internet los (…Eingabe in die Suchmaschine: „bf998 spice model“) und das Ergebnis zeigt dieses Bild. Es sieht zwar schön aus und die Anschlüsse sind sauber durchnummeriert, aber darin gibt es eine ganz bösartige Tücke:

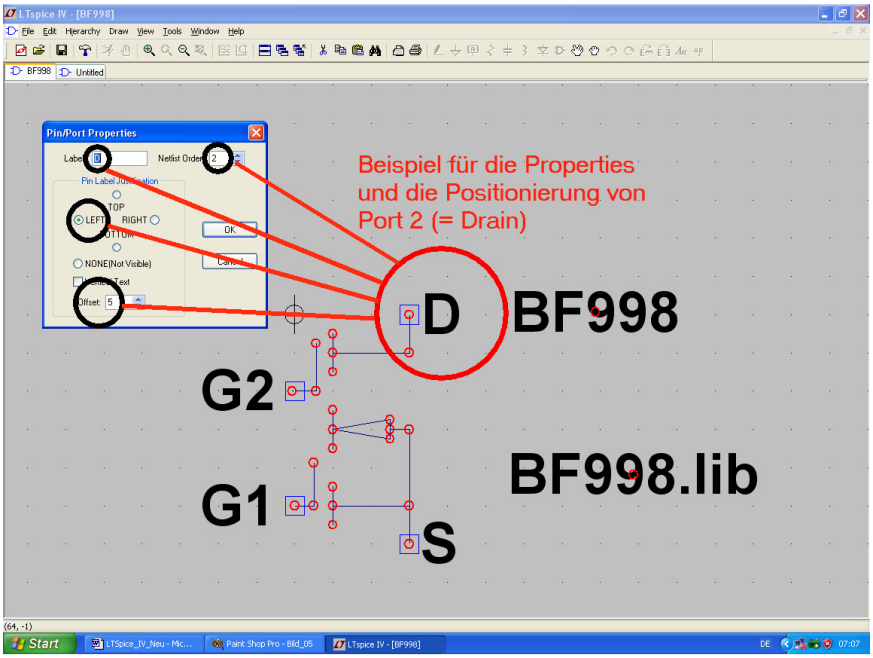
Der Modell-Autor hat es mit den Anwendern wohl zu gut gemeint und bei den Induktivitäten bzw. Kapazitäten die Zusätze „L=“ und „C=“ eingefügt. Dagegen protestiert jedoch SPICE und lehnt jede Simulation brüsk ab.

Deshalb: raus mit diesem Zeug und dann das Ergebnis als „**BF998.lib**“ im LTSpice-Bibliotheksordner „**lib** / **sub**“ gespeichert!

Nun wollen wir zuerst das Symbol für einen Dualgate-MOSFET erstellen, gehen in das Menü „File“ und klicken auf „**New Symbol**“. Die zum Zeichnen erforderlichen Werkzeuge (Linie, Kreis, Rechteck etc.) gibt es unter „Draw“ und über dem Bildschirm liegt ein Punkraster zur Erleichterung der Zeichenarbeit.



Bevor man mit dem Zeichnen beginnt, sollte man sich aus der mitgelieferten Bibliothek (Pfad: „LTspiceIV \ lib \ sym“ das Schaltzeichen „**nmos4.asy**“ eines MOSFETs auf den Bildschirm laden und ausdrucken. So sieht man anhand des Rasters sofort, wie groß das eigene Symbol werden muss, damit es anschließend in die Familie passt und weder zu groß oder zu klein ist.
Dann hat man die Wahl: entweder löscht man dieses Symbol ganz vom Schirm (= erst die Taste F5 drücken, auf das Symbol mit der linken Maustaste klicken, schließlich mit „Escape“ wieder zurück) und zeichnet den Dualgate-MOSFET komplett neu.

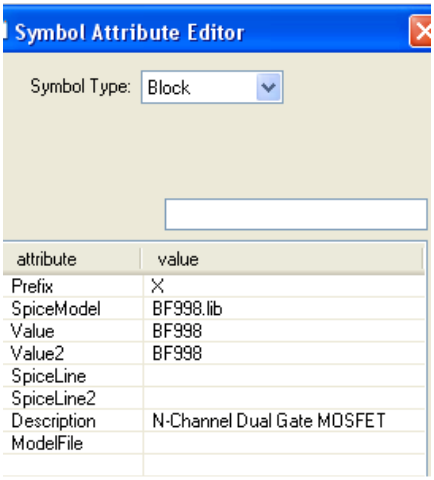


Oder man ändert nur das Schaltzeichen „**nmos4.asy**“ entsprechend ab und tauft es dann „**BF998.asy**“. In jedem Fall sind drei Hinweise nötig:

1) Die kleinen roten Kreise markieren die Enden von Linien und sind sowohl Zeichenhilfen wie auch später Fangpunkte beim Einsatz des Bauteils.

2) Die eigentlichen Bauteilanschlüsse (korrekte Bezeichnung: „Ports“ = kleine Rechtecke mit einem roten Kreis im Innern) können ***nicht gezeichnet werden***, denn dafür existiert eine eigene Funktion. Sie steckt hinter „Edit“ und heißt „**Add Pin** / **Port**“. Nach dem Anklicken erscheint zunächst das Property Menu und das Bild zeigt, was darin einzutragen ist, nämlich die Anschlussbezeichnung, die Port / Pin – Nummer laut SPICE-Modell und die Position der Anschlussbezeichnung neben dem Symbol (= Left / Offset =5). Dabei gilt beim BF998: **Source = Port 1** / **Drain = Port 2** / **Gate2 = Port 3** / **Gate 1 = Port 4**. Nach dem Absetzen wird jeder Port mit einer einfachen Linie mit dem Symbol verbunden.

3) Will man irgendetwas auf dem Schirm **verschieben**, dann drückt man die Taste **F7**. Nach einem Klick auf das entsprechende Teil hängt es am Cursor und kann an seinen neuen Platz bewegt werden. Ein weiterer Mausklick setzt es ab und mit ESC verlässt man diese Funktion.



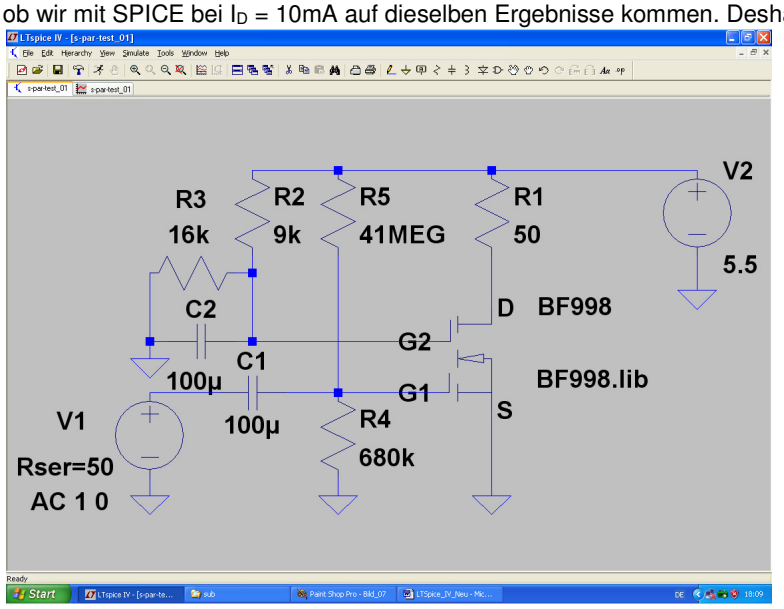
Jetzt wird das fertig gestellte Symbol erst mal im Bibliothekspfad „SpiceIV / lib / sym“ als „**dgate_MOSFET.asy**“ für weitere spätere Aktionen gespeichert. Erst dann machen wir einen echten BF998 daraus, indem wir unter „Edit“ das Menü „**Attributes** / **Edit Attributes**“ öffnen und diese Liste ausfüllen. Das ist etwas trickreich, da man jedes Mal erst auf die betreffende Zeile klicken und anschließend die Eingabe im ***Fenster oberhalb der Liste vornehmen muss***. Erst wenn man anschließend auf eine andere Zeile klickt, wird die Eingabe übernommen!

Aber dann haben wir es fast schon geschafft: das „Edit“ – Menü wird nochmals geöffnet, aber nun „**Attribute** / **Attribute Window**“ gewählt (…damit wird festgelegt, was an zusätzlichen Informationen auf dem Bildschirm angezeigt wird). Da taucht eine Liste auf, bei der wir auf die Zeile „Value“ klicken und mit OK bestätigen. Damit finden wir plötzlich die Bezeichnung „**BF998**“ neben dem Symbol. Also wiederholen wir die Prozedur und sorgen dafür, dass auch noch die SPICE-Modell – Bezeichnung „**BF998.lib**“ sichtbar wird. Nun kann man sich im „**lib** / **sub**“ – Ordner einen weiteren Ordner mit dem Namen Dualgate-MOSFETs anlegen und endlich das Teil als „**BF998.asy**“ darin speichern. Fertig!

3.2. SPICE-Simulation der S-Parameter bei einem Drainstrom von 10mA

3.2.1. Arbeitspunkteinstellung

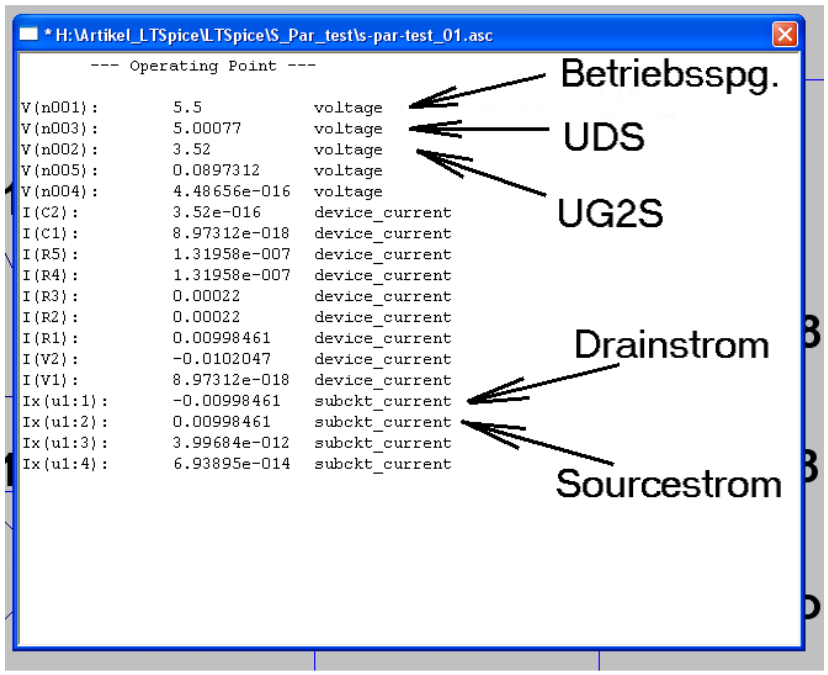
Im Internet finden sich nur die S-Parameter für einen Drainstrom von 5mA und 10mA. Da schauen wir mal nach, ob wir mit SPICE bei I_D = 10mA auf dieselben Ergebnisse kommen. Deshalb zeichnen wir uns zuerst eine passende Messschaltung und wollen dafür sorgen, dass sich derselbe Arbeitspunkt wie beim S-Parameter-File (dort werden I_D = 10mA / U_{DS} = 5V / U_{G2S} = 3,5V angegeben) einstellt. Widerstand, Kondensator und Massezeichen finden sich oben in der Menüleiste, während man für die Spannungsquellen erst auf das „component"-Verzeichnis (rechts neben der Diode, als UND-Baustein dargestellt) klicken muss. Dann sucht man in der auftauchenden Liste nach „voltage" und platziert 2 Quellen in der Schaltung. Unser neuer BF998 findet sich jetzt ebenfalls in der Liste und mit „Wire" verbinden wir schließlich alles zur korrekten Schaltung.



Die Programmierung der beiden Spannungsquellen ist recht einfach: ein rechter Mausklick auf das Schaltzeichen und anschließend auf „Advanced" öffnet ihr Property Menü. Bei der Quelle V1 (= Eingangssignal) tragen wir unter „Small Signal AC Analysis" eine Amplitude von „1" und eine Phase von „0" sowie den zur späteren S-Parameter-Simulation nötigen Innenwiderstand mit 50Ω ein. Die Quelle V2 liefert dagegen die Versorgungsgleichspannung und deshalb ist nur unter „DC Value" ein Eintrag von 5,5V nötig. Ebenso leicht lassen sich neue Werte bei den Widerständen und Kondensatoren nach einem rechten Mausklick auf ihr Symbol eingeben.

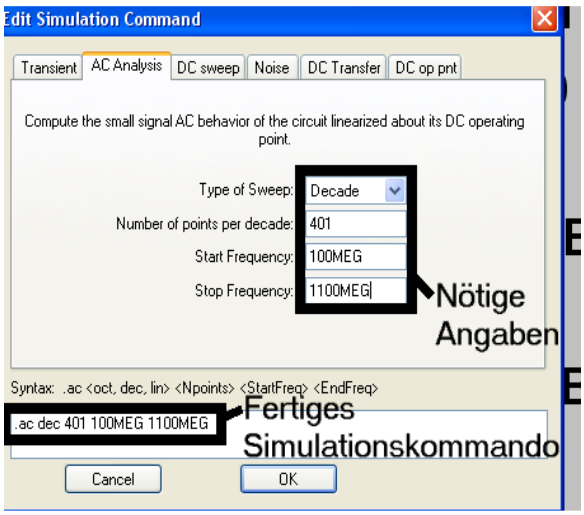
Wer sich die Schaltung genauer ansieht, erkennt schon, worauf wir hinauswollen: der Drainwiderstand R1 wird natürlich bei der S-Parameter-Bestimmung den korrekten Abschluss der Schaltung mit 50Ω darstellen, während die übrigen Widerstände als Spannungsteiler zur Einstellung des Arbeitspunktes dienen. Kondensator C1 trennt den Gate1-Spannungsteiler vom 50Ω-Innenwiderstand der Quelle V1 und Kondensator C2 erdet Gate2 im betrachteten Frequenzbereich.

Jetzt fehlt nur noch ein Simulationskommando für die DC-Analyse. Unter „Simulate" findet sich „**Edit Simulation cmd**" und das führt uns zu einem Auswahlmenü, in dem wir auf die Karteikarte „**DC op. Pnt**." wechseln. Ein Druck auf OK sorgt dafür, dass das Kommando „**.op**" am Cursor hängt und in der Schaltung platziert werden kann. Fertig!



Wer nun auf „das rennende Männchen neben dem Hammer" in der Menüleiste klickt, bekommt diese Tabelle zu sehen, in der alle Spannungen und Ströme aufgelistet sind. Wem die Auswertung zu mühsam ist, der schließe sie und fahre den Cursor auf einen interessierenden Punkt in der Schaltung: unten links wird dann der zugehörige Spannungs- oder Stromwert eingeblendet. Da alle Werte im grünen Bereich sind und mit den Messbedingungen im S-Parameter-File übereinstimmen, kann es nun mit der eigentlichen PSPICE-S-Parameter-Simulation losgehen!

3.2.2. Ermittlung der S-Parameter im gewählten Arbeitspunkt



Wieder geht es über „Simulate“ erst zu „Edit Simulation cmd.“, dann aber auf die Karteikarte „AC Analysis“. Wir wählen einen dekadischen Sweep, eine Startfrequenz von 100MHz (..im offiziellen S-Parameter-File geht es sogar bei 50MHz los…), eine Stoppfrequenz von 1100MHz und 401 Punkte pro Dekade.
Jetzt fehlt nur noch eine „SPICE-Direktive“ zur Bestimmung der S-Parameter. Über „Edit“ und „SPICE Directive“ kommen wir an diese Option heran und tragen im freien Feld ein:

.net I(R1) V1

Das ist leicht zu verstehen:

Mit „.net“ wird die Netzwerks-Simulation aufgerufen.
„I(R1)“ ergibt die Spannung am Abschlusswiderstand R1 = 50Ω
„V1“ bedeutet, dass die Spannungsquelle V1 die „hinlaufende Welle“ (= incident wave) erzeugt.

Die vorige Direktive „.op“ wird vom Programm automatisch mit einem Strichpunkt versehen und damit ist von der Arbeitspunktbestimmung auf AC-Sweep umgeschaltet worden. Falls das Ergebnisdiagramm nach der Simulation noch leer bleibt, klicken wir mit der rechten Maustaste darauf und wählen „Add Trace“. In dieser Liste brauchen wir nur nacheinander die vier S-Parameter S11 / S21 / S12 / S22 anzuklicken und OK zu drücken. Hinterher sollte man anschließend gleich mit der linken Maustaste auf die Zahlenteilung der waagrechten Frequenzachse fahren, bis ein Lineal auftaucht. Mit einem linken Klick kommt man an die Achseneinstellungen heran und kann auf lineare Darstellung von 100MHz bis 1100MHz sowie einen Tick von 100MHz umschalten. In gleicher Weise ändert man bei der senkrechten Achse den Amplitudenbereich auf +20db bis -80dB mit einem Tick von 10dB. Vom Programm wird zunächst außer dem Amplitudenverlauf auch der Phasenverlauf ausgegeben. Wenn das stört (oder nicht interessiert), dann fährt man mit der Maus auf die Zahlenteilung der senkrechten Achse am **rechten Bildrand** und klickt. Dort findet man dann die Option „**Don’t plot Phase**“.

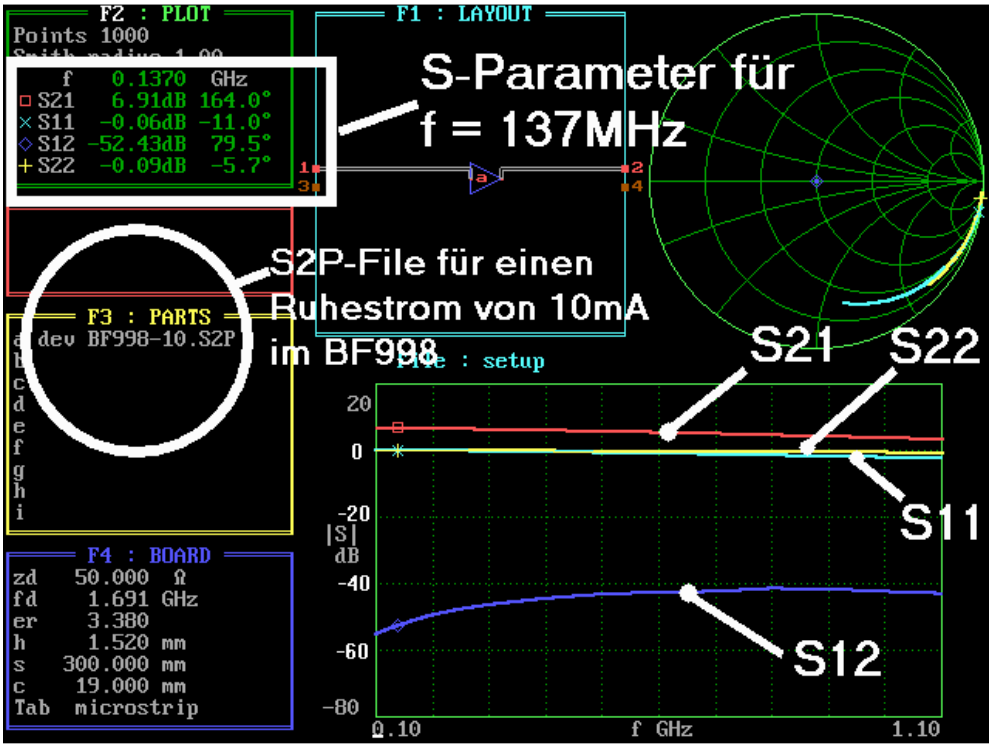
Hier kann der Erfolg bewundert werden, aber das Wichtigste ist doch jetzt der Vergleich mit einer echten S-Parameter-Simulation.

Da ist erst mal die aus dem Internet geholte und auf dem eigenen Rechner gespeicherte S2P-Datei für I_D = 10mA zu sehen.

```
! 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-Parameters:                    February  1991
# GHz  S  MA  R  50
!  f          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
```

Arbeitspunkt

Die damit durchgeführte Simulation zeigt dieses Bild und jeder Experte erkennt sofort, dass hier zu „PUFF“ gegriffen wurde…



Vergleicht man jetzt beide Simulationen (LTSpice und PUFF) miteinander, so fällt einem schon ein Stein vom Herzen, denn die Kurvenverläufe stimmen (bis auf leichte, aber nicht kritische Abweichungen bei S12) weitgehend überein. Da hat sich der Schöpfer des BF998-SPICE-Modells wirklich den Kopf zerbrochen und gute Arbeit geleistet!

Zum Abschluss des Kapitels wollen wir für die Frequenz f = 137MHz noch die genauen Simulationsergebnisse beider Simulationsprinzipien (Time Domain Simulation gegen S-Parameter Simulation in der Frequency Domain) in einer Tabelle zum Vergleich gegenüberstellen. Man erhält sie in beiden Fällen durch den Einsatz eines Cursors:

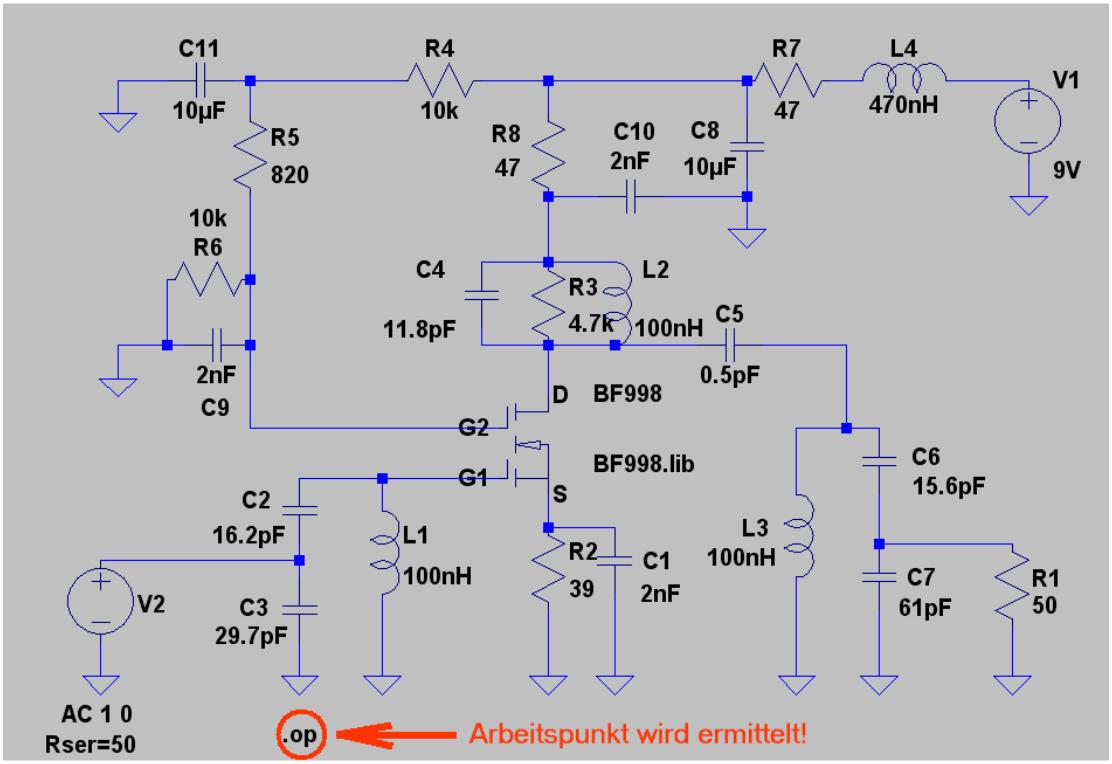
S-Parameter	LTSpice	PUFF
S11	-0,085dB	-0,06dB
S21	+7,95dB	+6,91dB
S12	-58dB	-52dB
S22	-0,05dB	-0,09dB

Wer jetzt wegen der Unterschiede die Stirn runzelt, möge bedenken: das SPICE-Modell kam von der Firma Philips, das S-Parameter-File dagegen von der Firma Siemens. Und von Serienstreuungen haben wir noch gar nicht gesprochen….
Also trauen wir uns mal und simulieren die komplette LNA-Schaltung -- genau so, wie sie aufgebaut wurde.

3.3. LTspice-Simulation beim 137MHz-LNA

3.3.1. Bestimmung des Arbeitspunktes

Man hat natürlich etwas Arbeit vor sich, bis man das Schaltbild aus der Veröffentlichung in ein passendes Schaltbild für die SPICE-Simulation umgewandelt hat .



Nicht vergessen darf man, bei jeder Schwingkreisspule mit L = 100nH im zugehörigen Property Menü jeweils einen Verlustwiderstand von 1Ω für eine Güte von etwa Q = 60 einzutragen. Doch dann geht es schnell, denn das Simulationskommando „**.op**“ kennen wir schon aus Kapitel 3.2.1. und in der Ergebnistabelle finden wir:

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

Das sind genau die Messbedingungen, wie sie im S-Parameter-File für einen Drainstrom von 5mA angegeben werden!

3.3.2. Ermittlung der S-Parameter beim LNA

Nach den Übungen in den vorigen Kapiteln ist das jetzt eine Kleinigkeit, wenn man diese Schaltung weiterverwendet:

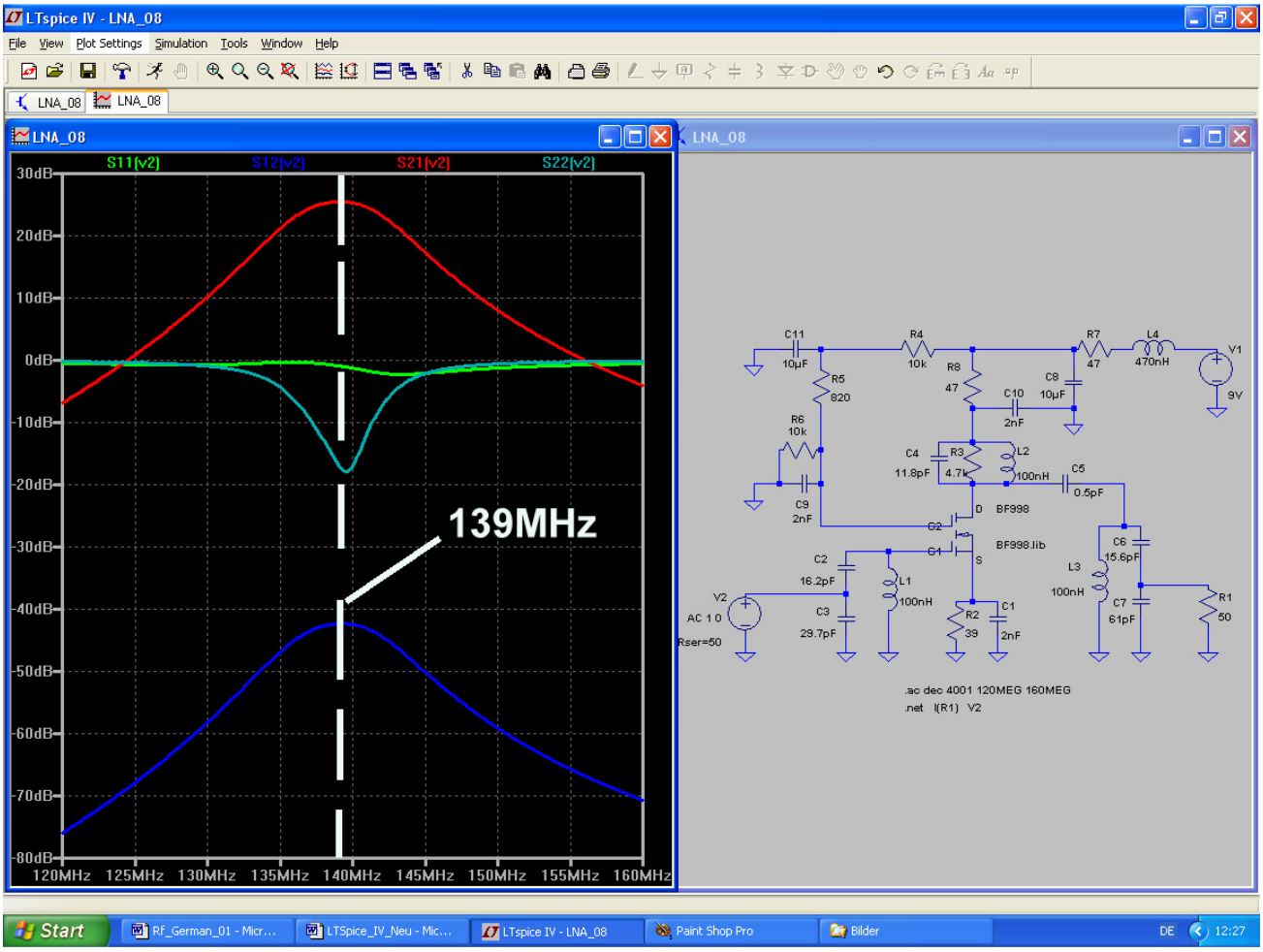
a) Man programmiert unter „Edit Simulation Command“ einen dekadischen AC-Sweep von 120….160MHz und wählt **4001 Punkte** pro Dekade. Dazu gehört das Kommando

.ac dec 4001 120MEG 160MEG

b) Zusätzlich schreibt man eine SPICE-Direktive für die S-Parameter-Simulation (passend zum gezeichneten Schaltbild):

.net I(R1) V2

R1 stellt den Abschlusswiderstand der Stufe mit 50Ω dar und die ansteuernde Signalquelle heißt eben V2. Bei ihr achten wir darauf, dass die Einstellungen „**AC- Amplitude AC = 1**“ und „**AC-Phase = 0**“ sowie der **Innenwiderstand von 50Ω** stimmen.
Nun können wir simulieren und sorgen anschließend dafür, dass bei allen S-Parametern die Phase nicht angezeigt wird, dass eine lineare Frequenzachse von 120MHz bis 160MHz mit einem Tick von 5MHz aktiviert ist und dass der Amplitudenbereich von +30dB bis -80dB mit einem Tick von 10dB dargestellt wird.



Das Ergebnis ist hier zu sehen und zum direkten Vergleich folgt dazu noch auf der nächsten Seite das Bild aus der ursprünglichen Veröffentlichung [1].

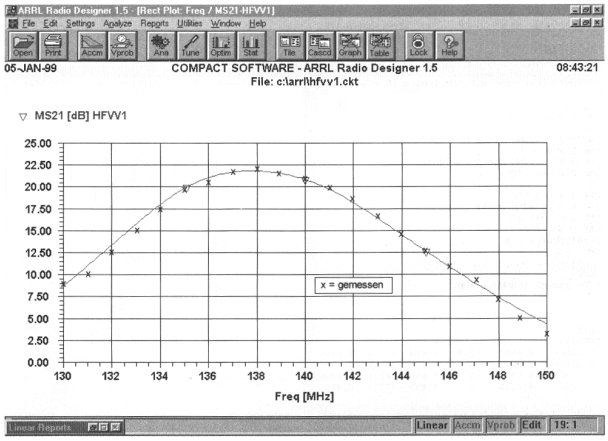
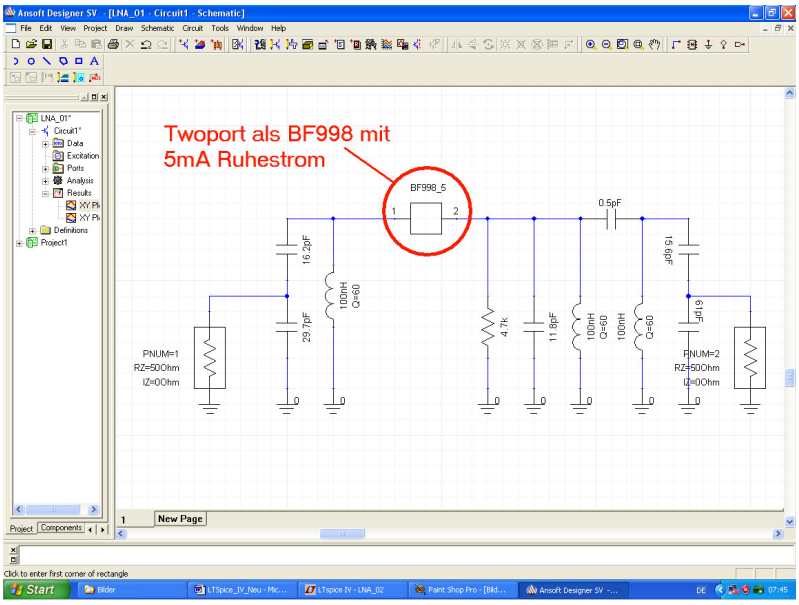
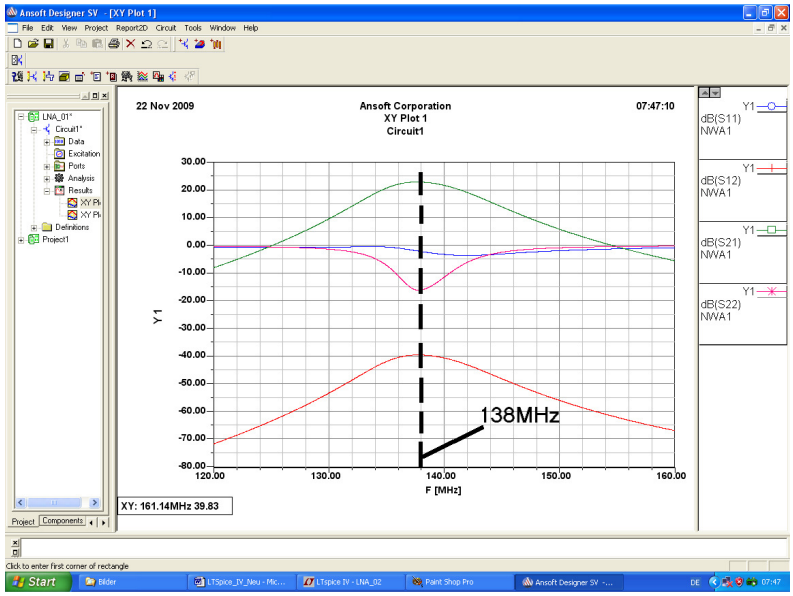


Bild 21: Vergleich zwischen Simulation und Messung

Es zeigt die damalige Simulation mit dem ARRL-Radio-Designer und die am fertigen Objekt gemessene S21-Kurve. Sie sagt eine (anschließend auch durch Messung bestätigte) Verstärkung von ca. 22,5 dB bei Mittenfrequenz aus und das sind etwa 2,5dB weniger, als SPICE behauptet. Außerdem liegt die gemessene Mittenfrequenz der Durchlasskurve bei138MHz, während SPICE eher auf 139MHz tippt.



Also werfen wir noch den Ansoft Designer SV an und simulieren die Schaltung mit dem S-Parameter-File für I_D = 5mA. Das ist die Schaltung, mit der dieses Programm gefüttert wird …..



…und hier folgt die S-Parameter-Simulation.

Der Vergleich von SPICE- und Ansoft-Ergebnis ist dann wirklich aufschlussreich:

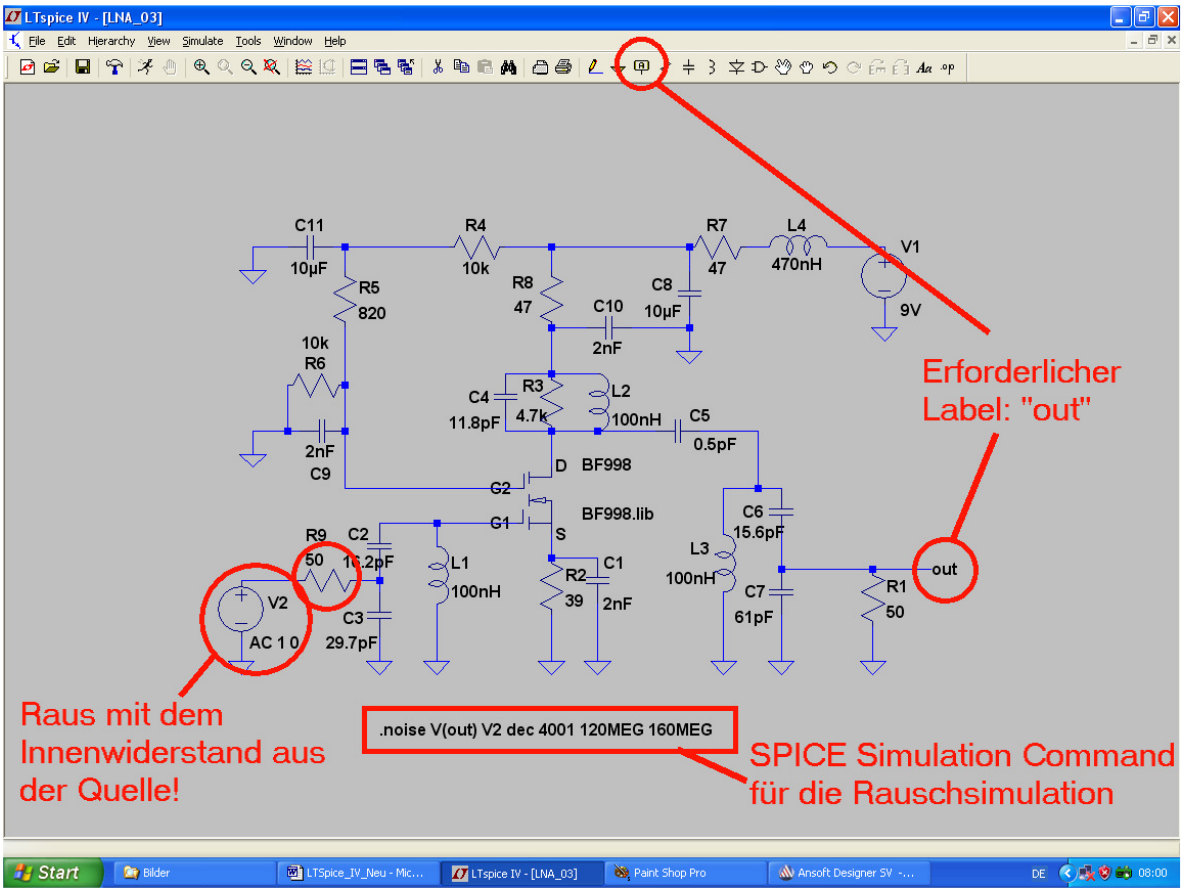
Ansoft ist für eine Verstärkung von ca. 23 dB bei knapp 138MHz -- also 1MHz weniger als SPICE. Ansonsten sind beide Simulationen bei allen S-Parametern fast identisch!

Die Sache mit der Mittenfrequenz ist dabei besonders harmlos: Bezogen auf 137MHz beträgt die Abweichung bei der Ansoft-Simulation weniger als 0,7%. Bei SPICE sind es dann ca. 1,3%. Aber dazu haben ja die Spulen Abgleichkerne, um so etwas aufzufangen. Da kann man sich jetzt wirklich beruhigt zurücklehnen…..

3.3.3. Simulation der Rauschzahl (NF in dB) mit LTspice

Um ein Gespür für die zu erwartende Größenordnung zu bekommen, geht ein Studium verschiedener Datenblätter (mit der Angabe der typischen Rauschzahl) und des S-Parameter-Files für ID = 10mA (…nur darin finden sich auch Rauschparameter) für den BF998 voraus. Je nach Hersteller finden sich darin Werte von 0,6dB (Siemens / Infineon) über 0,7dB (Philips) bis hin zu 1dB (Vishay) für den Bereich zwischen 100 und 200MHz. Da wird es ja richtig interessant, was SPICE dazu meint.

Dazu müssen wir die Schaltung etwas ändern und passende Kommandos bereitstellen. **So sieht sie dann vor der Simulation aus:**



Bei der ansteuernden Spannungsquelle V2 müssen wir den Innenwiderstand von 50Ω herausnehmen und ihn durch den zusätzlich eingefügten Widerstand R9 ersetzen.

Am Abschlusswiderstand R1 ist ein Label „out“ erforderlich. Er findet sich rechts neben dem Massezeichen in der Menüleiste.

Über „Simulate“ rufen wir wieder „Edit Simulation Command“ auf und schreiben das für die Rauschsimulation erforderliche Kommando. Es lautet für einen dekadischen Sweep mit 4001 Punkten von 120 bis 160MHz:

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

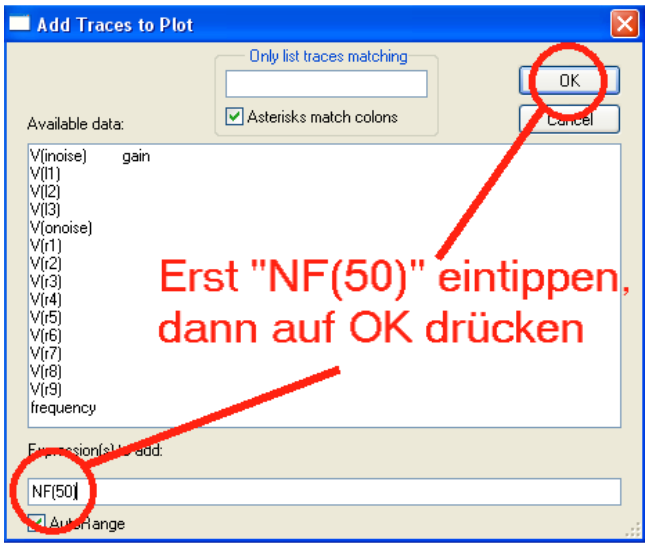
Wenn wir nun simulieren und mit der Prüfspitze auf den Abschlusswiderstand R1 tippen, dann bekommen wir die „**Spektrale Rauschleistungsdichte für die Ausgangsseite**“ zu sehen. Wer lieber die auf den **Eingang zurückgerechnete Spektrale Rauschleistungsdichte** sehen möchte, der klicke einfach auf die Leitung zwischen Eingangsspannungsquelle V2 und Widerstand R9.

Für die Simulation der Rauschzahl „**NF in dB**“ ist dagegen noch etwas zusätzlicher Aufwand nötig. Wir öffnen dazu ein leeres Blatt in einem Texteditor (z. B. Notepad) und tippen folgende Zeile ein:

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

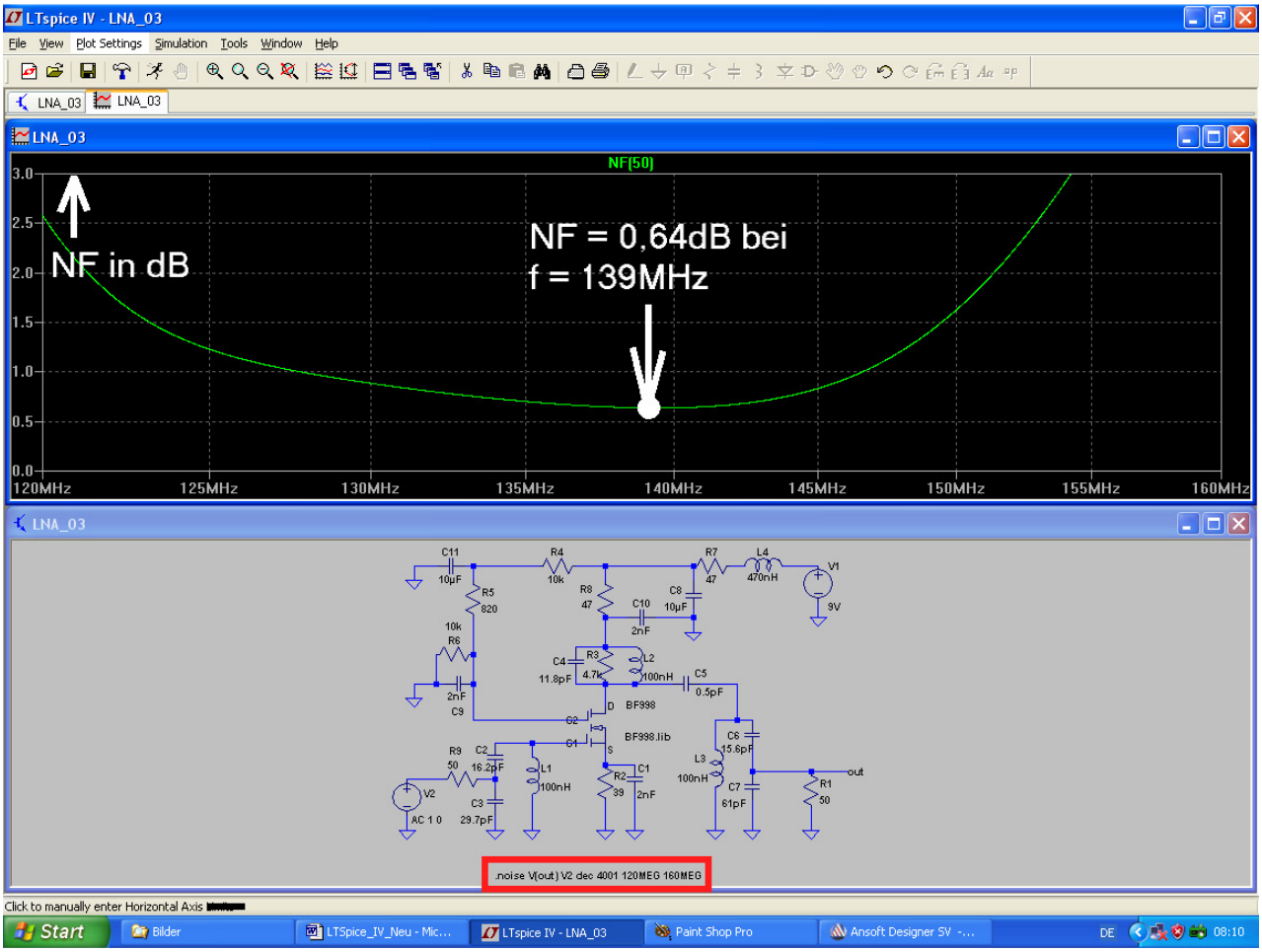
Anschließend wird diese neue Datei als „**plot.defs**“ im **LTSpice-Ordner** gespeichert und das Programm neu gestartet, um die Änderung wirksam zu machen.
Durch diese Maßnahme stellen wir eine neue Funktion (= „.func“) mit dem Namen „NF(50)“ zur Verfügung, mit deren Hilfe die Rauschzahl NF für ein 50Ω-System mit einer Temperatur von 300 Kelvin berechnet und in der Einheit „dB“ geplottet werden kann.

Und so geht das weiter:



a) Man simuliert die Schaltung

b) Nach Abschluss der Simulation sehen wir auf dem Bildschirm die Schaltung und ein leeres Ergebnisdiagramm. Darauf klicken wir „mit Rechts“ und öffnen das Menü „Add Trace“. In das leere untere Eingabefeld tippen wir nun den Funktionsaufruf „NF(50)“ ein und erhalten nach OK die simulierte Rauschzahl NF in dB.



Hier sieht man den Verlauf von NF mit einer vernünftigen Einstellung der Diagrammachsen und wenn man das mit den vorhin aufgeführten Datenblattwerten vergleicht, kann man nur noch „Donnerwetter“ sagen.

3.3.4. Stabilitätskontrolle mit LTspice

Dafür besitzen moderne HF-Simulationsprogramme mindestens die Möglichkeit zur Ausgabe des Stabilitätsfaktors „k“. Da wir die S-Parameter bereits simuliert haben (und mit ihnen „k“ berechnet werden kann), müssen wir uns in diesem Fall nur um die erforderliche Mathematik kümmern und dann eine Funktion schreiben. Da holt man sich als erstes die korrekte Formel aus einem der modernen schlauen Bücher über Mikrowellentechnik, z. B. aus [2]:

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

Anschließend wird wieder mit einem einfachen Texteditor (wie Notepad) die Datei „plot.defs“ entweder geöffnet oder neu angelegt und darin diese Formel in einer Form eingetragen, die LTspice versteht.

Achtung: der Buchstabe „k“ ist in LTspice schon für andere Zwecke reserviert, also nehmen wir „**klinear**“.

So sieht die Umsetzung der Formel in eine SPICE-Funktion aus (die anschließend im LTspice-Ordner in der Datei „**plot.defs**“ gespeichert sein muss). Ist etwas mühsam und erfordert pingelige Sorgfalt bei der Eingabe, gibt aber sonst keine Probleme. Und bei geduldiger Analyse der geschweiften Klammer kann man gut die Umsetzung der obigen Formel in eine für SPICE verständliche Form mitverfolgen:

.func klinear(S11,S21,S12,S22)

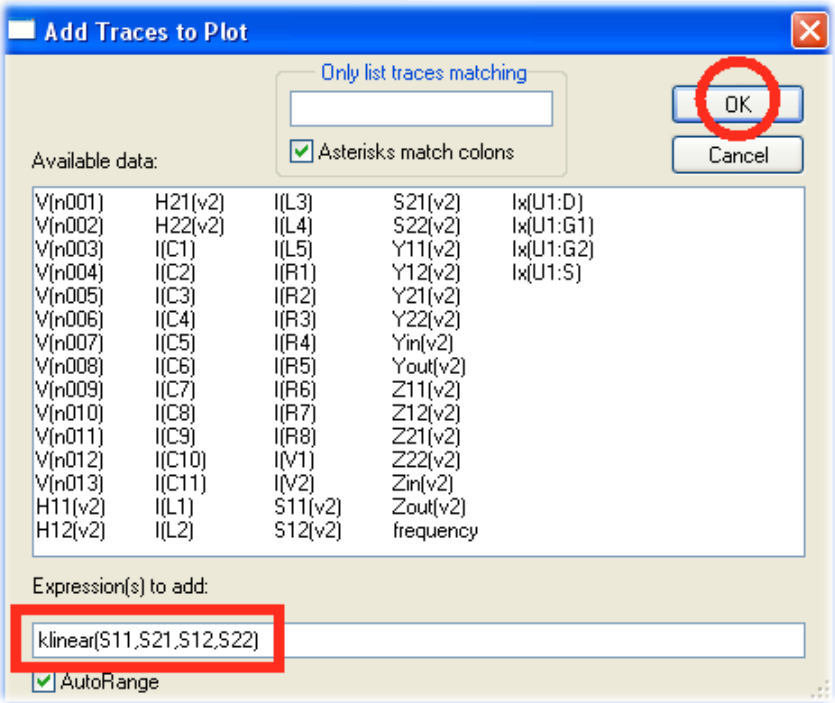
{(1-abs(S11(v2))*abs(S11(v2))-abs(S22(v2))*(S22(v2))+abs(S11(v2)*S22(v2)-S12(v2)*S21(v2))* abs(S11(v2)*S22(v2)-S12(v2)*S21(v2)))/(2*abs(S12(v2))*abs(S21(v2)))}

(Bitte keinerlei Leerzeichen oder sonstige Steuerzeichen in die Formel, also in die geschweifte Klammer, einbringen. Einfach ohne Pause alles am Stück eintippen!).

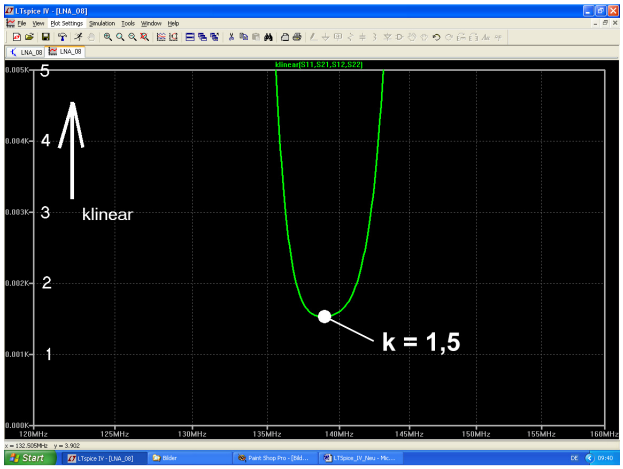
Eine Sache darf man wieder nicht vergessen:

Nach dem Abspeichern von „plot.defs“ muss LTSpice nochmals neu gestartet werden, damit der Zugriff klappt!

Jetzt wird nochmals die **Simulation der S-Parameter laut Kapitel 3.3.2**. aufgerufen und wiederholt. Dann sorgt man (nach einem rechten Mausklick auf das Ergebnisdiagramm) erst mal mit „**Add Plot Pane**“ für ein neues, leeres Diagramm.



Eine Wiederholung dieser Prozedur mit „ADD TRACE“ ergibt **dieses Bild**, bei dem wir im freien Feld **„klinear(S11,S21,S12,S22)“** eintippen und dann OK drücken.



Mit etwas Nachbearbeitung (lineare Darstellung, Amplitudenbereich von k = 0 bis k = 5) erhält man schließlich diesen Verlauf mit kmin = 1,5 bei 139MHz.

(Dabei wurde die Strichbreite der dargestellten Kurve vergrößert. Diese Einstellung versteckt sich hinter dem „Control Panel“, also dem Button mit dem Hammer, auf der Karteikarte „Waveforms“,. Man muss dort „**plot Data with thick lines**“ aktivieren).

Einen sehr ähnlichen Verlauf erhalten wir (allerdings mit einem Minimum von k =1,0 statt wie hier k = 1,5) mit einer Ansoft-Designer-SV-Simulation!
Wer deshalb beim Entwurf der praktischen Schaltung ganz sicher gehen will, der füge noch im Sourcekreis eine kleine Induktivität zwischen MOSFET und Sourcewiderstand als „Schwingschutz bei Streuungen“ ein. Ein Wert von 1,5…2 nH reicht schon, um „k“ auch in der Ansoft-Simulation auf einen Minimalwert von 2 zu bringen.

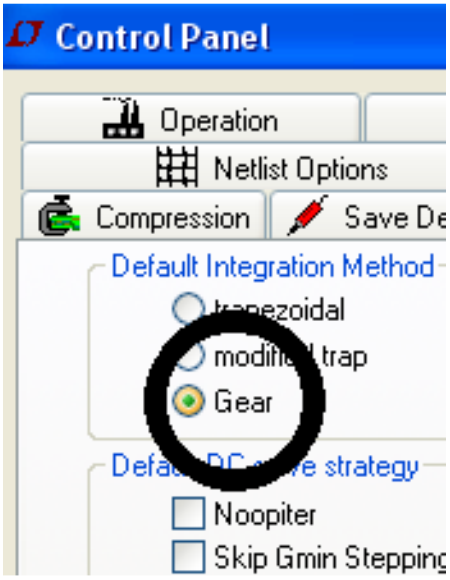
3.3.5. Simulation im Zeitbereich: ein 137MHz-Signal am Eingang

Wir haben bis jetzt bei unserer Schaltung nur im Frequenzbereich gearbeitet und das wäre mit normalen S-Parameter-Simulationsprogrammen auch gegangen (…sofern man die passende S-Parameter-Datei besitzt). Die wichtigste Grundeigenschaft von SPICE ist jedoch seine Simulation im Zeitbereich, also die Ermittlung von Spannungs- und Stromverläufen. Also steuern wir unsere Schaltung mal mit einem Sinussignal (f = 137MHz, Spitzenwert = 30mV) an und schauen, welche HF-Spannungen in der Schaltung zu erwarten sind. Dazu sind folgende Änderungen nötig:

-- Die ansteuernde Quelle wird auf „**Sinus“ mit f = 137MHz und U = 30mV** umgestellt

-- Wir simulieren die ersten 200ns mit einem größten Zeitschritt von 0,1ns. Das zugehörige Simulationskommando lautet: **.tran 0 200ns 0 0.1n**

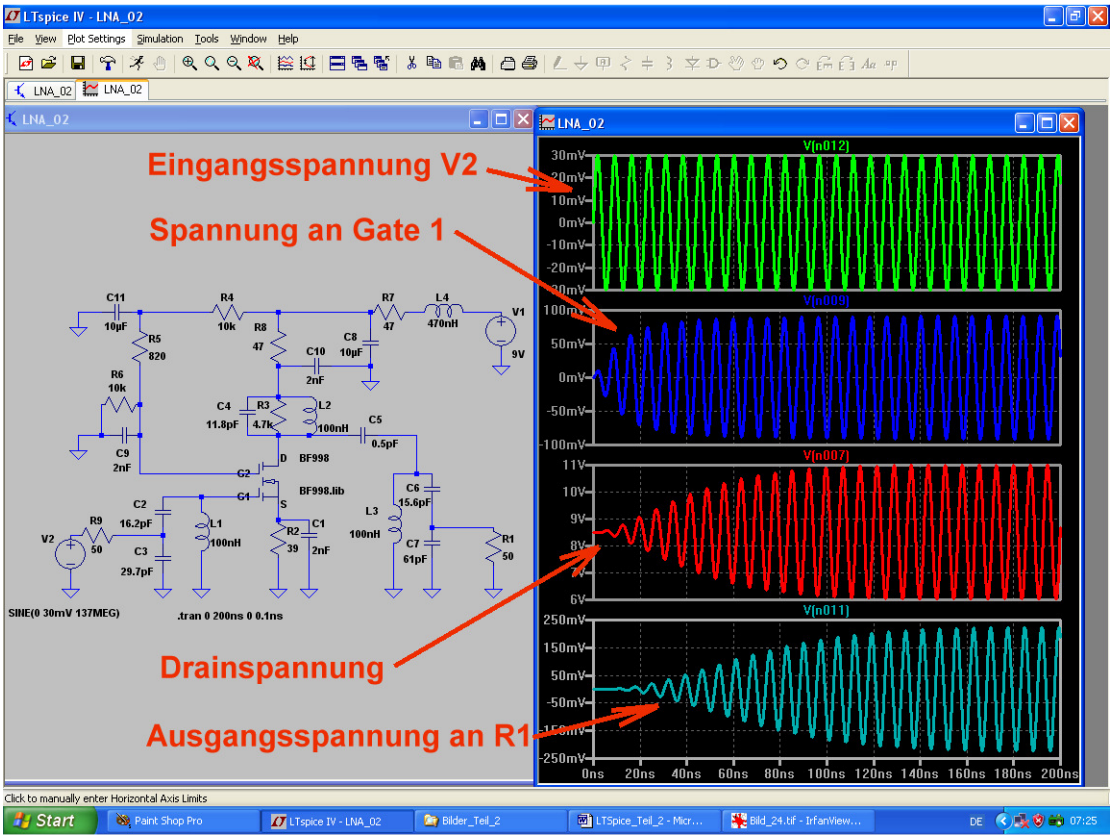
Alle anderen Direktiven und Kommandos können vom Bildschirm gelöscht werden.



Jetzt kommt die wichtigste Sache:
Man klickt auf den „**Control Panel Button**“ (= Button mit dem Hammer) und geht zur Karteikarte „SPICE“. Dort wird „**Gear**“ als Integrationsmethode gewählt, **denn die anderen beiden Einstellungen machen nur Probleme bei der HF-Simulation!**

Und aus gegebenem Anlass folgt der Hinweis:

Diese Einstellung kann man gar nicht oft genug kontrollieren, denn manchmal wird sie vom Programm (z. b. bei Schaltungsänderungen oder beim Neuzeichnen oder beim Erweitern..) ohne Vorankündigung geändert…



Da haben wir das Ergebnis mit vier in der Schaltung messbaren HF-Signalen. Dabei ist als Eingangssignal nicht die „hinlaufende Welle“ (= 0,5 x V2), sondern die eigentliche Urspannung V2 mit einem Spitzenwert von 30mV vor dem Innenwiderstand R9 dargestellt.

3.3.6. Bestimmung des IP3-Punktes

Die Simulation des „***Third Order Intercept Points***“ liefert die erforderlichen Informationen über das Verzerrungsverhalten der Stufe bei steigendem Eingangspegel. Beim IP3 wären (…wenn das ginge, denn vorher laufen wir schon in die Begrenzung…) die Verzerrungen dritter Ordnung genau so groß wie das zugehörige Nutzsignal.
Dazu steuern wir die Stufe mit zwei „In Band-Signalen“ an, die nur einen geringen Frequenzunterschied aufweisen und in der Mitte des Durchlassbereiches liegen. Der Pegel wird so hoch gewählt (bei solchen Verstärkern meist irgendwo zwischen -30 und -10dBm, also zwischen Ueff = 7mV und etwa 70mV), dass damit bereits ein deutlicher Intermodulationseffekt zu erwarten ist.

Zur Simulation werden einfach zwei Eingangsspannungsquellen in Serie geschaltet und folgendermaßen programmiert:

Quelle V2 liefert einen Sinus mit einem Spitzenwert von 20mV und f = 137MHz.
Quelle V3 liefert einen Sinus mit einem Spitzenwert von 20mV und f = 137,5MHz.

Beim Simulationskommando ist Vorsicht geboten, denn da müssen wir wegen der anschließenden FFT (= Fast Fourier Transformation) zur Darstellung des Spektrums eine Grundregel der Systemtheorie beachten:

Die Auflösung -- also der kleinste dargestellte Frequenzschritt im Spektrum -- wird nämlich durch die „Gesamt-Simulationszeit“ festgelegt. Wollen wir eine Frequenzauflösung von 50 kHz erreichen, dann muss sich die Simulation über eine Zeit von

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

erstrecken.

Der „**Maximum Time Step**“ stellt dagegen den zeitlichen Abstand zwischen zwei aufeinander folgenden Sample-Punkten dar und entspricht der Periodendauer der Samplefrequenz. Die minimale Samplefrequenz selbst ergibt sich dann zu

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

und die muss deutlich höher sein als der höchste im Nutzsignal vorkommende Frequenzanteil (nach der Shannon-Bedingung mindestens doppelt so hoch!). Sonst gibt es „Aliasing-Verzerrungen“, die das Ergebnis verfälschen, viel Ärger verursachen und nicht mehr beseitigt werden können.

Also bestimmt die Samplefrequenz die höchste ohne Fehler im Spektrum darstellbare Signalfrequenz. Aber nicht nur das, denn entsprechend der Beziehung

$$\text{Samplezahl} = \frac{\text{Simulationsdauer}}{\text{Maximum Time Step}}$$

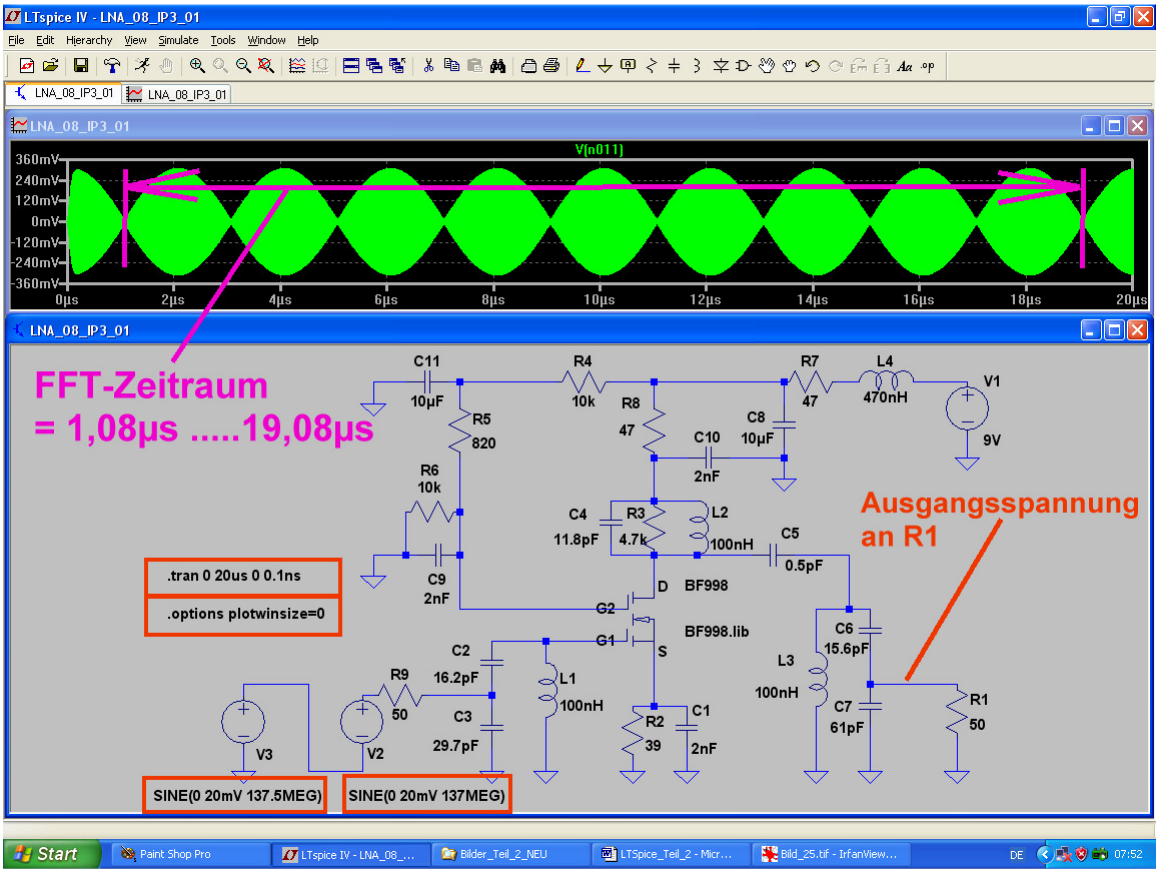
müssen wir uns einen bestimmten **Vorrat** an simulierten Kurvenpunkten (= „Samples“) anlegen. Das FFT-Programm fragt uns nämlich nachher, wie viel Kurvenpunkte zur Berechnung des Spektrums herangezogen werden sollen und da darf man natürlich nicht mehr Werte vorgeben, als tatsächlich simuliert wurden (…gibt sonst unerklärliche Fehler oder Abweichungen oder erhöhtes „Grundrauschen“, dazu manchmal irre lange Rechenzeiten….in jedem Fall Ärger).
Bei der FFT muss die Samplezahl immer eine Zweierpotenz sein und das Programm schlägt uns passende Werte vor -- zunächst 65 536 Samples. Wir wollen es etwas genauer haben und wählen den doppelten Wert (131 072). Deshalb sorgen wir für einen Vorrat von 200 000 Samples, indem wir den kleinsten Zeitschritt auf folgenden Wert festlegen:

Maximum Time Step = Simulationszeit / 200 000 = 20µs / 200 000 = 100ns

LTspice nimmt außerdem zur Verkleinerung der Ergebnisdatei automatisch eine Datenkompression vor, aber genau das können wir in diesen Fall überhaupt nicht brauchen. Deshalb schalten wir sie durch die selbst geschriebene SPICE-Direktive

.options plotwinsize=0

einfach ab (über den Weg: „Edit / Spice Directive“). Und eine letzte Kontrolle sollte wieder der richtigen Grundeinstellung für die Integrationsmethode bei der Spektrumsberechung gelten (= Button mit dem Hammer und dann „**Gear**“ auf der Karteikarte „Spice“). So sieht das dann aus:

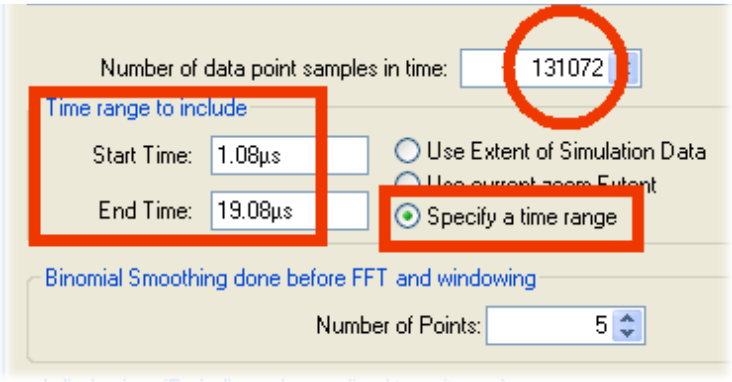


Die Simulationsschaltung, alle Kommandos und Direktiven und der zeitliche Verlauf der simulierten Ausgangsspannung sind gut zu sehen, ebenso die erwähnte Reihenschaltung der beiden Eingangsspannungsquellen. Die ist natürlich bei SPICE ganz einfach, aber in der Praxis erfordert die gleichzeitige Ansteuerung mit zwei Generatoren einen beträchtlichen Aufwand und stellt hohe Anforderungen an die Geräte bzw. die zur Addition erforderliche Schaltung (Siehe dazu [3]).

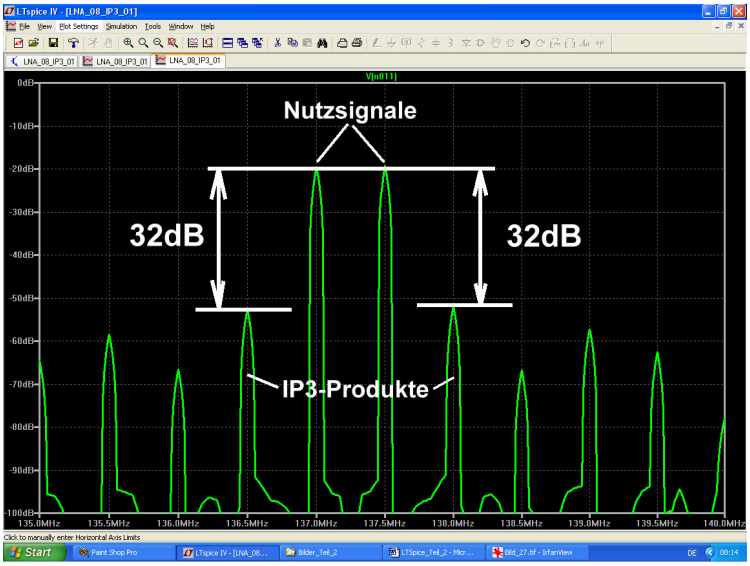
Beim Ausgangssignal ist noch eine wichtige Information eingetragen. Man sollte nämlich alles vermeiden, was das korrekte FFT-Ergebnis verfälschen kann und darf deshalb

a) den „Einschaltbereich der Schaltung“ nicht für die FFT verwenden, sondern nur den „**eingeschwungenen Zustand**“.

b) Bei der Wahl des FFT-Simulationszeitraums darf man keine Kurvenzüge „anschneiden“. Deshalb wurde der **Zeitraum von 1,08µs bis 19,08µs** (…diese genauen Nulldurchgänge wurden mit dem Cursor bestimmt….) ausgesucht und das reicht immer noch für 131072 Samples.



Hier sehen wir die nötigen Einstellungen im FFT-Menü. Bitte aber **ZUERST** die Sample-Anzahl von131 072 wählen und **HINTERHER** den zugehörigen Zeitbereich eintippen -- sonst wird das nicht akzeptiert….



Beim gezoomten FFT-Ergebnis genügt der dargestellte Frequenzbereich von 135…140MHz, denn da sind jetzt neben den Nutzsignalen die beiden Intermodulationsprodukte dritter Ordnung gut zu erkennen, mit deren Hilfe wir den IP3 ausrechnen werden. Da können wir nun eine wichtige Information herauslesen:

Die Pegeldifferenz zwischen jedem der beiden Nutzsignale und dem zugehörigen Intermodulationsprodukt dritter Ordnung beträgt ca. 32dB.

Damit lässt sich der IP3-Punkt folgendermaßen bestimmen:

IP3 in dBm = (Pegel des Nutzsignals) + 0,5(Pegeldifferenz zum IM3-Produkt)

Dabei hat man die Wahl zwischen der Angabe des IP3 für die Eingangsseite (= IIP3) oder für die Ausgangsseite (= OIP3). Die beiden Werte unterscheiden sich genau um die Stufenverstärkung.

Fangen wir mit der Eingangsseite an:
Die Urspannung beider Quellen weist einen Spitzenwert von 20mV auf. Der zugehörige Effektivwert ist um den Faktor 0,707 kleiner, damit erhalten wir 14,14mV. Bei einem Eingangswiderstand der Schaltung von genau 50Ω würden wir eine Eingangsspannung bekommen, die exakt halb so groß ist. Das wäre die berühmte „Hinlaufende Welle“ oder „Incident Wave“ mit 7,07mV. Sie entspricht einem Pegel von

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

Der Rest ist jetzt eine Kleinigkeit, denn damit erhält man den Eingangs-IP3 zu:

IIP3 = -30dBm + 0,5(32db) = -30dBm + 16dB = -14dBm

Bei einer Stufenverstärkung von ca. 23…24dB (Siehe die Simulationen in den vorigen Kapiteln) gehört dazu ein Ausgangs-IP3 von

OIP3 = IIP3 + (23…24dB) = -14dBm + (23…24db) = +9dBm…+10dBm

Etwa diesen Wert findet man auch meist im Internet in Anwendungsbeispielen und Application Notes zu Vorstufen und LNA's mit MOSFETs.
Das wär's mit der Eingangsstufe -- das Ganze ist doch etwas umfangreicher geraten. Aber dafür war es, glaube ich, recht interessant. Nun geht es mit der Mischstufe weiter.

4. Der „Double Balanced Mixer" (= Ringmodulator)

4.1. Zuerst etwas Grundlagen

Mit dieser bekannten Schaltung aus der Kommunikationstechnik kann man die Frequenzen von Signalen verändern, ohne dass die darin enthaltenen Informationen darunter leiden (Fachausdruck: „Frequenz-Umsetzung").

Sie besitzt **zwei Eingänge** und **einen Ausgang**:

a) Den Eingang für die Information, deren Frequenz umgesetzt werden soll (Name: RF = radio frequency signal). Es muss stets eine kleine Amplitude aufweisen.

b) Den Eingang für das Signal, das die Frequenzumsetzung bewirkt (Name: LO = local oscillator signal). Es muss stets eine große Amplitude aufweisen

c) Den Ausgang, an dem die umgesetzten Signale zur Verfügung stehen (Name: IF = intermediate frequency)

Wichtig:
Grundsätzlich handelt sich um eine Schaltung, in der zwei Eingangssignale miteinander multipliziert werden.
Und dafür hat die Mathematik eine Formel bereit, denn es gilt:

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

Wenn man die rechte Seite der Formel betrachtet, erkennt man:

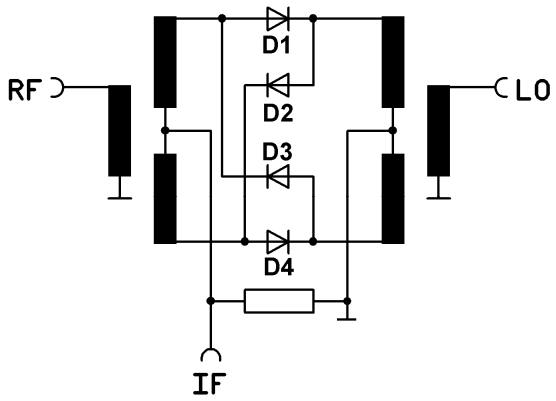
Nach dem Durchlaufen dieser Schaltung sind die beiden zugeführten Eingangssignale am Ausgang plötzlich verschwunden. Dafür beobachtet man dort ihre Summenfrequenz und ihre Differenzfrequenz!

Wenn man weiß, dass als LO-Signal meist eine symmetrische Rechteckspannung verwendet wird (oder die Amplitude eines sinusförmigen LO-Signals so groß wählt wird, dass es auf die Schaltung wie ein Rechtecksignal wirkt….), kann man ahnen, was da passiert:

Das symmetrische Rechtecksignal besteht bekanntlich selbst wieder aus einer Summe von sinusförmigen Signalen. Neben der „Grundwelle" findet man die „Ungeradzahligen Oberwellen", also die dreifache, fünffache, siebenfache, neunfache…..Grundfrequenz, wobei die Amplituden entsprechend dem Grad der Oberwelle abnehmen. Der Anteil mit der dreifachen Frequenz hat eine Amplitude von 1/3 der Grundwelle, der Anteil mit der fünffachen Frequenz eine Amplitude mit 1/5 der Grundwelle usw.

Jede dieser einzelnen Oberwellen wird nun im DBM mit dem RF-Signal multipliziert und erzeugt deshalb am Ausgang ebenfalls ein „Signal-Pärchen" mit der zugehörigen Summen- und Differenzfrequenz.

Der Anwender pickt sich dann mit Hilfe passender Filter das für seine Zwecke am besten geeignete Signal aus diesem Sammelsurium heraus.



Dieses in der gesamten Kommunikationstechnik unentbehrliche Teil gibt es in passiver wie auch in aktiver Version. Wir wollen uns mal die Prinzipschaltung der passiven Version ansehen. Sie besteht aus einem Ring von 4 schnellen Schottky-Dioden (bitte aufpassen: sie sind **NICHT** so gepolt wie beim Brückengleichrichter!) sowie zwei Übertragern. Jeder Übertrager ist mit drei gleichen Wicklungen ausgestattet. Die beiden Sekundärwicklungen werden dabei in Reihe geschaltet und die „Mittenanzapfung“ herausgeführt.

Und so funktioniert das mit der Multiplikation:

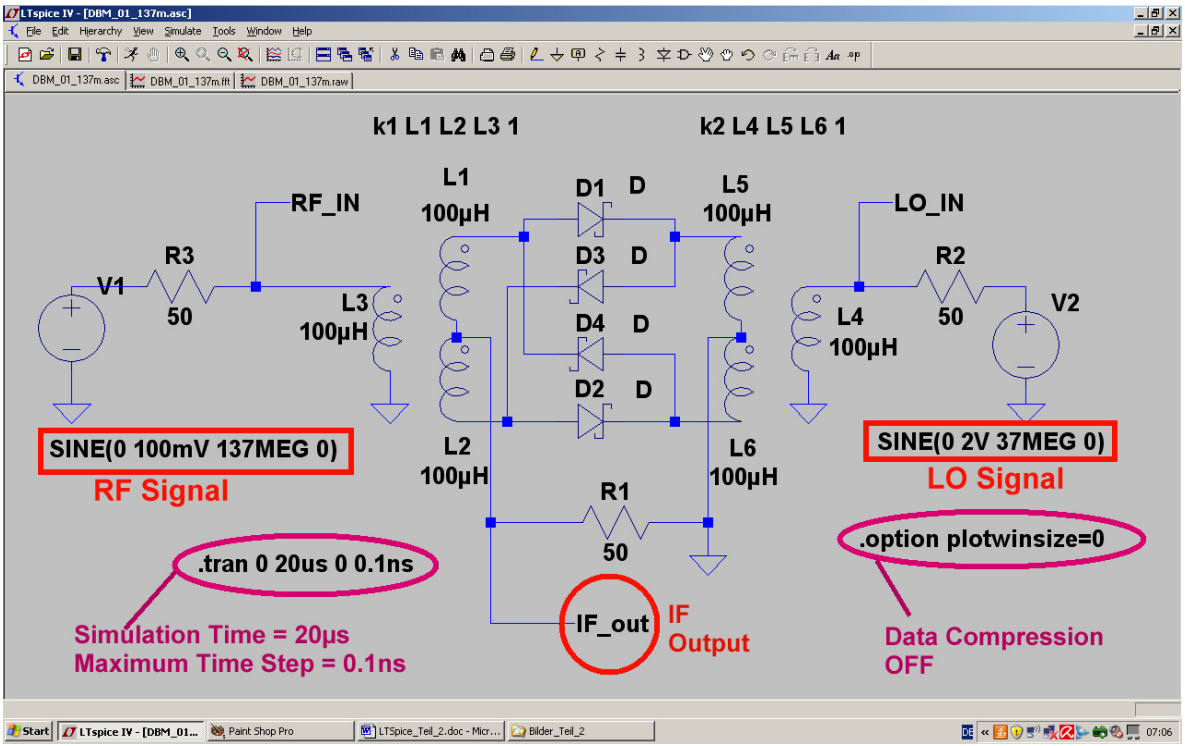
Durch das (große) LO-Signal werden bei dessen positiver Halbwelle -- also bei der Amplitude „+1“ -- die beiden in Reihe liegenden Dioden D2 und D4 eingeschaltet. Bei der negativen Halbwelle dagegen -- sie entspricht der Amplitude „-1“ -- werden die beiden Dioden D1 und D3 als elektronische Schalter betrieben. So wird das von links kommende RF-Signal entweder über die untere bzw. die obere Trafowicklung zum IF-Ausgang geleitet. Da die Spannungen dieser beiden Sekundärwicklungen jedoch gegenphasig sind, erreicht man auf diese Weise das gewünschte Umpolen des RF-Signals im Rhythmus des LO-Signals. Und genau das entspricht der Multiplikation einer Sinusspannung mit einem Rechtecksignal..

Vorteile: sehr verzerrungsarm. Bei richtiger Konstruktion sehr breitbandig (z. B. von Null bis 5 GHz, in Spezial-Microstripversionen bis über 50GHz) und auch von der Anpassung her ohne Probleme. Braucht keine Betriebsspannung, verlangt aber an allen Ports möglichst exakt den Systemwiderstand von z. B. 50 Ω als Innen- oder Abschlusswiderstand. Wird von vielen Firmen in großen Stückzahlen hergestellt und vertrieben, kostet also nicht viel. Ist im Einsatz völlig laiensicher und kann wie ein Bauklötzchen benützt werden, da er meist schon mit drei passenden HF-Buchsen (BNC- oder N- oder SMA-Steckverbindern) ausgeliefert wird.

Nachteile: wegen der fehlenden Betriebsspannung muss die „Schaltleistung“ für die Schottky-Dioden komplett vom LO-Signal aufgebracht werden -- es sind also recht große LO-Pegel nötig, um die Dioden schnell ein- und auszuschalten. Es ist keine Verstärkung möglich, deshalb besitzt der Baustein grundsätzlich eine **Dämpfung** (typisch: zwischen 5 und 8dB) und verschlechtert deshalb das Rauschverhalten eines Systems. Da kann als Konsequenz: mehr Vorverstärkung bei einem Empfänger nötig sein.

4.2. Simulation des DBM mit LTspice

4.2.1. Schaltung für die LTspice-Simulation



Schon eine feine Sache: alle benötigten Bauteile gibt es in der Bauteil-Library und man muss sich lediglich die neue Schaltung zusammenbauen. Die Symbolbezeichnung der Dioden lautet „schottky“ und nur zu den Übertragern ist etwas anzumerken:

Sie bestehen aus jeweils drei identischen Induktivitäten „ind2“, bei denen das Schaltzeichen einen kleinen Kreis zur Kennzeichnung der Polung trägt. Ihr Induktivitätswert wird in der Praxis meist so groß gewählt, dass der zugehörige induktive Blindwiderstand bei der tiefsten vorgesehenen Frequenz noch mindestens zwei- bis dreimal so groß ist wie der Systemwiderstand von 50Ω (hier sind es je 100µH).
Eine SPICE-Direktive wird zur Angabe der magnetischen Kopplung verwendet, die aus den drei einzelnen Spulen einen Transformator macht. Sie lautet für die linke Anordnung (= RF-Eingang):

k1 L1 L2 L3 1

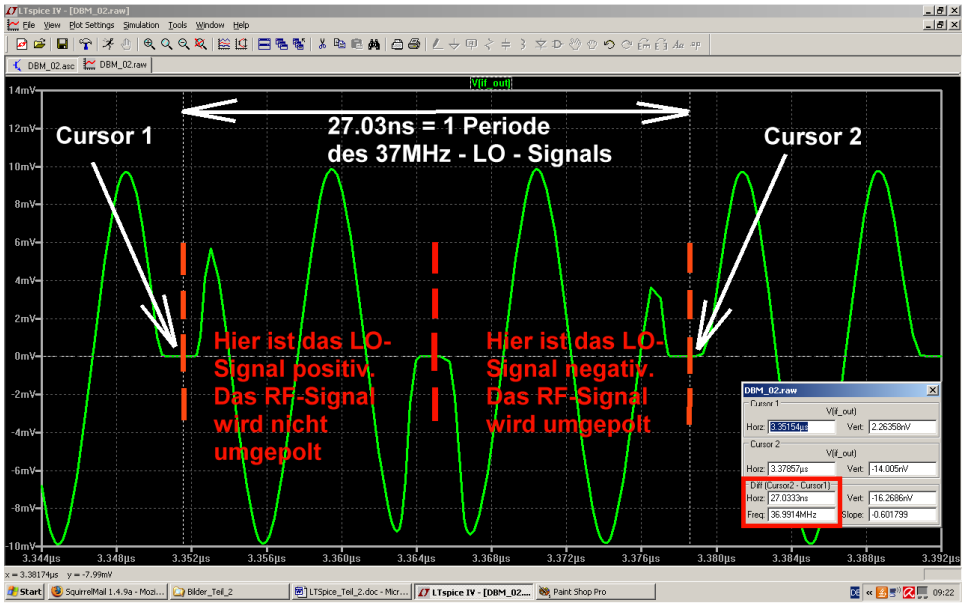
und beim dadurch festgelegten Kopplungsfaktor k1=1 haben wir einen idealen Übertrager vor uns. Für den rechten LO-Eingang gilt dasselbe.

Am linken Eingang (RF) liegt beim fertigen Konverter das Antennensignal an, das bereits von der Vorstufe um mehr als 20dB verstärkt wurde. Deshalb wählen wir bei der Mixersimulation eine sinusförmige RF-Spannung mit einem Spitzenwert = 100mV und einer Frequenz = 137MHz, um den Praxisfall nachzubilden.
Über den rechten Eingang wird der Mixer mit dem LO-Signal versorgt. Für eine vorgesehene Zwischenfrequenz von 100MHz ist ein Umsetzoszillator-Signal mit f = 37MHz nötig und mit einem Spitzenwert von 2V für die Urspannung sollte ein verzerrungsarmer Betrieb möglich sein.
Der IF – Ausgang für die gewünschte Zwischenfrequenz wird mit 50Ω abgeschlossen und einem passenden Label versehen.

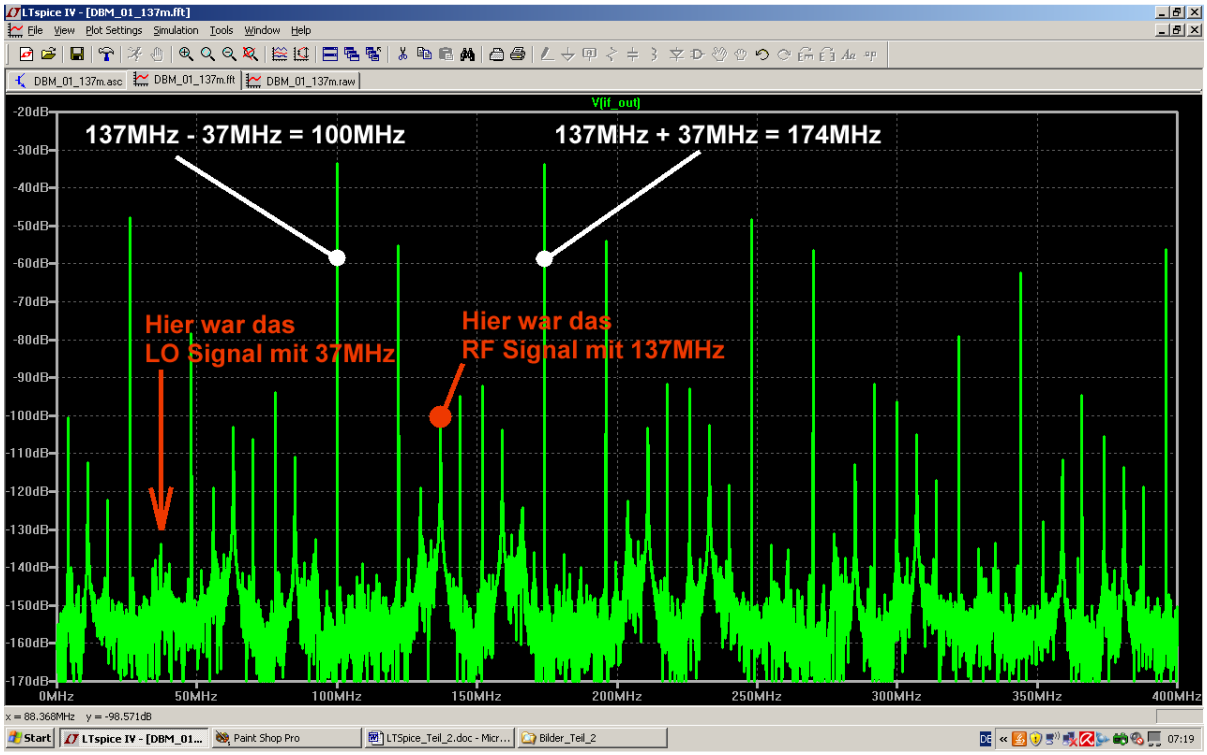
4.2.2. Simulation der Mischereigenschaften

Simuliert wird für eine Zeit von 20µs und damit erhalten wir eine Frequenzauflösung von 1 / 20µs = 50kHz. Sehen wir noch eine kleinste Schrittweite (= Maximum Time Step) von 0,1 ns vor, dann stehen uns wieder mindestens 20µs / 0.1ns = 200 000 echte berechnete Samples zur Verfügung. Damit können wir bei der FFT bis zu 131 072 Werte vorgeben.

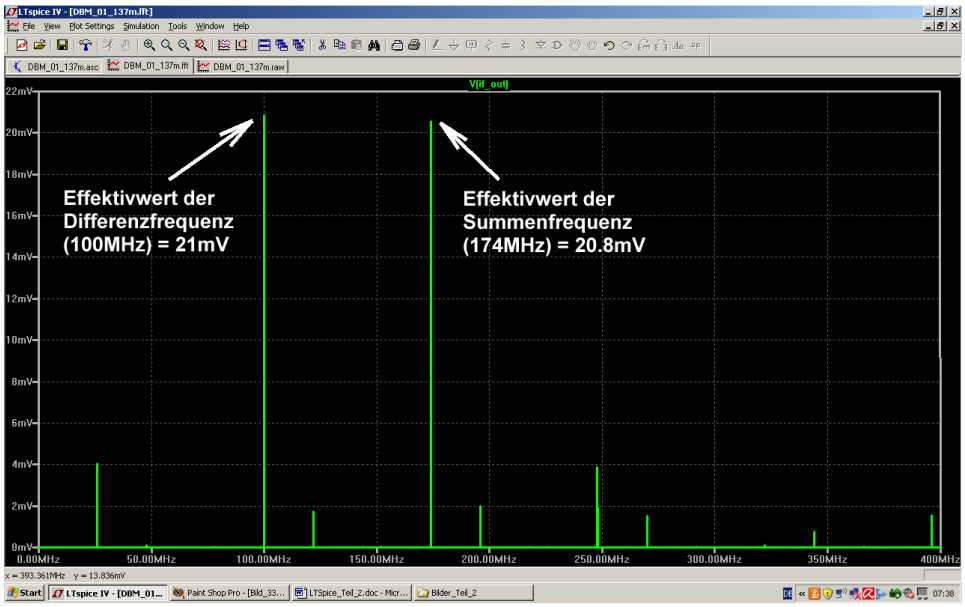
(**Für Spezialisten**: mehr Samples geben stets eine höhere Grenzfrequenz der Ergebnisdarstellung und größere Dynamik, fressen aber Rechenzeit und liefern sehr große Ergebnisdateien. Außerdem sinkt der „Grundrauschpegel bei der Spektraldarstellung“. Er entsteht einmal als Quantisierungsfehler, da eine Digitale Signalfolge halt doch immer eine Treppenfunktion im Vergleich zum Analogen Signal bildet. Zweitens haben wir noch einen weiteren Anteil durch eine spezielle LTspice-Eigenheit, nämlich die nicht konstante Samplefrequenz - - schließlich geben wir nur den „Maximum Time Step“, also die minimale Abtastfrequenz vor. LTspice darf und wird aber bei schwierigen Signalstellen öfters „sampeln“ und das wirkt sich wie eine „nicht vorhersehbare Frequenzänderung“, also z. B. wie eine Rauschmodulation, aus. Vermindern wir den Maximum Time Step, dann wird der Frequenzhub für diese Rauschmodulation kleiner und damit auch dieser Anteil am Grundrauschpegel. Aber wenn diese Frequenzänderungen regelmäßig auftreten, dann finden wir im Spektrum plötzlich zusätzliche unbekannte Linien und müssen erst mal rätseln…..).



Auch die Datenkompression wird wieder mit **.option plotwinsize=0** abgeschaltet und dann kann simuliert werden. Hier haben wir nun den Verlauf des ZF-Signals am Ausgang (IF_Out) in gedehnter Form und man kann daran sehr schön das erwähnte Umpolen des RF-Signals im Rhythmus der LO-Spannung erkennen. Und nicht vergessen, vor der Simulation nach der „Gear-Einstellung“ zu schauen….



Führen wir nun eine FFT mit 131072 Samples durch und zoomen wir den Bereich von 0….400MHz heraus, dann ist es nicht schwer, im obigen Bild die Summen- und Differenzfrequenz von RF- und LO-Signal zu identifizieren. Recht gut ist die Unterdrückung der beiden Eingangssignale, aber es tauchen auch sehr viele unerwünschte Linien auf, die beim Umschaltvorgang an den nichtlinearen Kennlinien der Dioden erzeugt werden. Bei weniger als je 0,4V an den beiden jeweils in Reihe liegenden und eingeschalteten Dioden werden die Kennlinien halt sehr krumm und wenn es an den anschließenden Nulldurchgang geht (= Umschaltung auf das andere Diodenpärchen), fehlt sogar ein kurzes Signalstück. Das gibt Verzerrungen und dieser Effekt ist auch im vorausgegangen Bild mit dem zeitlichen Verlauf der Ausgangsspannung gut zu erkennen. Da kann man vor den Entwicklern von Mischern gehörig Respekt bekommen, die hier Lösungen zur Verbesserung der Linearität suchen müssen!



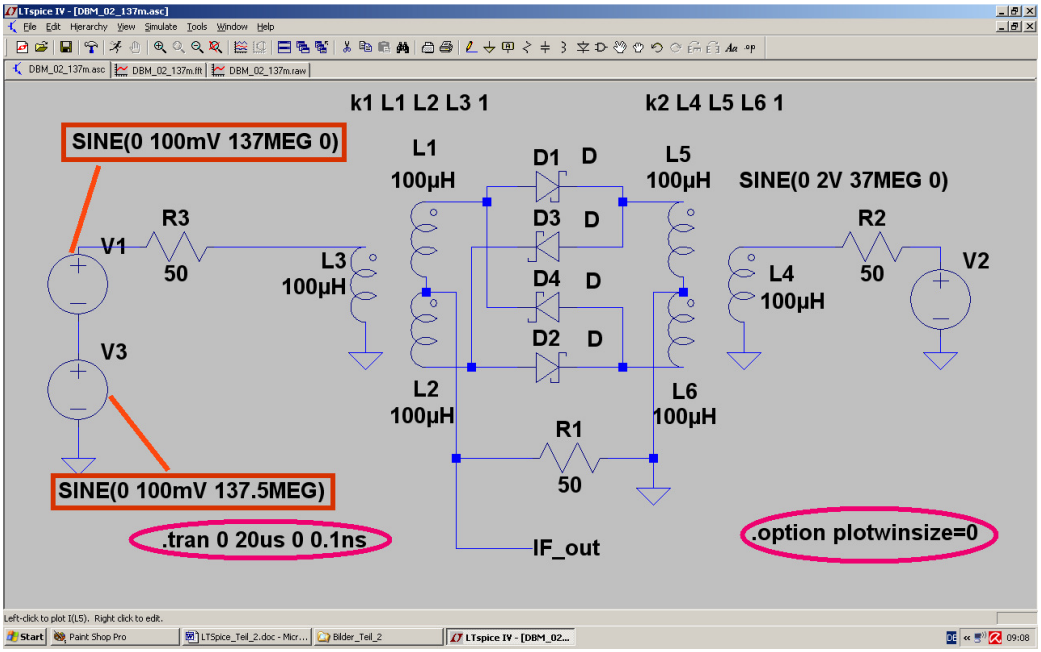
Durch eine einfache Änderung der Diagrammteilung kommen wir auch gleich zur Umsetzdämpfung. Man braucht nur die senkrechte Achse auf „linear“ umzustellen und erhält dadurch die Effektivwerte von Summenfrequenz (20,8mV) und Differenzfrequenz (21mV). Dann kann man für die Differenzfrequenz mit 100MHz folgende Rechnung aufmachen:

Der Spitzenwert der Urspannung beim angelegten RF-Signal beträgt 100mV und das entspricht einem Effektivwert von 70,7mV. Die „Hinlaufende Welle (Incident Wave) hat dann die halbe Amplitude, also 35,35mV und sie erscheint am Mischerausgang nur noch mit 21mV bei 100MHz. Damit beträgt die Umsetzdämpfung

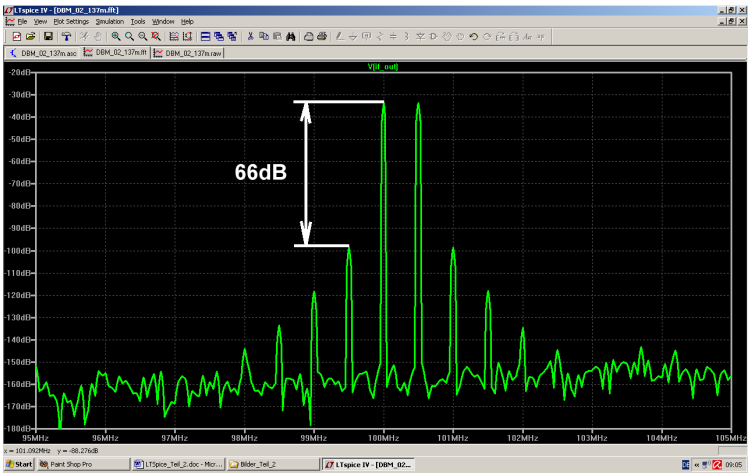
$$a = 20 \bullet \log \frac{35.35mV}{21mV} = 4{,}5dB$$

Wir haben allerdings ideale Übertrager vorgesetzt und so werden es in der Praxis leider schnell 5,5 oder sogar mehr als 6dB. Bei der Summenfrequenz ist außerdem bereits eine leichte Abnahme der Amplitude festzustellen. Das rührt vom Schwächeln der Schottkydioden bei steigender Frequenz her -- und wenn dann noch die obere Frequenzgrenze der Übertrager dazukommt, wird die Umsetzdämpfung irgendwann rasch zunehmen.

Doch nun wollen wir uns noch das Intermodulationsverhalten der Schaltung in Form der IP3-Simulation ansehen. Dazu steuern wir den Mischer am RF-Eingang wieder mit zwei „Inband-Signalen", nämlich 137MHz und 137,5MHz an, deren Amplituden mit je 100mV Spitzenwert gleich groß sind:



Im Ausgangsspektrum finden sich nun neben den beiden ZF-Signalen (bei 100MHz und 100,5MHz) die Intermodulationsprodukte dritter Ordnung, die um **66dB** gedämpft sind. Dann kommt wieder die nun schon bekannte Rechnung:



Der Pegel am RF-Eingang ergibt sich mit der Amplitude der Hinlaufenden Welle von 35,35mV zu

$$20 \bullet \log \frac{35{,}35mV}{224mV} = -16dBm$$

und damit lässt sich der IP3-Punkt bestimmen:

$$IP3 = Eingangspegel + 0{,}5(IM - Pegeldifferenz) = -16dBm + 0{,}5 \bullet 66dB = +17dBm$$

Glücklicherweise liegt der Ausgangs-IP3 des Vorverstärkers, der den Mischer speist, bei nur +10dBm und so ist es zur Abwechslung mal nicht der Mischer, der die Aussteuergrenze des Konverters bestimmt und zuerst in die Begrenzung läuft bzw. zu „klirren" anfängt.

5. Der Ausgangs-Bandpass

5.1. Vorbemerkungen

Da sollte man schon hohe Ansprüche stellen, wenn man an das fast erschreckende simulierte Ausgangsspektrum des Mischers bei Ansteuerung mit nur einem Signal denkt. Da kann die Flankensteilheit und die Selektion gar nicht groß genug sein. Bei einer Mittenfrequenz von 100 MHz sollte zusätzlich die Bandbreite nicht über 5 MHz liegen und das Alles fordert eine hohe Güte der Bauteile UND einen Erfolg versprechenden Schaltungsentwurf. Hervorragend eignet sich dazu der „Narrow Bandpass“ (…beim Ansoft Designer SV heißt er „Coupled Resonator Bandpass“) mit folgenden Vorgaben und Forderungen:

a) alle eingesetzten Induktivitäten müssen dieselbe Induktivität aufweisen (bewährt: 100nH für fo= 100MHz)

b) Die Koppelkondensatoren werden als Interdigitalkondensatoren ausgeführt und so ist keine zusätzliche Bestückung erforderlich.

c) Die Bandbreite soll nicht mehr als 5 MHz betragen, aber 10MHz neben der Mittenfrequenz sollte bereits eine Dämpfung von ca. 70dB vorhanden sein.

d) Die Weitab-Selektion bei 1 GHz darf nicht unter einen Wert von 70dB sinken.

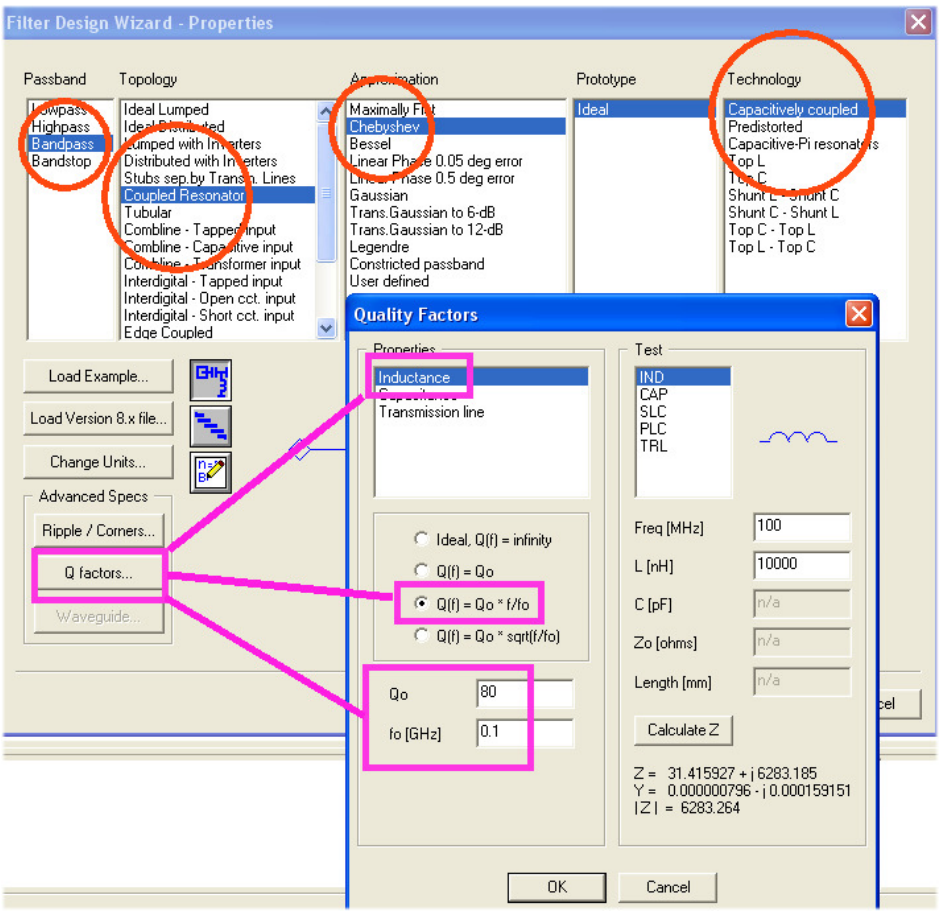
e) Ein problemloser Betrieb im 50Ω-System muss möglich sein.

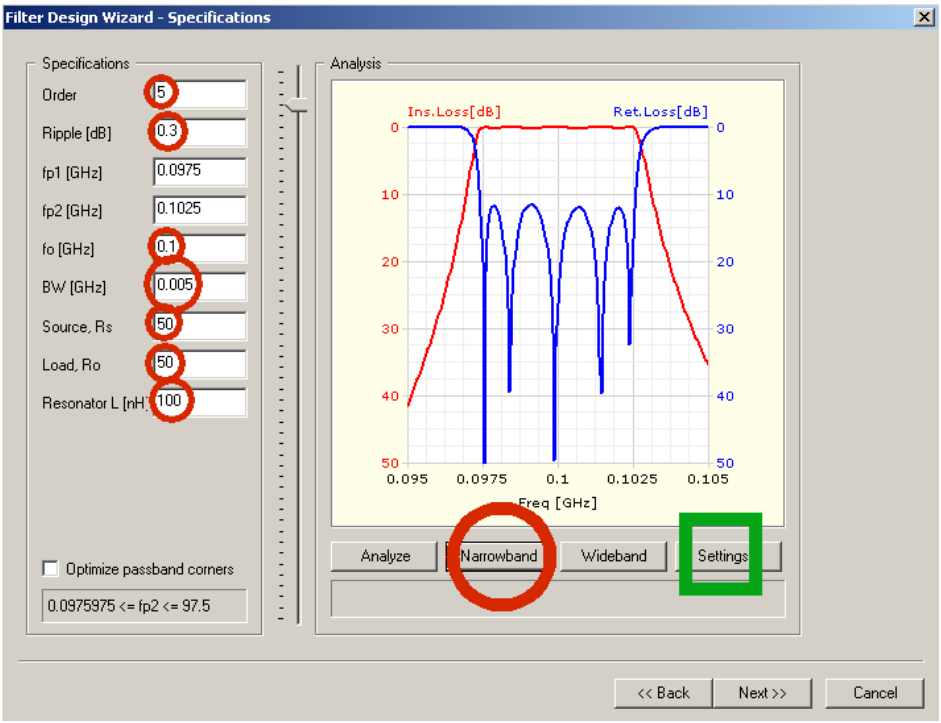
Der Autor hat das Vorgehen beim Bau einen solchen Bandpasses für 100MHz in {5] genau beschrieben, aber die Situation hat sich in Bezug auf die zur Verfügung stehenden Simulationsmöglichkeiten in der Zwischenzeit doch sehr angenehm für den „Normalverbraucher“ entwickelt. Deshalb soll ein Entwurf nach dem heutigen Stand demonstriert werden und dafür folgt nun das Pflichtenheft.

Filtertyp	Tschebyschef – Bandpass, Coupled Resonator Typ
Systemwiderstand	Z = 50Ω
Filtergrad	n = 5
Mittenfrequenz	100MHz
Passband Ripple	0,3dB
Ripple – Grenzfrequenz	5MHz
Kreiskondensatoren	SMD 0804, Werkstoff NP0
Induktivitäten	Neosid Helixfilter im versilberten Abschirmgehäuse. L = 100nH, Q = 80 (selbst gemessen bei 100MHz)
Koppelkondensatoren	Gedruckte Interdigitalausführung
Platinenmaterial	Rogers RO4003, Dicke = 32MIL = 0,813mm, 35µm Kupfer, er = 3,38

5.2. Schaltungsentwurf mit Ansoft Designer SV

Immer wieder muss man die Firma und das Programm loben, denn außer der kompletten linearen S-Parameter-Simulation samt Hunderten von bereitgestellten Bauteilmodellen gibt es einen hervorragenden Filter Calculator -- natürlich ebenfalls kostenlos.
Auch der Entwurf der Interdigital-Koppelkondensatoren ist problemlos möglich. Er erfordert aber einige Tricks, um an den genauen Wert der Koppelkapazität heranzukommen und ihn von den unvermeidlichen und zur Ersatzschaltung gehörenden Parallelkondensatoren abzutrennen, weil das Programm nur die S-Parameter des Bauteils simuliert. Wie man das genau macht, steht in einer weiteren Veröffentlichung des Autors [6] -- bitte dort nachlesen.
Sehen wir uns nun mal den Filterentwurf mit dem Designer an, denn das ist die reinste Freude. Dazu wird das Programm gestartet und unter „Project“ die Option „Insert Filter Design“ gewählt. Nacheinander werden zuerst die Angaben zum Filter (Bandpass / Coupled Resonator / Chebychev / Capacitively coupled) eingetippt und schließlich „Q Factor“ gedrückt (**Siehe nächste Seite)**. Da sorgt man bei den Induktivitäten für eine Güte von Q = 80 bei 100MHz mit linearer Frequenzabhängigkeit (**Siehe nächste Seite**).





Wenn man nun „Weiter“ drückt, dann wird man von der Vorgabeliste für die Filterparameter empfangen.
Folgende Einträge muss der Benutzer vornehmen:

Filtergrad: 5

Ripple: 0.3dB

Mittenfrequenz fo = 0.1GHz

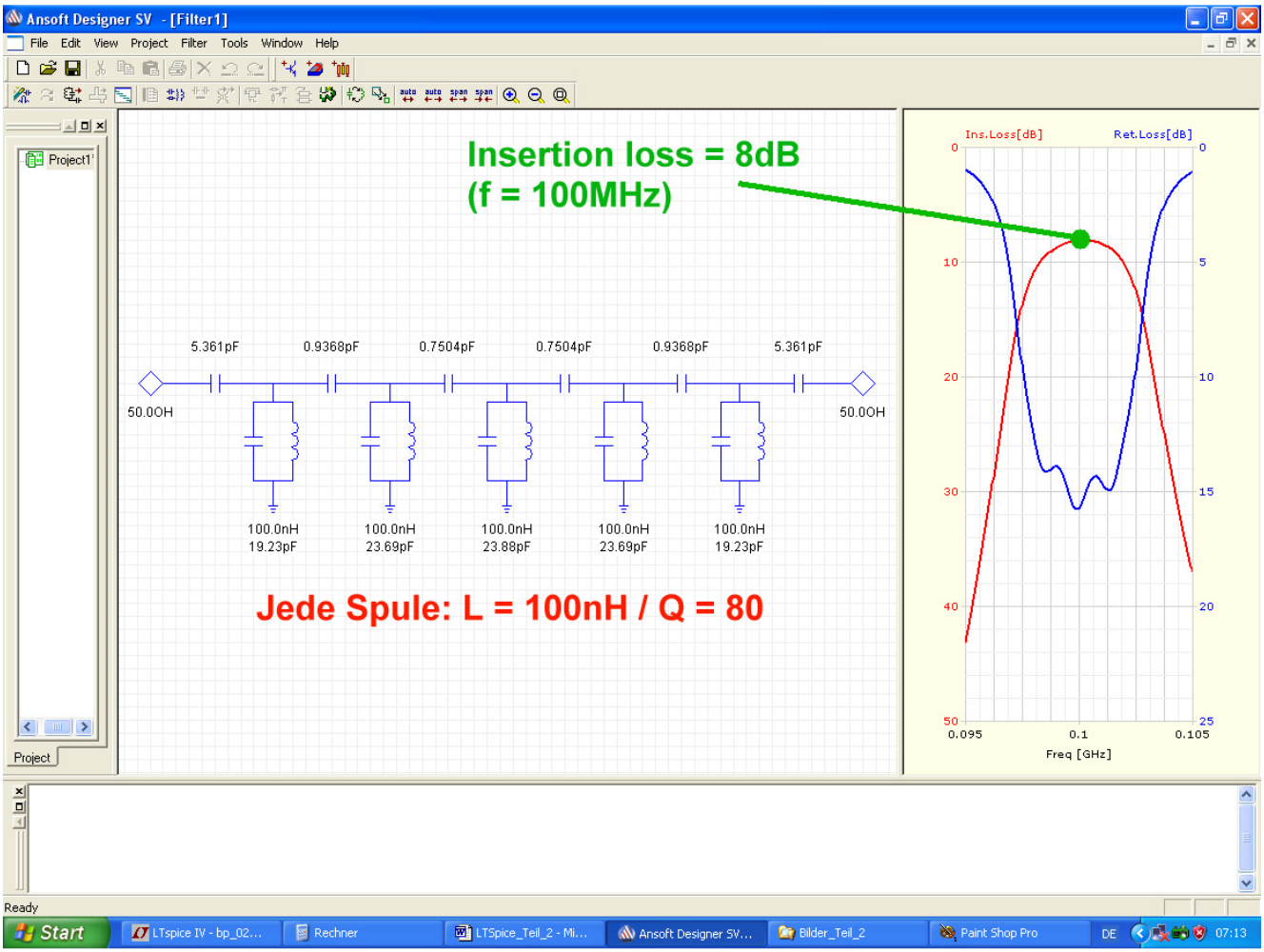
Bandbreite BW = 0.005GHz

Sourcewiderstand Rs = 50Ω

Lastwiderstand Ro = 50Ω
Resonator L = 100nH

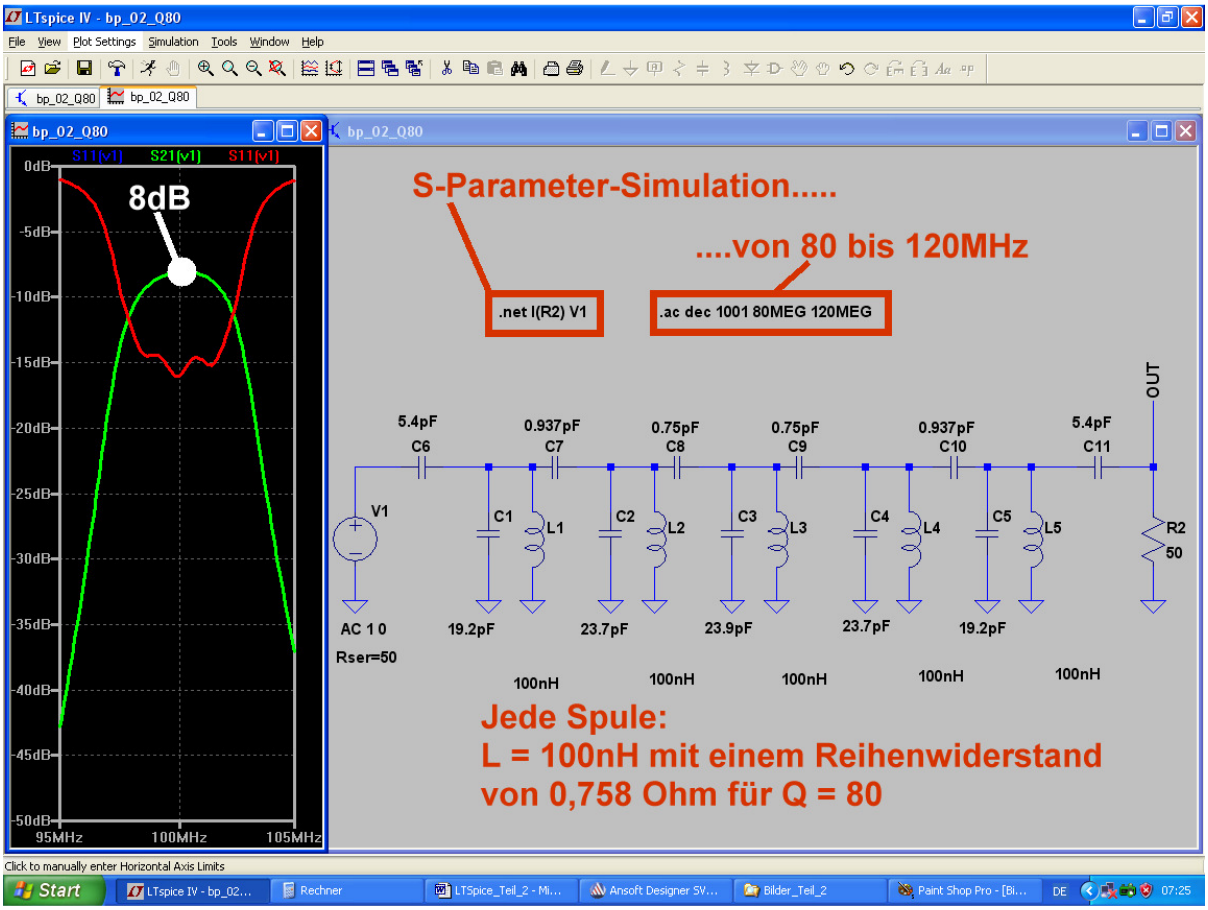
Die zusätzlich in der Liste enthaltenen Grenzfrequenzen fp1 und fp2 berechnet das Programm selbständig aus der Mittenfrequenz und der Bandbreite und trägt es anschließend in die zugehörigen Felder ein, wenn man auf „Narrowband““ klickt. Dann tauchen auch gleich alle S-Parameter im Diagramm auf -- aber für eine ideale Schaltung ohne Verluste!

Wenn man die einbeziehen möchte (und das muss man für die Praxis), dann reicht ein Klick auf „**Settings**“ sowie ein Häkchen bei „**Include Q factor losses in the response**“. Mit „Weiter“ und „Fertigstellen“ kommt man schließlich zur kompletten Filterschaltung einschließlich S11- und S21-Verlauf. Typisch für so schmalbandige Filter ist leider die starke Einfügungsdämpfung (Insertion Loss) von 8dB als Folge der Spulenverluste.

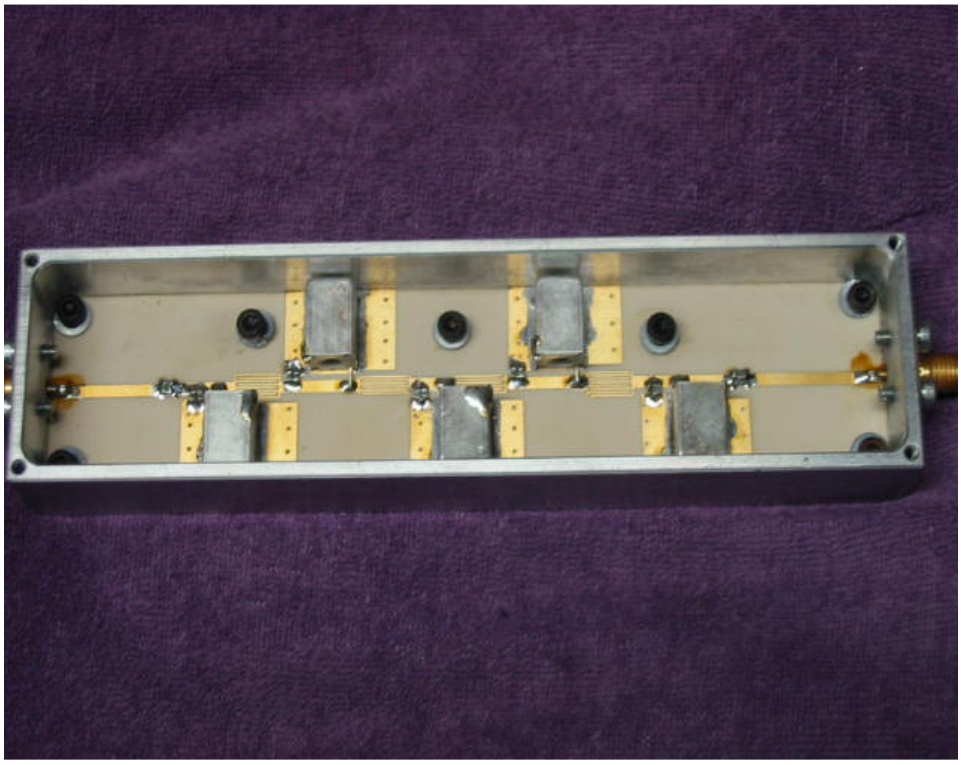


Zum Abschluss wollen wir nachprüfen, was LTspice zu diesen Kurven sagt. Dazu zeichnen wir die Schaltung, sehen eine S-Parameter-Simulation im gleichen Frequenzbereich vor und berücksichtigen die Spulengüte in Form eines Reihenwiderstandes zu jeder Spule mit dem Wert

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



Hier das Ergebnis der S-Parameter-Simulation, wobei die Werte der Kondensatoren, die diskret (ggf. als Parallelschaltung mehrerer SMD-Bauteile) eingelötet werden, gleich auf eine Stelle nach dem Komma gerundet wurden. Vergleicht man das mit dem Ansoft-Ergebnis, dann kann man doch zufrieden sein.



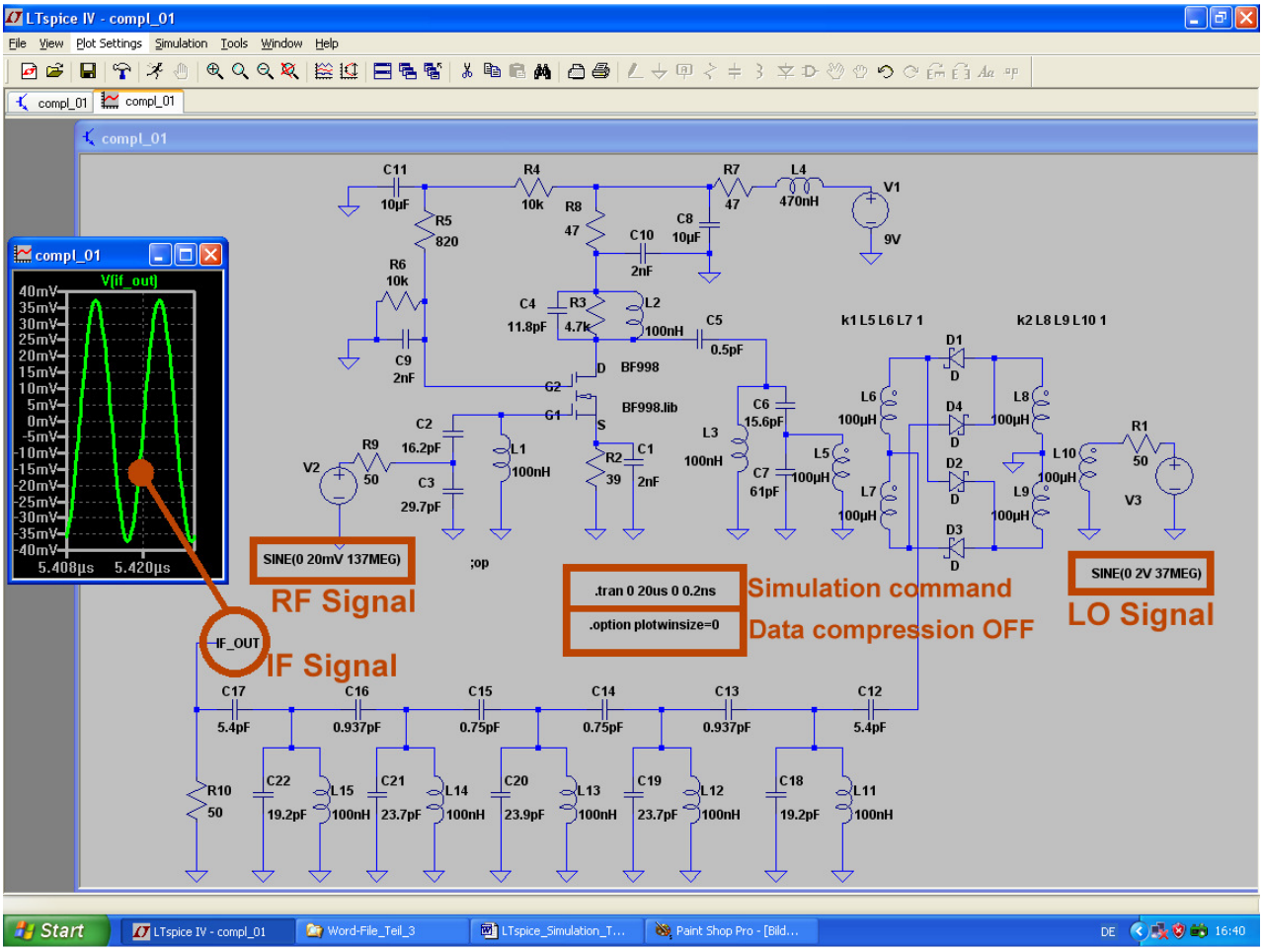
Und hier als kleines Bonbon ein Foto des fertigen Bandpasses bei abgenommenem Deckel.

Die Interdigitalkondensatoren sind gut zu erkennen. Ebenso die „Masse-Inseln mit Durchkontaktierungen bei jedem der fünf Schwingkreise. Die als Kreiskondensatoren eingelöteten SMD-Kondensatoren sind nur beim genauen Hinsehen zu erkennen.

6. Untersuchung der Gesamtschaltung

6.1. Vorbereitung

Sie beginnt mit einem gehörigen Stück Arbeit, denn die komplette Schaltung, bestehend aus Vorstufe, Mischer und Bandpass, muss in den LTSpice-Editor eingegeben werden. Da fängt man am besten mit der (hoffentlich noch als LTspice – Projekt vorhandenen) Vorstufe an, die man unter einem neuen Namen abspeichert und um die folgenden beiden Baugruppen ergänzt. Der Ausgang bekommt den Label „IF_OUT“, am RF-Eingang wird ein Sinussignal mit 20mV Spitzenwert und der Mittenfrequenz von 137MHz angelegt. Das LO-Signal ist ebenfalls ein Sinus, aber mit 37MHz und 2V Spitzenwert. Die Betriebsspannung der Vorstufe beträgt +9V.



Simuliert wird für 20µs bei einem Maximum Time Step von 0,1ns und damit stehen 200µs / 0,1ns = 200 000 echte Samples für die FFT zur Verfügung. Das Abschalten der Datenkompression darf wieder nicht vergessen werden und als Abschluss folgt wieder einmal der eindringliche Aufruf zu diesem lebenswichtigen Check:

Man drückt den Button mit dem Hammer (= Control Panel) und geht auf die Karteikarte „SPICE“. Dort MUSS oben links bei „Default Integration Method“ das Häkchen bei „Gear“ stehen. Sonst gibt es bei diesen hohen Frequenzen und dieser komplexen Schaltung massiven Ärger einschließlich unendlich langer Simulationszeiten oder sogar einen Abbruch! Bitte diese Prüfung immer wieder vornehmen -- besonders, wenn ein Projekt angelegt oder geändert oder etwas neu gezeichnet wurde, wird das vom Programm leider ohne Vorankündigung geändert!

6.2. Bestimmung der Gesamtverstärkung bei Mittenfrequenz

Da braucht man sich nur den Spitzenwert des Signals IF_OUT aus dem obigen Bild zu holen -- er beträgt 37,5mV. Dazu gehört ein Spitzenwert der Urspannung am RF-Eingang von 20mV und eine Hinlaufende Welle mit 10mV Spitzenwert.

Das ergibt ein S21 des Konverters von 37,5mV / 10mV = 3,75 bzw. 11,5dB.

Man sollte jedoch stets eine Überschlagsrechnung mit den in den vorausgegangenen Kapiteln ermittelten S21-Werten der einzelnen Baugruppen vornehmen und prüfen, ob das stimmen kann:

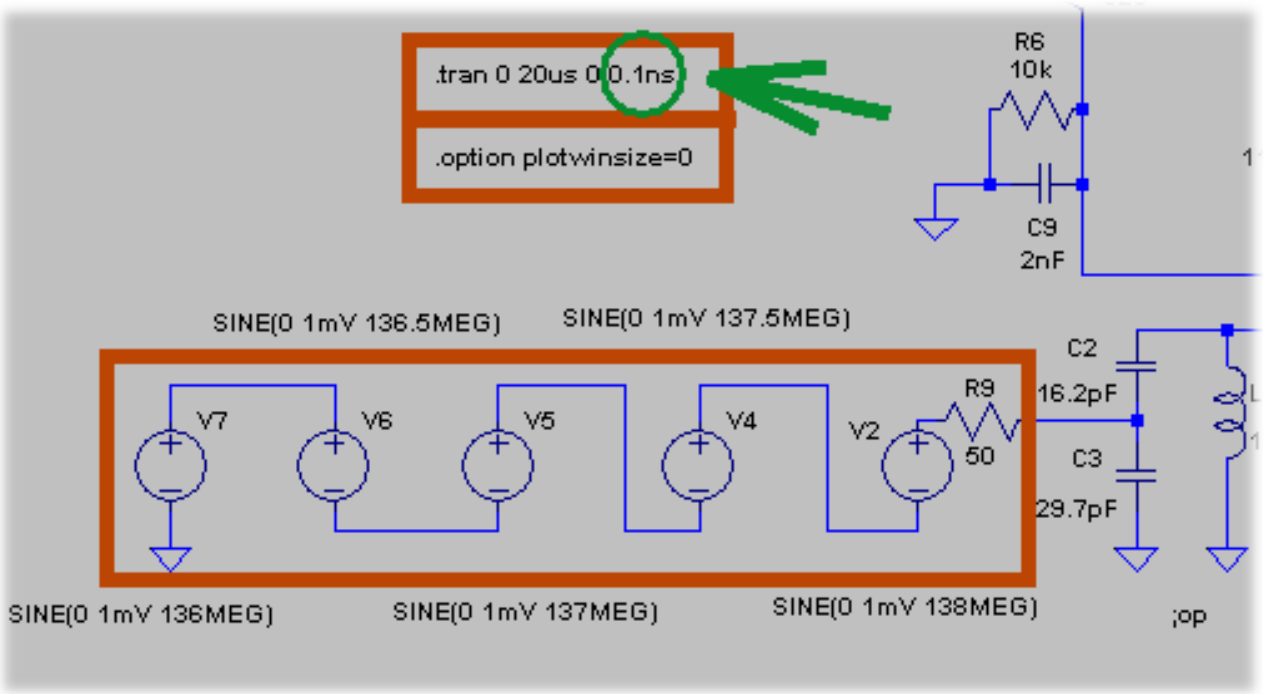
S21 in dB = Vorstufenverstärkung – Mischerdämpfung – Filterdämpfung = 25dB – 4,5dB – 8dB = 12,5dB.

Hier gibt es nichts zu jammern, denn die Abweichung von 1 dB stammt wohl von Fehlanpassungen zwischen den einzelnen Stufen. Außerdem werden es in der Praxis sicher noch 1….1,5dB weniger sein, denn da war doch was mit den in der Simulation als ideal angenommenen Übertragern beim Mischer…..

6.3. Ermittlung der Durchlaßkurve des Konverters

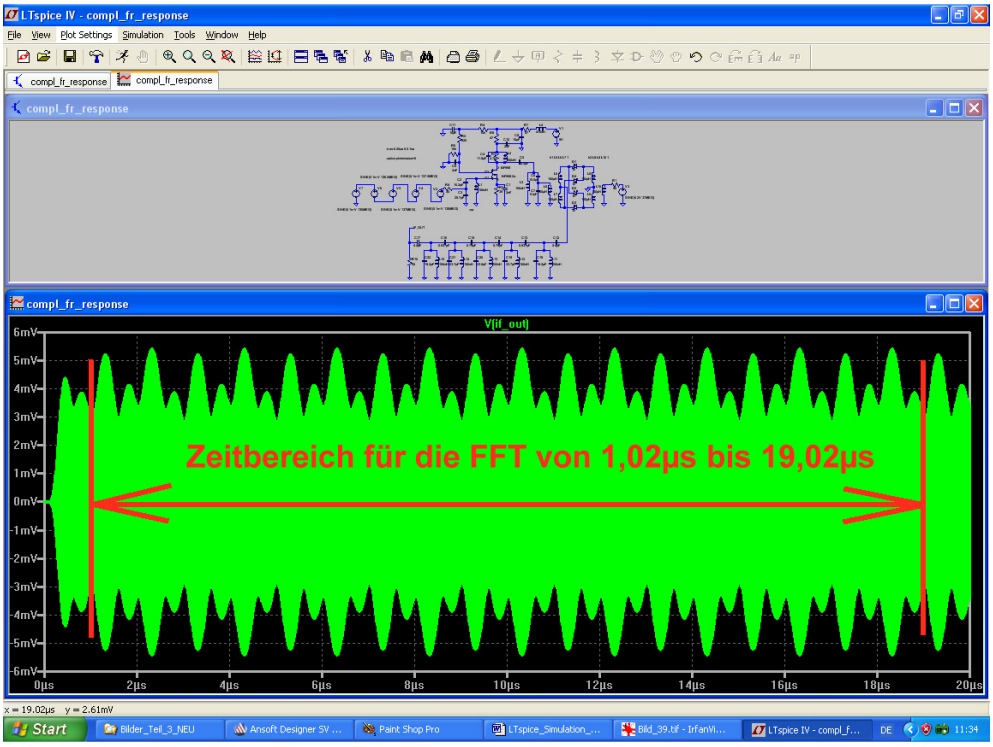
6.3.1. Simulation mit einzelnen Spektrallinien

Gar nicht so einfach, denn der AC-Sweep fällt hier aus -- wir haben es nämlich mit einer Frequenzumsetzung durch den DBM zu tun und da spielt sich alles im Zeitbereich ab.



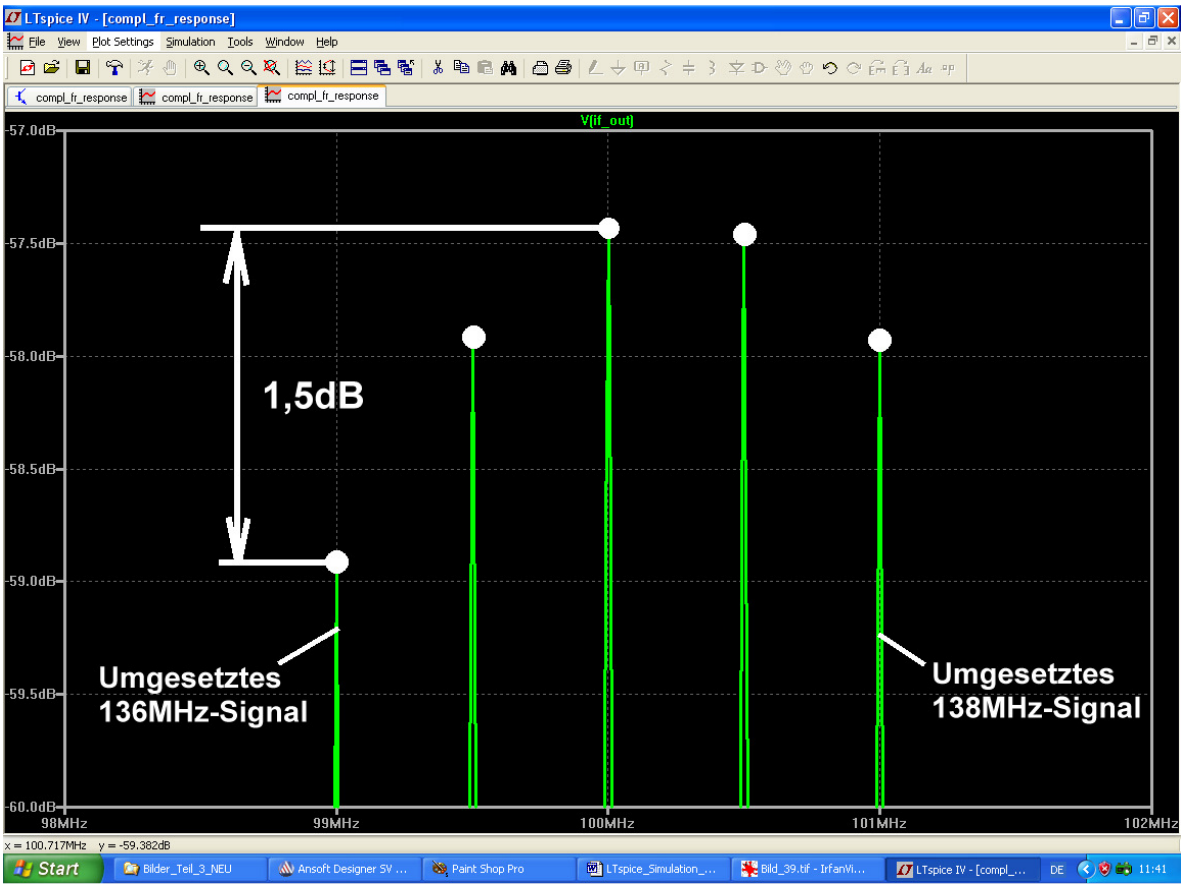
Also hecken wir wieder einen Trick aus und speisen den RF-Eingang mit einer ganzen Palette von gleich großen Signalen, die den Durchlassbereich abdecken, nämlich 136 / 136,5 / 137 / 137,5 / 138MHz. So erkennt man beim Ergebnis sofort, was los ist. Im obigen Bild ist der Schaltbildauszug mit der erforderlichen Änderung zu sehen und bei jedem Einzelsignal wurde ein recht kleiner Spitzenwert von 1mV gewählt, um Übersteuerungen oder Intermodulationseffekte zu vermeiden.

Bei einem Maximum Time Step von 0,1ns und einer Simulationszeit von 20µs erhalten wir wieder mindestens 20µs / 0,1ns = 200 000 echte Samples und können bei der FFT 131 072 Werte zulassen. Und dann noch die üblichen Spielereien: erstens die Datenkompression ausschalten und zweitens die „Gear-Kontrolle“ nicht vergessen...



Für die FFT blenden wir wieder die „Einschwingzeit“ am Anfang des Diagramms aus und achten peinlich darauf, dass von „Nulldurchgang bis Nulldurchgang“ simuliert wird. Das gibt einen Zeitabschnitt für die FFT **von 1,02µs bis 19,02µs, also 19 Mikrosekunden**.

Die FFT liefert ausreichende Informationen über die Gesamt-Durchlaßkurve mit einem maximalen Pegelunterschied von ca. 1,5dB.



Der Praktiker würde hier natürlich am liebsten sofort abgleichen, um die etwas zu hoch liegende Mittenfrequenz der Durchlasskurve zu korrigieren….

6.3.2. Frequenzgangbestimmung über einen Dirac-Impuls

Das ist ganz moderne Technik, die erst durch die rasanten Weiterentwicklungen der Kommunikations- und Rechnertechnik samt gewaltigen Speichermöglichkeiten richtig an Boden gewonnen hat. Wir wollen mal sehen, ob wir mit dieser Methode auf dieselben Ergebnisse kommen wie in Kapitel 6.3.1.
Dazu speisen wir den Eingang des Konverters mit einem „Dirac-Impuls". Das ist etwas ganz Verrücktes, nämlich ein Puls mit unendlich großer Amplitude, dafür aber unendlich kurzer Pulslänge. Aber mit einer konstanten Pulsfläche von „1" (also z. B. 1Volt x 1Sekunde). Ein solcher Puls weist ein konstantes Amplitudenspektrum (= konstante Spektrale Leistungsdichte) bei allen Frequenzen auf und das wirkt sich auf unsere Schaltung so aus, als ob wir einen Wobbelgenerator mit konstanter Ausgangsspannung an die Schaltung anlegen und dann den kompletten Frequenzbereich durchfahren würden.
Praxisgerecht wird das Ganze erst dadurch, dass wir zur Messung auch einen „normalen" (= angenäherten) Dirac-Puls verwenden dürfen Dessen Pulsdauer muss aber mindestens kürzer sein als:

$$t_{Puls} = 0{,}01 \bullet \frac{1}{\text{Systemgrenzfrequenz}}$$

und damit haben wir bis zur Grenzfrequenz des Systems eine praktisch konstante Spektrale Leistungsdichte -- das System merkt dann nicht mehr, dass das gar kein echter Dirac-Impuls ist! Der Trick dabei ist nun, dass wir bei einem linearen System auch die Pulsamplitude soweit verkleinern dürfen, dass es der Schaltung nicht schadet, denn die „Impulsantwort" fällt dann zwar entsprechend kleiner aus, weist aber weiterhin denselben prinzipiellen Verlauf auf. ***Und aus dieser Impulsantwort erhalten wir durch eine mathematische Behandlung namens „Convolution" den gesuchten Frequenzgang!***

Praxistest:

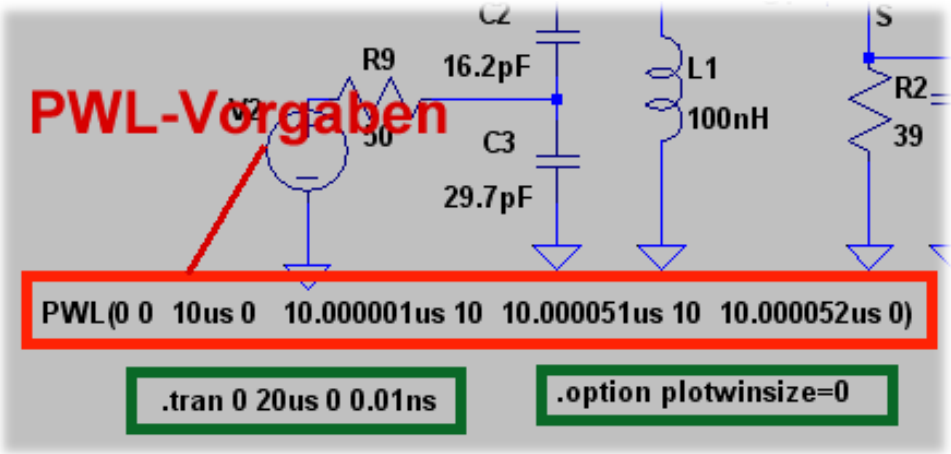
Unser Konverter soll bis **200MHz** getestet werden. Dann muss die Pulslänge kleiner sein als 1% von 1 / 200MHz = 1% von 5ns = **50 Picosekunden**.

Die **Anstiegs- und Abfallzeit** des Pulses wählen wir zu je 1…2% der Pulslänge, also z. B. zu je **1 Picosekunde**.

Eine **Pulsamplitude von 10V** dürfte der Schaltung noch nicht schaden und man sollte den Puls erst anlegen, wenn alles nach dem Einschalten zur Ruhe gekommen ist. Bei einer **Simulationsdauer von 20 Mikrosekunden** (…das gibt eine Frequenzauflösung von 1 / 20µs = **50kHz**) bauen wir einfach eine Verzögerung von 10µs ein. Lässt man nun nur einen **Maximum Time Step von 10 Pikosekunden** zu (…damit der Dirac-Impuls mit seinen 50 Pikosekunden Pulslänge nicht vom Programm übersehen wird...), dann brauchen wir nur noch die Datenkompression auszuschalten und es kann -- nach der „Gear-Kontrolle" -- losgehen.

Aber Vorsicht:
Erstens gibt das eine **sehr lange Rechenzeit** (die Simulation der Ausgangsspannung dauerte beim Autor 20 Minuten mit einem 2GHz-Rechner!) und eine **riesige Ergebnisdatei mit etwas mehr als 1 Gigabyte!** Schließlich haben wir bei diesen Einstellungen mindestens 2 000 000 Samples zu ermitteln -- und wenn man dann die Schaltung ansieht und überlegt, wie viel Spannungs- und Stromwerte pro Messpunkt berechnet und gespeichert werden müssen, dann ist das kein Wunder.

Sehen wir uns nun dieses Bild an.



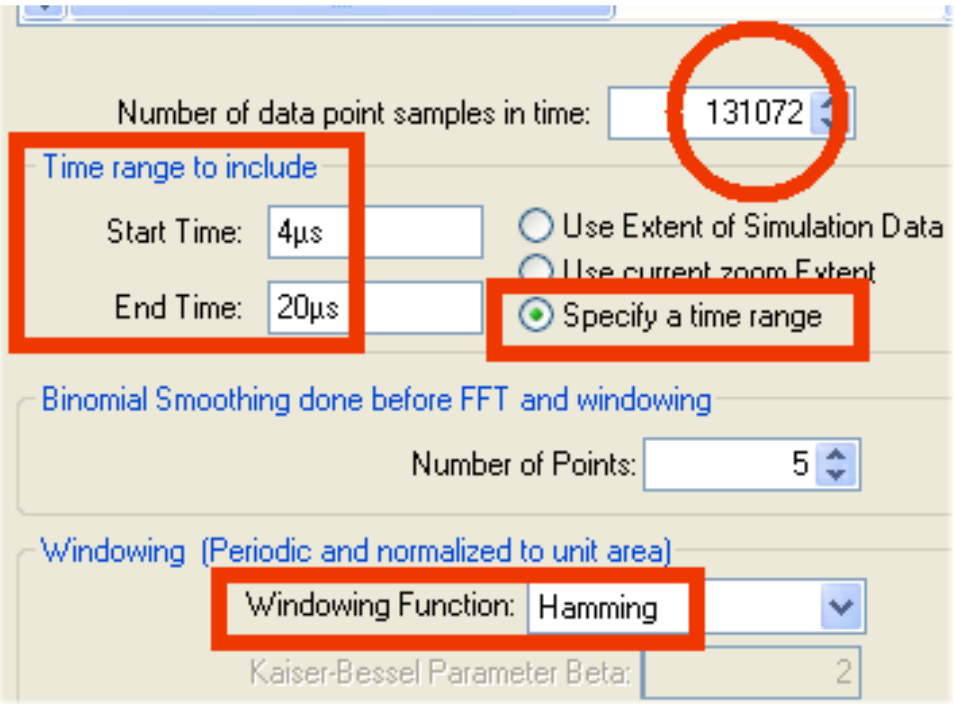
Da ist die Programmierung des Dirac-Impulses in Form eines PWL-Signals zu sehen, außerdem natürlich die Simulationsvorgaben und das Abschalten der Datenkompression. Zum Abschluss muss wie immer ein Blick hinter die „Taste mit dem Hammer" (= Control Panel) folgen, ob auf der Karteikarte „SPICE" wirklich die Integrationsmethode „Gear" aktiviert ist. Bei dieser aufwendigen Simulation wäre da der Ärger vorprogrammiert.

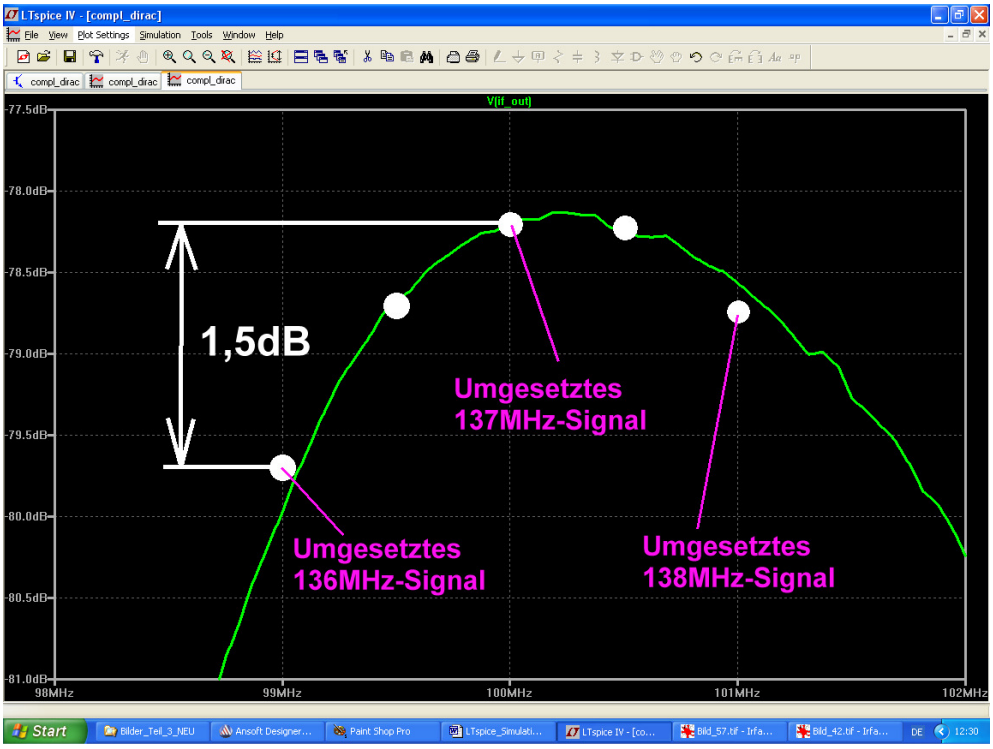
Wenn alles stimmt, dann wird gestartet und geduldig gewartet. Ein Blick auf die Ausgangsspannung ist dann interessant, denn außer unserer Impulsantwort bei 10µs gibt es bis 3µs einige Bewegung am Ausgang, denn da wird eingeschaltet, die Kondensatoren laden sich auf, Ströme steigen an, Spannungen pendeln sich im Arbeitspunkt ein……dieser Einschwingvorgang interessiert uns nicht und wir müssen ihn bei der FFT ausblenden.

Hier sehen wir das Einstellungsmenü für die FFT und da ist wichtig,

a) dass wir diesmal nur den **Bereich von 4µs bis 20µs** wählen dürfen und

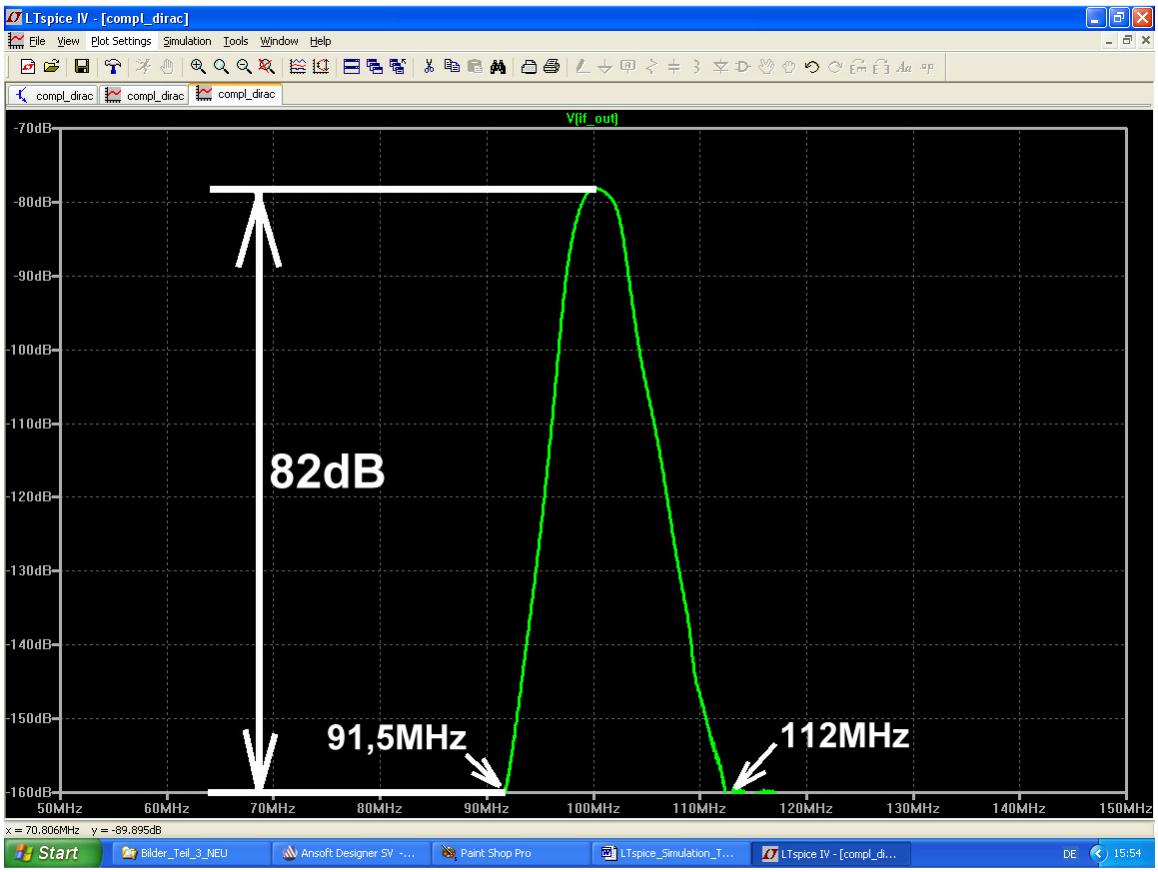
b) wir dann über das ganze Datenpaket eine „**Fensterfunktion nach Hamming**“ legen, um die abrupten Änderungen am Anfang und Ende (= Erst mal nix, dann plötzlich volle Pulle. Am Ende genau umgekehrt) und die dadurch verursachten Fehler zu dämpfen.





Nun kommen wir endlich zur FFT mit 131 072 Punkten und das Ergebnis ist richtig interessant, weil derselbe Frequenz- und Amplitudenbereich einstellt wurde wie bei der Simulation mit einzelnen Spektrallinien im vorigen Kapitel.

Wenn wir die Messwerte aus dem vorigen Kapitel in dieses neue Diagramm eintragen, dann zeigen sich wirklich nur noch minimale Unterschiede zwischen beiden Methoden.



Aber auch die simulierte Weitabselektion ist die reinste Freude.

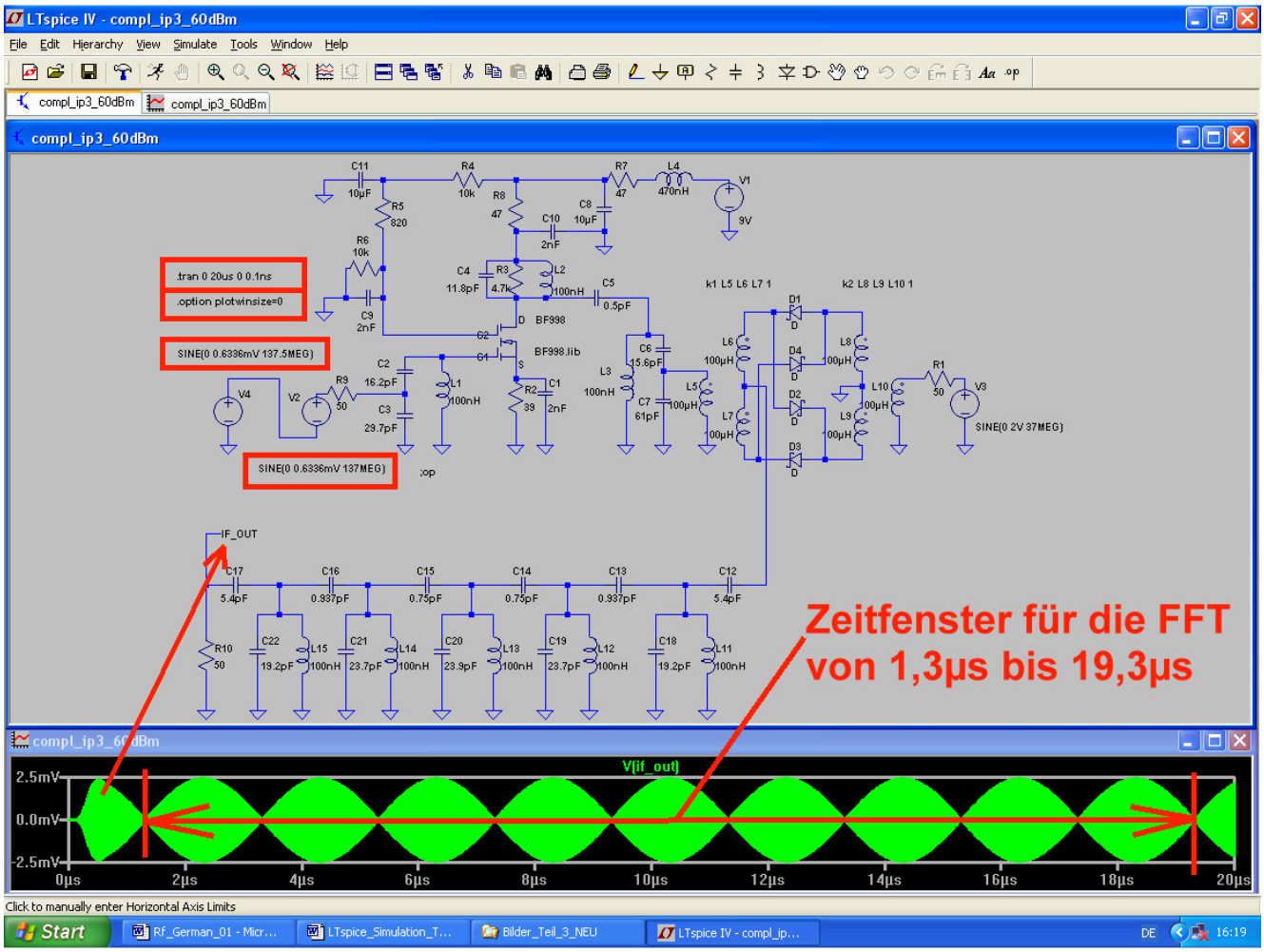
Schon eine faszinierende Sache, so ein Dirac-Impuls!

6.4. Großsignalverhalten und IP3-Punkt

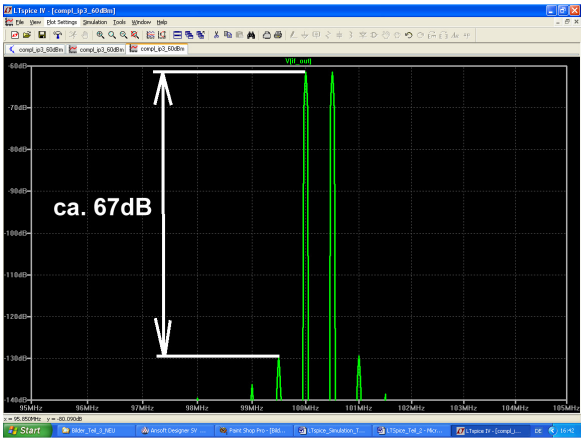
Das ist plötzlich gar nicht mehr so einfach, da wir in Form der Vorstufe und des Mischers nun zwei Baugruppen in der Schaltung haben, die bei steigender Aussteuerung zu immer stärkeren Verzerrungen neigen. Eine ganz neue, hervorragende Publikation [4] widmet diesem Problem ein eigenes Kapitel und zeigt auf, dass da verrückte Effekte auftreten können -- weil die einzelnen Stufen unterschiedliches Verzerrungsverhalten und unterschiedliche Begrenzungs-Einsatzpunkten aufweisen (da nehmen z. B. die IP3-Produkte nicht mehr weiter zu, wenn man das Eingangssignal weiter steigert, weil nämlich die Vorstufe oder der Mischer bereits in die Begrenzung gelaufen sind. Oder ein IP3-Signal geht plötzlich in der Amplitude stark zurück und das andere steigt, weil z. B. die beiden ansteuernden Eingangssignale durch irgendwelche Filterflanken unterschiedlich in den Amplituden bewertet werden. Oder es treten in den einzelnen Baugruppen aussteuerungsabhängige Phasendrehungen auf, die in der Summe der Verzerrungen plötzlich ein IP3- Produkt fast zum Verschwinden bringen können usw.). Das Studium dieses Buches ist deshalb wirklich interessant und empfehlenswert!

Das Ganze wird aber sehr schnell unübersichtlich und letztlich bleibt nur ein vernünftiger Weg:

Man beginnt die IP3-Simulation bei so kleinen Eingangspegeln, dass der Intermodulationsabstand rund 60dB beträgt und erhöht in weiteren Simulationen diese Eingangspegel immer um 10dB. Nach jeder Simulation wird kontrolliert, ob der Eingangspegel-Anstieg von 10dB tatsächlich den IP3-Abstand um 20dB vermindert.
Wir beginnen mit einem **Eingangspegel von -60dBm** -- dazu gehören ein Effektivwert von 0,224mV und ein Spitzenwert von 0,3168mV. Folglich müssen wir bei den beiden Signalquellen für die IP3-Messung die doppelte Amplitude als Spitzenwert eintragen -- das wären **0,6336mV**.
Die Signalfrequenzen betragen **137MHz und 137,5MHz**, damit weisen sie einen Frequenzabstand von 500kHz auf. Simulationszeit und Maximum Time Step bleiben unverändert (20µs mit einer Auflösung von 0,1ns) und wir erhalten wieder 200 000 echte Samples. Die Datenkompression ist abgeschaltet und die „Gear-Einstellung“ muss wieder geprüft werden.

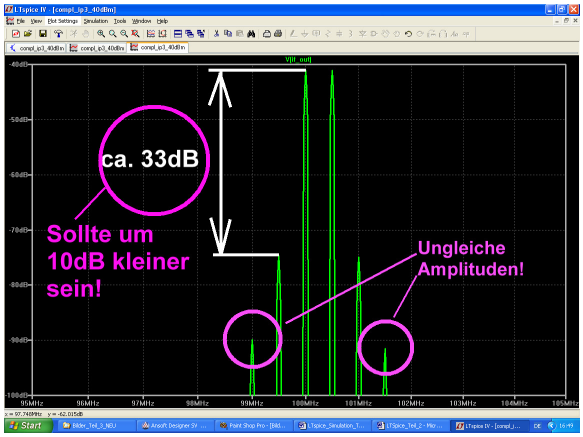


Damit fangen wir an: mit der schon bekannten Gesamtschaltung und den beiden in Reihe geschalteten Signalquellen (137MHz und 137,5MHz. Eingangspegel für den Konverter je -60dBm. **Das korrekte Zeitfenster für die FFT wurde zu 1,3µs bis 19,3µs festgelegt.**

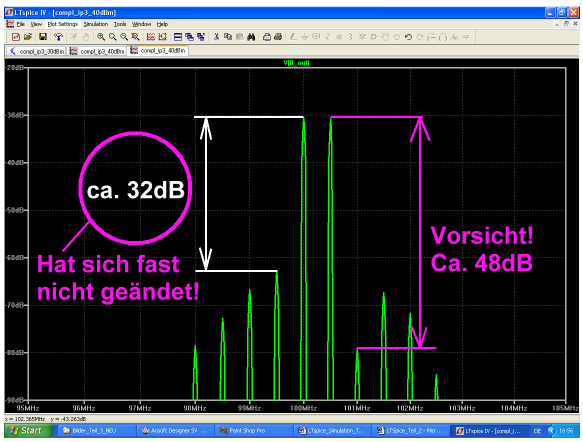


So sieht das für einen Eingangspegel von **-60dBm** aus. Damit bekommt man bei 131 072 Samples einen Abstand von ca. 67dB zwischen Eingangssignal und IP3-Produkt.

Jetzt wiederholt man die Simulation mit einem Spitzenwert der Eingangssignale, der um 10dB größer ist (also mit einem „Antennenpegel“ von **-50dBm**). Das sind 2 mV Spitzenwert bei den Eingangsspannungen und dadurch sinkt der Intermodulationsabstand auf von 67dB auf 45dB. Das läuft ja noch einigermaßen nach den bekannten Gesetzen ab, denn der theoretische Wert wäre 47dB.



Aber wenn man den Eingangspegel beider Signale noch um weitere 10dB auf **-40dBm** erhöht, treten die nicht voraussagbaren Effekte auf. Einerseits nimmt der IM-Abstand plötzlich nicht mehr korrekt ab, obwohl er das sollte. Aber dafür erwischt es die nächste IM-Linien im Spektrum, die ungleiche Amplituden aufweisen.



Bei einem Eingangspegel von **-30dBm** pro Signal wird es noch schlimmer:
Der IM-Abstand ändert sich auf einer Seite nicht mehr, auf der anderen ist er plötzlich viel besser.

Aber jetzt weiß man doch ziemlich genau, welche Antennenspannungen noch einigermaßen verzerrungsarm verarbeitet werden können.

6.5. Rauschzahl des vollständigen Konverters

Da muss LTspice leider passen, denn wegen des jetzt in der Schaltung enthaltenen **Mischers** müssen wir im **Zeitbereich** bleiben. Das Programm kann aber nur mit Spektralen Rauschdichten im **Frequenzbereich** umgehen und deshalb kommt der zugehörige AC-Sweep nicht in Frage. Bevor man nun ewig nach einer Lösung grübelt, geht es so schneller:

Man geht nach alter Väter Sitte vor und benützt einfach die Formel für die „**Cascaded Noise Figure**“ (= Gesamtrauschzahl von in Reihe geschalteten Bausteinen) aus dem Lehrbuch:

$$F_{total} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 \bullet G_2} + ...$$

Da wartet allerdings auch Arbeit, denn wir brauchen dazu den Rauschfaktor „F“ sowie die Leistungsverstärkung „G“ jeder Stufe -- **leider nicht in dB, sondern immer als Zahlen-Verhältnis.**

Fangen wir mit der Vorstufe an.

Die simulierte Rauschzahl betrug 0,64dB. Damit wird $F_1 = 10^{\frac{0,64}{10}} = 1,16$

Und mit einer Stufenverstärkung von 25dB erhält man $G_1 = 10^{\frac{25}{10}} = 316$

Dann geht es an den Mischer.

Als idealer Baustein würde er die Umsetzdämpfung ca. 4,5dB aufweisen.

Leistungsverstärkung: $G_2 = 10^{\frac{-4,5}{10}} = 0,355$

Rauschfaktor (= Kehrwert der Verstärkung): $F_2 = \frac{1}{G_2} = \frac{1}{0,355} = 2,82$

Und endlich noch unser Bandpass mit einer Durchgangsdämpfung von 8dB:

Leistungsverstärkung: $G_3 = 10^{\frac{-8dB}{10dB}} = 0,158$

Rauschfaktor (= Kehrwert der Leistungsverstärkung) $F_3 = \frac{1}{G_3} = \frac{1}{0,158} = 6,3$

Dann wird es spannend:

$$F_{total} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3}{G_1 \bullet G_2} = 1,16 + \frac{2,82 - 1}{316} + \frac{6,3 - 1}{316 \bullet 0,355} = 1,16 + \frac{1,82}{316} + \frac{5,3}{112,18}$$

$$F_{total} = 1,16 + 0,0058 + 0,047 = 1,21$$

Damit erhält man die Rauschzahl „NF“ (= noise figure) zu: $NF = 10 \bullet \log 1,21 = 0,83dB$,

was einer Verschlechterung des Vorstufenrauschens um 0,2dB entspricht.
Mit Sicherheit wird dieser Wert in der Praxis durch die realistische Mischerschaltung mit ihrer etwas höheren Dämpfung etwas zunehmen und die MOSFETS streuen ja auch ungeniert. Und an den hohen Verstärkungswert der Vorstufe werden wir wohl auch nicht bei allen gebauten Exemplaren der Platine kommen.
Aber man sieht wieder einmal die alte Regel bestätigt, dass das Rauschen der Vorstufe alles dominiert. Und was dahinter folgt, wirkt sich umso weniger aus, je höher Verstärkung dieser Vorstufe ist. Selbst wenn es ein passiver Mischer und ein Filter mit deutlicher Dämpfung sind.

Literaturverzeichnis

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

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

[3]: Agilent – Design –Tip aus dem Internet: “Measuring IP3” von Bob Myers

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