SPICE-Schaltungs-Simulation mit LTspice IV

Tutorial mit 18 Projekten (und 24 Kapiteln) zum erfolgreichen Umgang mit der frei im Internet (www.linear.com) erhältlichen SPICE-Vollversion der Firma Linear Technologies

Tutorial-Version 2.3

/

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

Email: mail@gunthard-kraus.de

Homepage: www.gunthard-kraus.de

05. Januar 2015

Inhaltsverzeichnis **Seite**

1. Einführung 6

2. Wie funktioniert „SPICE“? 7

3. Installation auf dem PC 7

4. Übungen am mitgelieferten Demo-Beispiel „Astable Multiivibrator“ 7
4.1. Aufruf von Schaltplan und Simulation 7
4.2. Darstellung der Signalverläufe 9
4.3. Löschen von Signalverläufen im Ergebnis-Bildschirm 12
4.4. Andere Farbe für eine Ergebniskurve 12
4.5. Änderung der Simulationszeit 13
4.6. Änderung des dargestellten Spannungs- oder Strombereichs 16
4.7. Cursor-Einsatz 17
4.7.1. Verwendung eines Cursors 17
4.7.1. Verwendung eines zweiten Cursors 17
4.8. Differenzmessungen 18
4.9. Strom-Messungen 19
4.10. Änderung von Bauteilwerten 20

5. Erstes eigenes Projekt: RC – Tiefpass 21
4.1. Zeichnen des Stromlaufplans mit dem Editor 21
4.2. Zuweisung neuer Bauteilwerte 22
4.3. Untersuchung von einmaligen Vorgängen 23
4.3.1. Die Sprungantwort 23
4.3.2. Ein- und Ausschaltvorgang 25
4.4. Periodische Signale am Eingang 26
4.4.1. Sinussigna/l mit f = 1591 Hz 26
4.4.2. Rechtecksignal mit f = 1591 Hz 27
4.4.3. Dreiecksignal mit f = 1591 Hz 28
4.5. AC-Sweep zur Ermittlung des Frequenzganges 29
4.6. Wirkleistung, Blindleistung, Scheinleistung 31

6. FFT (= Fast Fourier Transformation) mit LTSpice 32
6.1. Zusammenfassung der Grundlagen 32
6.2. Dazu ein praktisches Beispiel 34

7. Einbindung fremder Bauteil-Modelle (=Third-party Models) 40
7.1. Verwendung anderer Dioden 40
7.1.1. Was bietet LTSpice selbst als Vorrat? 40
7.1.2. Beschaffung eines Diodenmodells aus dem Internet 41
7.1.3. Erste Simulation mit dem neuen Modell: direkter Aufruf über seine Library 42
7.1.4. Zweite Simulation mit dem neuen Modell: direkte Einbindung der Modellbeschreibung in das Projekt 42
7.1.5. Dritte Simulation mit dem neuen Modell: Übernahme in die Standard-Dioden-Bibliothek von LTSpice 43
7.2. Übernahme neuer Bipolar-Transistoren 45
7.2.1. Ersatz des Standard-Typs durch ein Exemplar aus dem Vorrat 45
7.2.2. Beschaffung eines Transistormodells aus dem Internet 46
7.3. Richtiger Umgang mit Sub-Circuits (*.sub) 47
7.3.1. Beispiel: Verwendung des Operationsverstärkers TL072 47
7.3.1.1. Beschaffung des Spice Modells aus dem Internet 47
7.3.1.2. Simulation einer einfachen Verstärkerschaltung mit Modellaufruf über „include“ 48
7.3.1.3. Simulation mit Modellbeschreibung im Editor-Schaltplan 49
7.3.2. Die Krönung: Erstellung eines kompletten neuen Symbols mit *.subckt als Modellbeschreibung

8. Zweites Projekt: Gleichrichtung **55**
8.1. Einpuls-Gleichrichter ohne Trafo **55**
8.2. Einpuls-Gleichrichter mit Transformator **56**
8.2.1. Ein idealer Trafo muss her! 56
8.2.2. Einpuls-Gleichrichter mit idealem Trafo 57
8.2.3. Erstellung des SPICE-Modells für einen realistischen Transformator mit zwei Wicklungen 58
8.2.4. Erzeugung eines passenden Symbols für diesen Transformator 59
8.2.5. Einpuls-Gleichrichter mit realistischem Trafo 61
8.3. Verwendung einer anderen Diode in der Gleichrichterschaltung **63**
8.3.1. Einsatz einer Diode aus dem mitgelieferten Vorrat von LTspice 63
8.4. Zweipuls-Gleichrichter mit idealem Trafo **64**
8.5. Zweipuls-Gleichrichter mit realistischem Trafo **55**

9. Drittes Projekt: Drehstrom **67**
9.1. Programmierung eines Drehstromsystems **67**
9.2. Prinzip der Drehstrom-Lichtmaschine **69**
9.2.1. Die M3-Schaltung 69
9.2.2. Die B6-Schaltung 70

10. Viertes Projekt: Darstellung von Bauteil-Kennlinien **71**
10.1. Ohm'scher Widerstand **71**
10.2. Diode **72**
10.3. NPN-Transistor **73**
10.4. N-Kanal-Sperrschicht-FET **75**

11. Fünftes Projekt: Schaltungen mit Transistoren **76**
11.1. Einstufiger Verstärker **76**
11.1.1. Ansteuerung mit einem Sinus-Signal **76**
11.1.2. Simulation des Frequenzgangs („AC-Sweep") **78**
11.2. Zweistufiger gegengekoppelter Breitbandverstärker **79**
11.2.1. Pflichtenheft **79**
11.2.2. Simulations-Schaltung und Simulations-Vorgaben **80**
11.2.3. Simulation in der Time Domain (= im Zeitbereich) **80**
11.2.4. DC-Bias (= Gleichstrom-Analyse) **81**
11.2.5. AC-Sweep (= Frequenzgang von 1 Hz bis 200 MHz) **83**
11.3. Der Parameter-Sweep **84**

12. Sechstes Projekt: OPV-Schaltungen **86**
12.1. Wichtige Vorbemerkungen **86**
12.2. Simulationen mit dem idealen OPV-Modell **86**
12.2.1. Umkehrender, idealer Verstärker mit 10facher Verstärkung 86
12.2.2. Nicht umkehrender, idealer Verstärker 87
12.2.3. Idealer Analog-Addierer 87
12.2.4. Aktive Filterschaltung 89
12.3. Das erweiterte OPV-Modell „UniversalOpamp2" **94**
12.4. Probleme beim Betrieb mit nur einer Betriebsspannung **96**
12.5. Verwendung eines OPV-Modells aus dem Internet **97**
12.5.1. Ein invertierender Verstärker mit dem TL072 101
12.5.2. Ein nicht umkehrender Verstärker mit dem TL072 102

13. Siebtes Projekt: DC-DC-Konverter 103
13.1. Bereitstellung des Power-MOSFETs **103**
13.2. Der Step-Up-Konverter (= Aufwärtswandler) **104**
13.3. Der Flyback-Konverter (= Sperrwandler) **106**
13.4. Der Step-Down-Konverter (= Abwärtswandler) **108**

14. Achtes Projekt: Phasenanschnitt-Steuerung mit Thyristor 110
14.1. Das eingesetzte Thyristor-Modell 110
14.2. Schalten von Ohm'schen Lasten 111
14.3. Schalten von induktiven Lasten 113
14.4. Zündung des Thyristors über einen Gate-Transformator 114

15. Neuntes Projekt: Digitale Schaltungssimulation 115
15.1. Was man vorher wissen sollte 115
15.2. Einfacher Anfang: die Umkehrstufe (= NOT oder Inverter) 116
15.3. Der AND-Baustein 117
15.4. Das D-Flipflop 118
15.5. Dreistufiger Frequenzteiler mit D-Flipflops 119

16. Zehntes Projekt: Signalformen und Oberwellen 120
16.1. Grundlagen 120
16.2. Simulation des Spektrums eines einmaligen Pulses 123
16.3. Simulation des Frequenzspektrums bei einem periodischen Pulssignal 125
16.4. Der ideale Sinus 126
16.5. Der einseitig begrenzte Sinus 127
16.6. Der symmetrisch begrenzte Sinus 128

17. Elftes Projekt: Untersuchung eines Sinus-Oszillators 129

18. Zwölftes Projekt: Echos auf Leitungen 133
18.1. Leitungen -- nur zwei Drähte? 133
18.2. Echos auf Leitungen 135
18.3. Simulation des vorigen Rechenbeispiels mit LTSpice 137
18.4. Leerlauf oder Kurzschluss als Last am Kabelende 140
18.5. Verwendung von Kabel mit Verlusten (Beispiel: RG58 / 50Ω) 142
18.5.1. Wie simuliere ich RG58-Kabel? 142
18.5.2. Simulation der Kabeldämpfung bei 100MHz 143
18.5.3. Speisung der RG58-Leitung mit einer Pulsspannung 146
18.5.4. Ein Kurzschluss am Ende der RG58-Leitung 147

19. Dreizehntes Projekt: S-Parameter 148
19.1. Jetzt nochmals Echos, aber mit System 148
19.2. Praxisbeispiel: 110MHz – Tschebyschef – Tiefpassfilter (LPF) 151

20. Vierzehntes Projekt: Die Sache mit der Impulsantwort 156
20.1. Erstes Beispiel: Dirac-Impuls speist einen 160Hz-RC-Tiefpass 156
20.2. Zweites Beispiel: Dirac-Untersuchung des bekannten 110MHz-Tiefpasses 162
20.2.1. Bestimmung von S21 (= Forward Transmission) 162
20.2.2. Bestimmung von S11 (= Input Reflection) oder S22 (= Output Reflection) 167

21. Fünfzehntes Projekt: Rauschen -- Grundlagen, Anwendung, Simulation mit LTSpice 169
21.1. Erste Bekanntschaft 169
21.2. „Rauschen" -- woher kommt das? 169
21.3. Weitere Rauschquellen 173
21.4. Weißes Rauschen zur Messung einer Übertragungs-Funktion 174
21.5. Erzeugung eines Rauschsignals im Zeitbereich mit LTspice 180
21.6. Die „Noise Figure NF in dB" bei Verstärkern 182
21.7. Die Bestimmung der „Noise Figure NF in dB" in der Praxis 184

22. Sechzehntes Projekt: Double Balanced Mixer, Teil 1 = Ringmodulator 185
22.1. Etwas Grundlagen 185
22.2. Standardschaltung des Ringmodulators 186
22.3. Ein Problem: die erforderlichen Übertrager 187
22.4. Simulation des DBM-Verhaltens mit idealen Übertragern 188

23. Siebzehntes Projekt: Double Balanced Mixer, Teil 2 = Active Mixer = Gilbert Cell 190
23.1. Einführung 190
23.2. Prinzip 191
23.3. Untersuchung des RF-Verstärkers 193
23.3.1. DC-Analyse 193
23.3.2. Die Gilbert Sell als Verstärker-Schaltung 195
23.3.2.1. Der Emittergekoppelte Verstärker / Unsymmetrischer Betrieb 195
23.3.2.2. Der Differenzverstärker 200
23.3.2.3. Symmetrischer Betrieb 200
23.4. Die Gilbert-Cell, als Double Balanced Mixer (DBM) eingesetzt 203

24. Achtzehntes Projekt: Modulation 207
24.1. Prinzip der Amplitudenmodulation (AM) 207
24.2. Amplitude Shift Keying (ASK) 209
24.3. Die Frequenzmodulation (FM) 209
24.3.1. Erzeugung eines FM-Signals mit der Spannungsquelle „voltage“ 209
24.3.2. FSK = Frequency Shift Keying 212
24.4. Quadratur-Amplitudenmodulation (QAM) 214
24.4.1. Grundlegende Informationen 214
24.4.2. Erzeugung eines QAM-Signals 214
24.4.3. Filterung des QAM-Signals zur Anpassung an die Übertragungs-Kanalbandbreite 221
24.4.4. Demodulation des QAM-Signals 227
24.4.4.1. Prinzip der Synchron-Demodulation 227
24.4.4.2. Die Simulation der vollständigen QAM-Demodulatorschaltung 228

1. Einführung

Die Schaltungssimulation ist aus der modernen Elektronik nicht mehr wegzudenken, denn nur mit ihr sind kürzeste und damit intensiv Kosten sparende Entwicklungszeiten beim Entwurf von Schaltungen oder Änderungen möglich. Außerdem kann jede neue Idee „ohne Lötkolben“ bis ins letzte Detail hinein ausgetestet werden, bevor überhaupt eine Leiterplatte gefertigt wird. Deshalb eröffnen sich nicht nur für die professionellen Entwickler, sondern auch für Hobbyelektroniker, Studenten, Schüler und Auszubildende völlig neue kreative Möglichkeiten.

Das sollte man wissen: der Industriestandard für Schaltungssimulationsprogramme lautet

SPICE.

Er wurde noch vor 1980 an der Berkeley-Universität entwickelt und um die eigentliche “SPICE-Machine” herum (…so heißt das spezielle Simulationsprogramm) haben nun viele, viele Leute Bedienungsoberflächen erdacht und auf dem Markt und daraus wurde dann „PSPICE = SPICE for the PC“.
Auf dem Markt tummeln sich nun dafür entsprechend viele Anbieter. Dabei werden die Profi-Industrieprogramme meist mit einem Leiterplatten-CAD-Programm kombiniert -- da kann man hinterher gleich seine komplette Schaltung auf der entworfenen Platine austesten, wobei die Mischung von Analog- und Digitalschaltkreisen oder die Ermittlung der EMV-Störstrahlung oder das gegenseitige Übersprechen der Leiterbahnen komplett simuliert wird. Diese Programme kosten aber kräftig Geld und deshalb gibt es auch einfachere Sachen. Deshalb ist es sehr erfreulich, dass man bei genauer Suche auf Programme stößt, die

a) leistungsfähig, gut bedienbar und dennoch kostenlos sind und

b) trotzdem keine empfindlichen Beschränkungen bei der eigentlichen Schaltungssimulation aufweisen.

Der absolute und kostenlose Einsteigerhit heißt

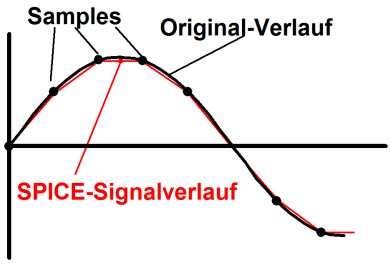
LTspice IV (früher: LT Switcher CAD)

von Linear Technologies. Diese Software will **nur Schaltungen simulieren** und weist **keinerlei Einschränkungen** auf, da sie als freundliches Angebot der Firma an alle Entwicklungsingenieure gedacht ist (…natürlich zuerst zur Simulation von Schaltreglern und Schaltnetzteilen unter Verwendung von Halbleitern dieser Firma). Sie wird in der Zwischenzeit auch für sehr viele andere Zwecke eingesetzt und hat sich einen großen Freundeskreis erworben. Es gibt deshalb auch eine „LTSpice Community“ auf Yahoo, bei der man als Anwender mitmachen sollte und die kostenlos, aber ungemein hilfreich bei Problemen ist.

Etwas drollig ist die Bedienung -- sie besteht nämlich aus einer **kräftigen Mischung aus Mausklicks UND Tastenkommandos UND Kommandozeilen-Eingaben**… und manche Dinge muss man erst mal suchen und finden! Aber wenn man den Dreh heraushat, läuft es prima.

Zum Schluss noch eine ganz wichtige Vorbemerkung:
Diese unterschiedlichen Eingabemöglichkeiten bedeuten, dass viele Funktionen des Programms (Beispiel: „Simuliere jetzt die Schaltung“) auch auf völlig verschiedene Weise zugänglich sind oder gestartet werden können!
Im Tutorial werden jedoch NICHT ALLE dieser unterschiedlichen Methoden einzeln besprochen. Hier muss der Anwender selbst prüfen, wozu etwas da ist bzw. welche Methode er in Zukunft verwenden will.

2. Wie funktioniert „SPICE“?



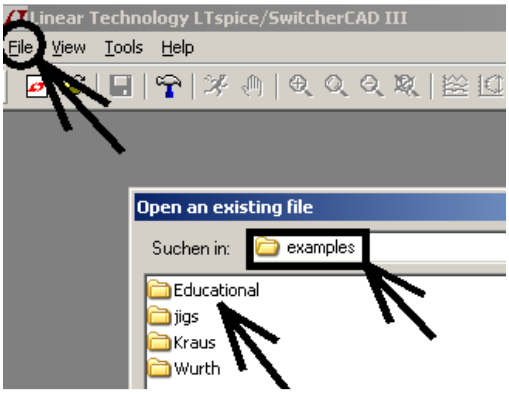
Ganz einfach: man ersetzt einen beliebigen Kurvenverlauf beim Eingangssignal durch eine „Punktfolge“ (= Samples), die durch gerade, also lineare Stücke miteinander verbunden werden. Verwendet man genügend Samples, dann erkennt man die „Ecken“ nicht mehr und die Kurve nähert sich immer mehr dem Originalverlauf. Entscheidend ist dabei, dass die Änderungen zwischen zwei aufeinander folgenden Punkten im Strom- oder Spannungsverlauf nun durch **lineare Gleichungen** (und nicht mehr wie bei L oder C durch Differentialgleichungen) beschrieben und ermittelt werden können. Das spart unglaublich Rechenaufwand und Rechenzeit. **Allerdings muss nun der Anwender die Zahl der Samples über ihren zeitlichen Abstand (= „Time step“) vorgeben und einen Kompromiss zwischen Genauigkeit und Rechenzeit schließen**.

3. Installation auf dem PC

Nichts leichter als das: nach dem Download aus dem Internet (www.linear.com) klickt man die „ ***.exe-Datei**“ an und wird in gewohnter Weise durch die Windows-Installation geführt. Der zugehörige Icon wird hierbei auf dem Windows-Startbildschirm automatisch angelegt. Fertig!

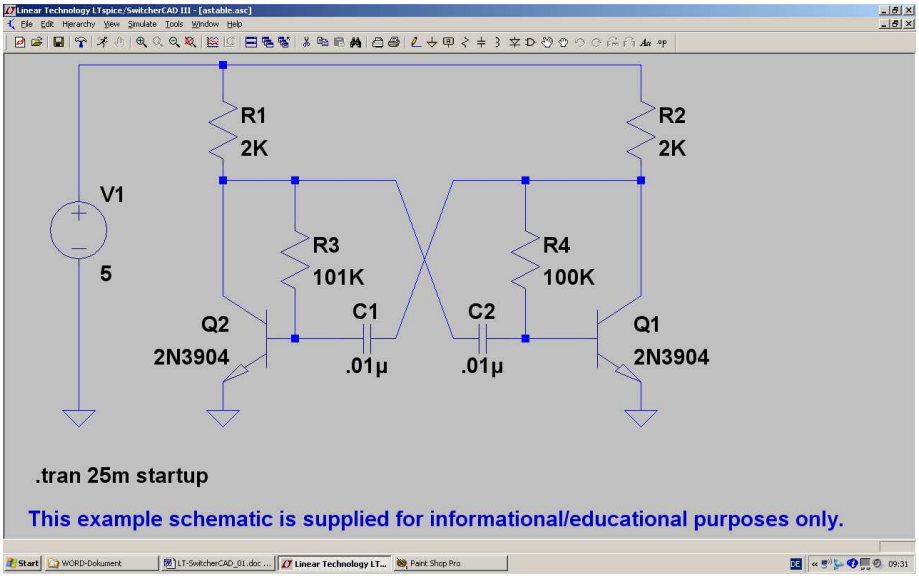
4. Übungen am mitgelieferten Demo-Beispiel „Astable Multivibrator“

4.1. Aufruf von Schaltplan und Simulation



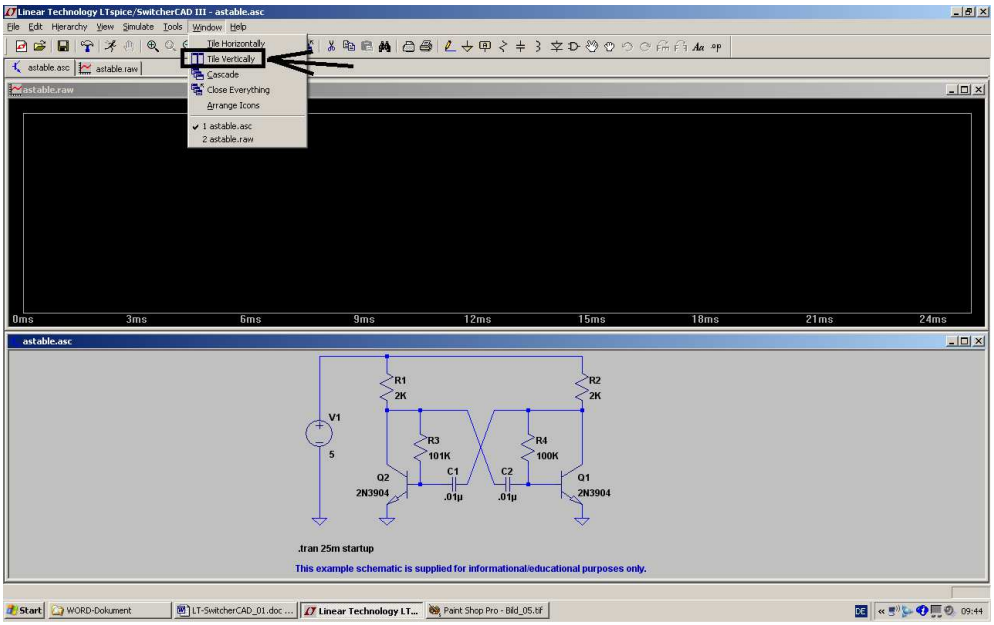
Nach dem Programmstart klicken wir im linken oberen Eck erst auf „**File**“ und dann „**Open**“.

Im Verzeichnis unseres Programms gibt es dann mehrere Ordner. Wenn wir darin „**Examples**“ und anschließend „**Educational**“ öffnen, können wir jetzt einen **astabilen Multivibrator** auswählen (= „astable.asc“).



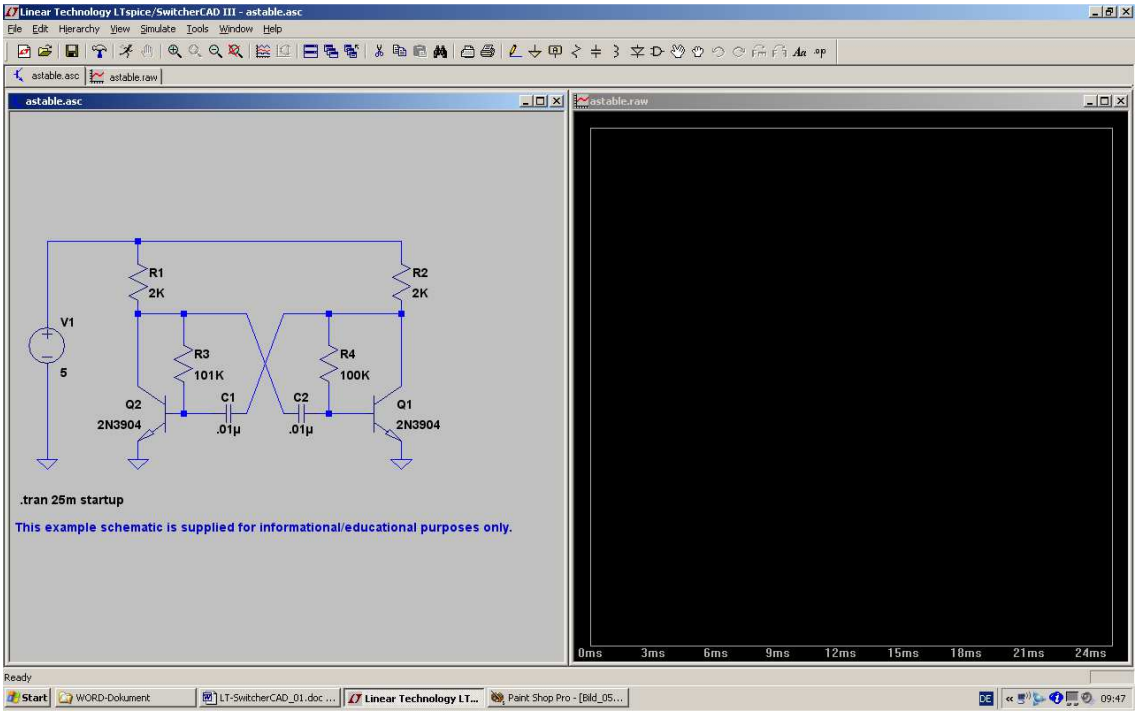
Haben wir alles richtig gemacht, dann empfängt uns der nebenstehende Bildschirm mit der fertigen Schaltung.

In der oberen Menüleiste finden wir (neben einem Hammer!) einen Button mit einem rennenden Männchen drauf. Das ist die **Starttaste für die Simulation** und sie wird nun gedrückt.



Das obere Diagramm ist für die Ergebnis-Darstellung gedacht, aber noch leer.
Dass die Schaltung zusätzlich in der unteren Bildhälfte präsentiert wird, hat seinen Grund…

Übrigens:
Wem diese Darstellung nicht behagt, weil er lieber links die Schaltung und rechts die Signalverläufe sehen möchte, der öffne das Menü „**Window**“ und rufe „**Tile vertically**“ auf. Dann sieht das Alles so aus:

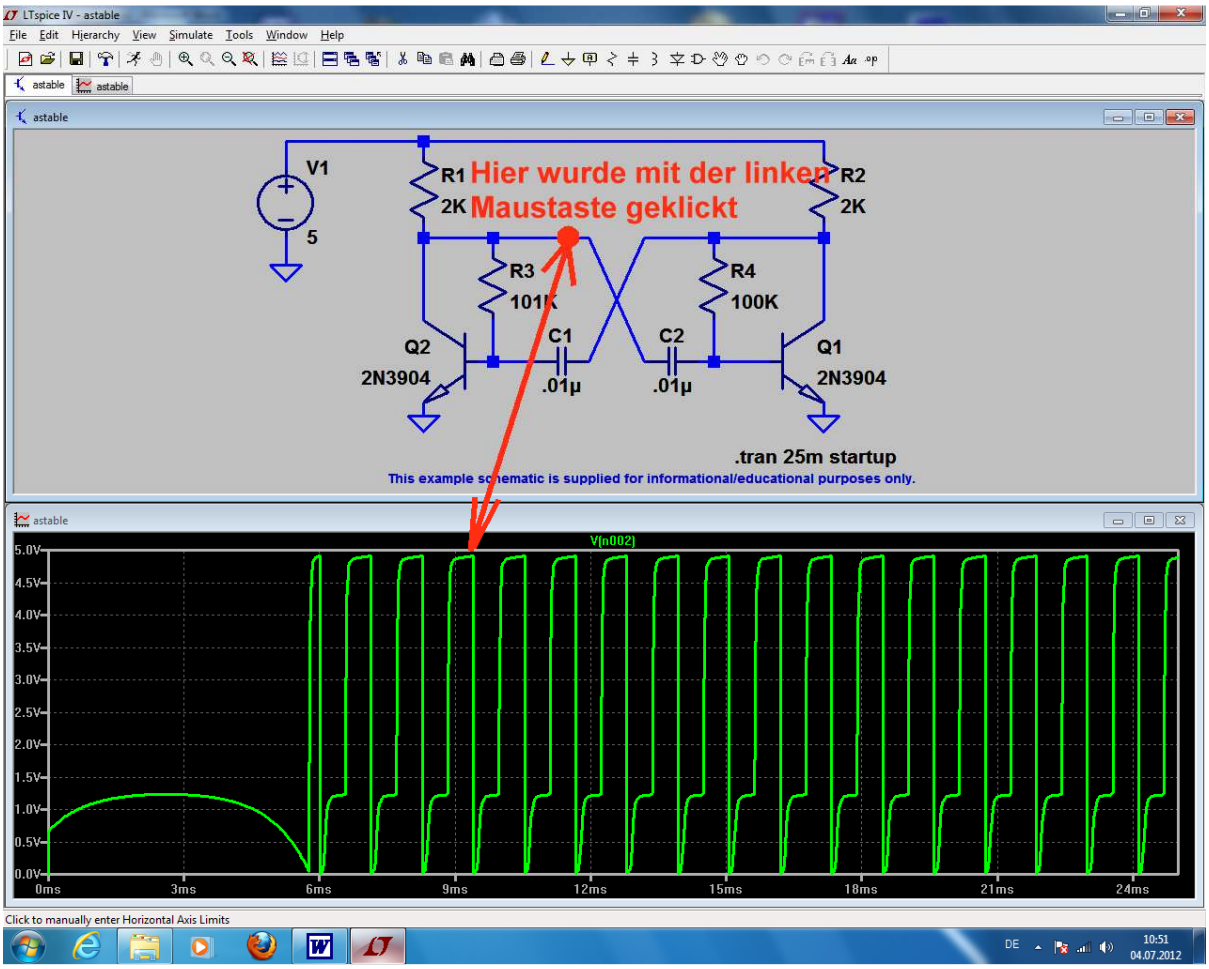


Und wer nach der Simulation die Ergebniskurven gern etwas dicker hätte, der drücke jetzt schon auf den „Button mit dem Hammer“. Da gibt es eine Karteikarte „Waveforms“ und darauf „Plot data with thick lines“.

Ebenso findet man hinter dem Hammer eine Karteikarte für „Drafting options“. Die dritte Möglichkeit von unten herauf heißt: „Draft with thick lines“ und das gibt schöne dicke Leitungen im Schaltbild.

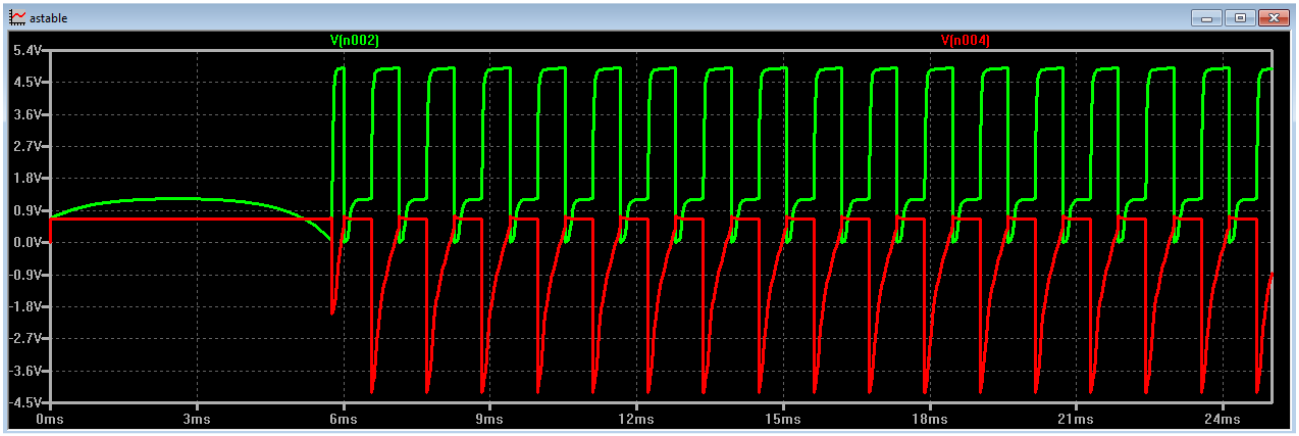
4.2. Darstellung der Signalverläufe

Will man den Verlauf der Spannung (gegen Masse) an einem bestimmten Punkt der Schaltung wissen, dann fahre man mit dem Mauscursor darauf. Sobald man genügend in die Nähe kommt, ändert der Cursor seine Form von einem einfachen Kreuz in eine Prüfspitze. Dann reicht ein simpler Mausklick, um die Darstellung zu aktivieren:



So kann man nacheinander alle gewünschten Spannungen sichtbar machen -- ***in einem einzigen Diagramm. Aber leider ohne jede Spur innerhalb der Schaltung, wo gerade gemessen wurde! Lediglich die Kennzeichnung anhand der Schaltungsknoten und die unterschiedlichen Kurvenfarben helfen da bei der Identifizierung weiter…..***

Hat man beispielsweise alle Kollektor- und Basisspannungen desselben Transistors in der Schaltung gemessen, dann sieht das Ergebnisfenster so aus:



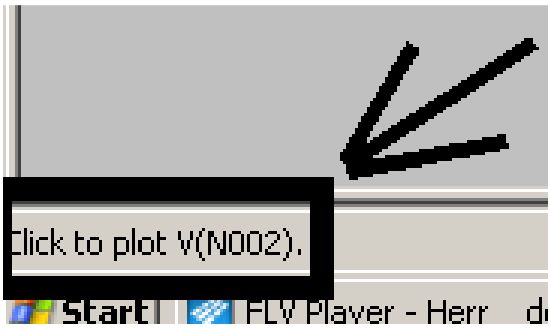
Grundsätzlich kann man sich da **auf mehrere Arten** helfen:

a) Man informiert sich über die **Knotenbezeichnung innerhalb der Schaltung** (denn darnach werden die simulierten Spannungs- oder Stromverläufe durchnummeriert). Das Schlüsselwort dafür heißt „**Netzliste“ (= netlist)** und man findet diese Liste im Menü „**View**“ unter „**SPICE Netlist**“.

SPICE Netlist: C:\Programme\LTC\SwCADIII\examples\Educational\astable.net

```
* C:\Programme\LTC\SwCADIII\examples\Educational\astable.asc
R1 N001 N002 2K
R2 N001 N003 2K
R3 N002 N004 101K
R4 N003 N005 100K
C1 N003 N004 .01µ
C2 N005 N002 .01µ
V1 N001 0 5
Q1 N003 N005 0 0 2N3904
Q2 N002 N004 0 0 2N3904
.model NPN NPN
.model PNP PNP
.lib C:\Programme\LTC\SwCADIII\lib\cmp\standard.bjt
.tran 25m startup
* This example schematic is supplied for informational/educational purposes onl
.backanno
.end
```

So sieht die Netzliste für unsere Schaltung aus und sie ist nicht schwer zu verstehen. Jede Zeile steht für ein Bauteil und beginnt mit dem Bauteilnamen. Dann folgen die Knoten, zwischen denen das Bauteil eingefügt ist und den Abschluss bildet der Bauteilwert.



b)
Noch einfacher geht es, wenn man mit dem Cursor **nochmals an die interessierende Leitung heranfährt.** Sobald sich dann nämlich der **Cursor in die Prüfspitze verwandelt**, wird

ganz links unten im Eck eine Meldung mit der Spannungsbezeichnung samt Knotennummer eingeblendet.....

---

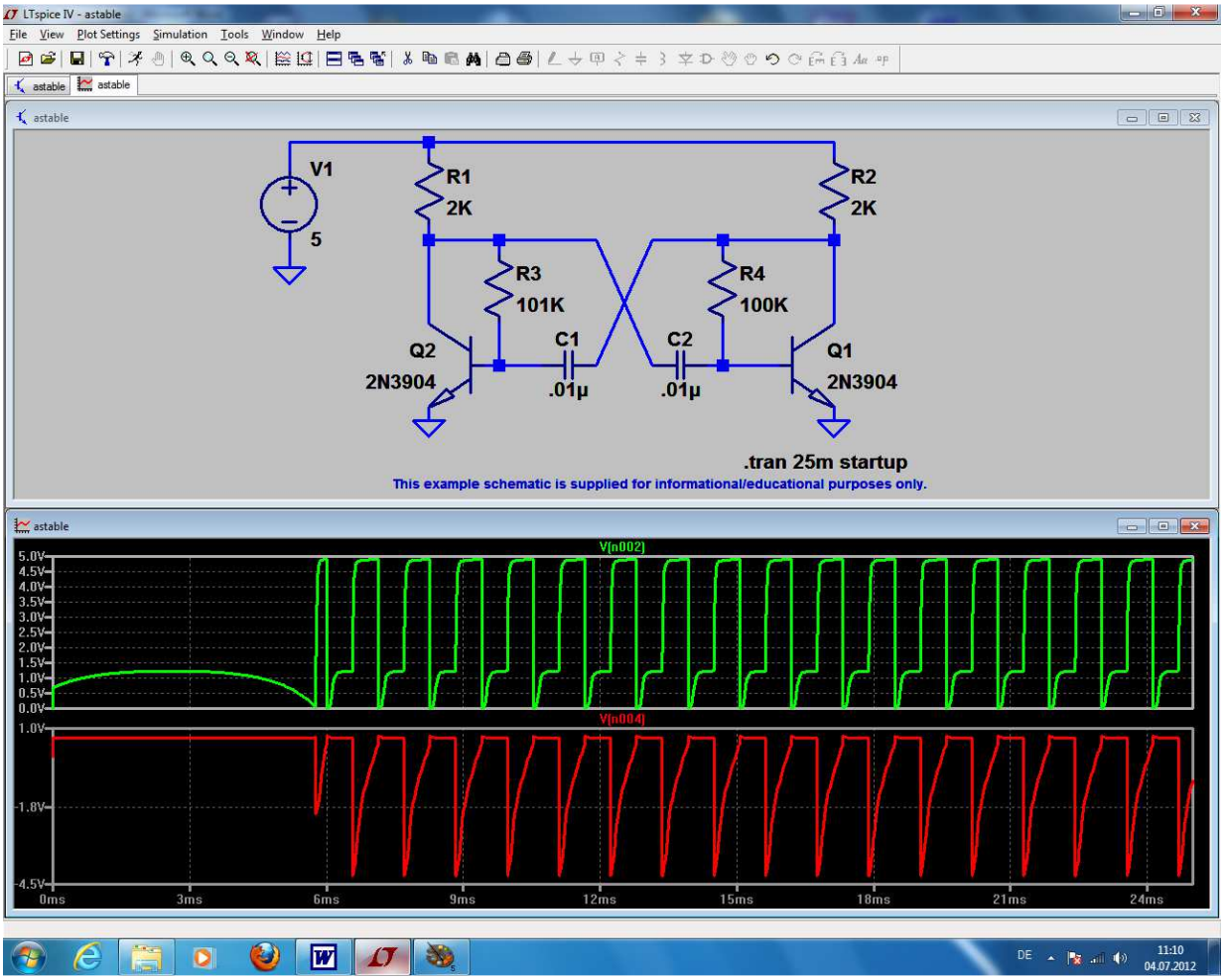
c)

Wenn zu viele Verläufe in einem Diagramm angezeigt werden, verliert man leicht die Übersicht. Da ist es besser, gleich ***mehrere getrennte Fenster zu benützen***.

Dazu klickt man mit der rechten Maustaste auf das Ergebnisfenster und wählt „**Add Plot Pane**“.
Nun haben wir ein weiteres Ergebnisfenster zur Verfügung, in das man kurz hineinklickt (dadurch wird es aktiviert).

Nun reicht ein Klick im Schaltbild auf die gewünschte Leitung (z. B. den Basis-Anschluss des Transistors) zur getrennten Darstellung der Signalverläufe.

Oder man klickt mit der rechten Maustaste auf das Diagramm, wählt „**Add Trace**“ und sucht sich die gewünschte Kurve aus.



Diese Prozedur kann man natürlich wiederholen, aber ab 4 Diagrammen wird es schon wieder etwas unübersichtlicher....

Wie überflüssige Kurven in einem Fenster wieder gelöscht werden können, steht im folgenden Kapitel.

4.3. Löschen von Signalverläufen im Ergebnis-Bildschirm

Wenn z. B. nur ein einziges Fenster für die Ausgabe der Simulationsergebnisse gewählt wurde, sollte man **einzelne oder alle Kurven schnell löschen** können, wenn plötzlich noch ein anderer Punkt der Schaltung interessiert. Das geht so:

1) Wir klicken mit der linken Maustaste irgendwo einmal in das Ergebnisfenster, um auf diesen Bildschirmbereich umzuschalten. **Der Erfolg kann leicht kontrolliert werden, denn der Rahmen des Fensters färbt sich dann dunkelblau.**

2) Dann drücken wir die Taste **F5.** Sofort ändert sich der Cursor und wird zu einer **Schere**!

3) Diese Schere wird nun auf die Bezeichnung des zu löschenden Kurvenverlaufs am oberen Diagrammrand -- z. B. auf „V(n001)“ -- gestellt und anschließend einmal links mit der Maus geklickt. Damit ist die betreffende Kurve gelöscht und aus dem Diagramm verschwunden.

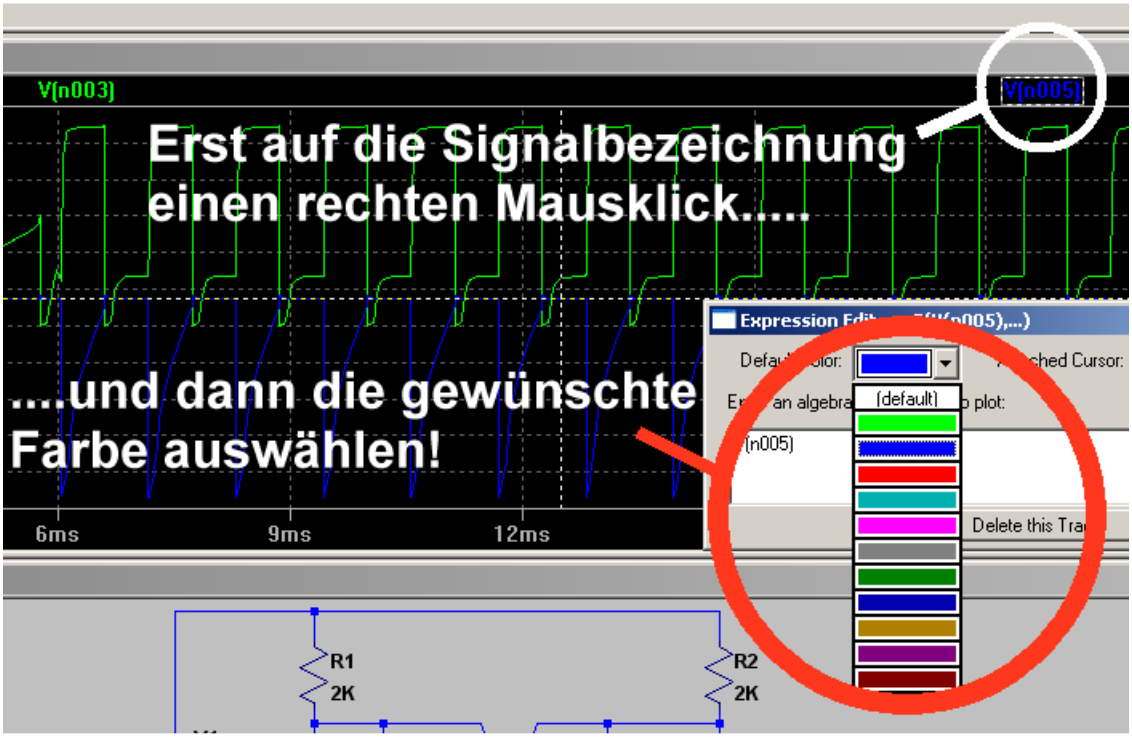
Wichtig:
Um jetzt wieder neu eine Spannung messen zu können, muss man zuerst durch **Druck auf die „Escape“-Taste die Löschfunktion ausschalten** und dann zum **Fenster mit der Schaltung zurückkehren!**

Das geht natürlich wieder mit einem linken Mausklick irgendwo auf die Schaltung. Bitte beobachten, ob sich nun der Rand dieses Fensters dabei wirklich dunkelblau färbt! Wenn ja, dann kann in bekannter Weise erneut mit der Maus an den gewünschten Punkt herangefahren werden, bis der Cursor zur Tastspitze wird. Ein linker Mausklick genügt dann.

Übrigens: um ein komplettes Diagramm (= plot pane) zu löschen, klicke man rechts auf dieses Diagramm. Dann findet man in der Liste das Kommando „Delete this Pane“).

4.4. Andere Farbe für eine Ergebniskurve

Oft ist die Farbe einer Kurve (Beispiel: Blau) vor dem dunklen Hintergrund schlecht sichtbar, aber die Umstellung ist eine Kleinigkeit:



4.5. Änderung der Simulationszeit

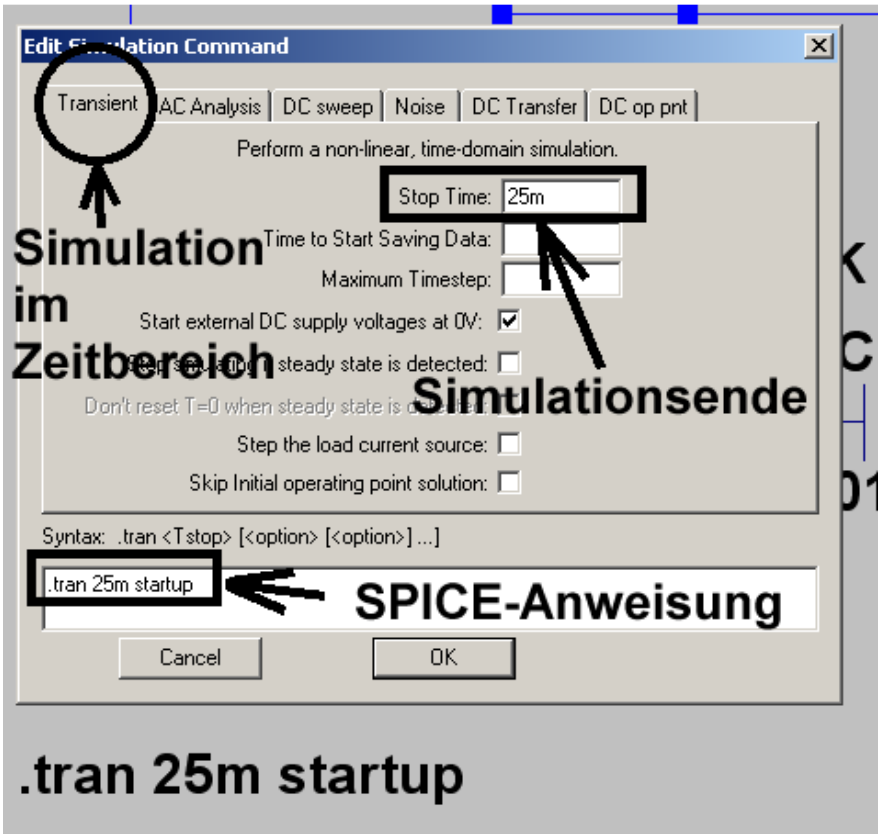
In dieser Anweisungszeile versteckt sich der SPICE-Simulationsauftrag!

Er sagt:
Simuliere die Verläufe aller Signale im Zeitbereich (time domain) von Null bis 25 Millisekunden.

(Der Simulationsstart ist **IMMER bei Null**)

Der Zusatz „**startup**“ hat dabei laut Online-Hilfe folgende Bedeutung:

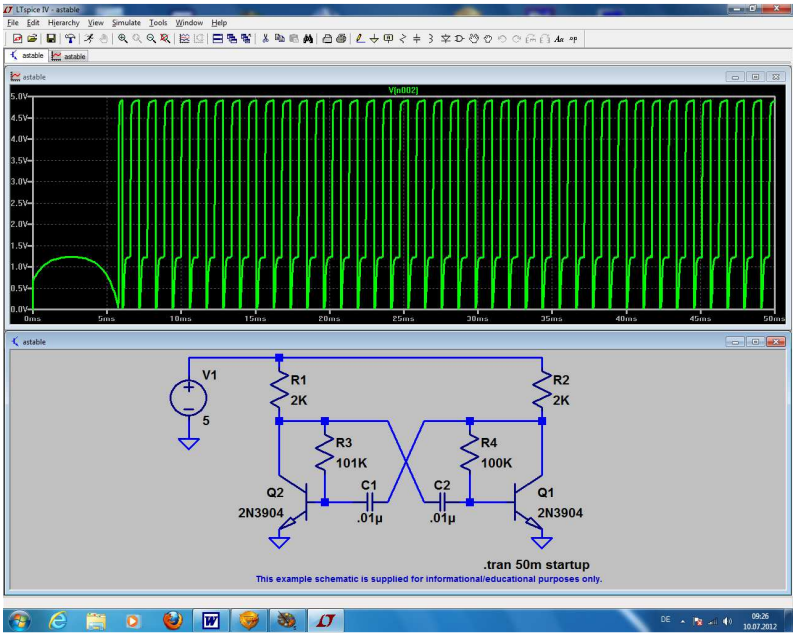
```
startup
This is similar to SPICE's original "uic". It means that
independent sources should be ramped on during the
first 20µs of the simulation. However, a DC operating
point analysis is performed using the constraints specified on
a .ic directive.
```



Stellt man nun den Mauszeiger mitten in den Text der „tran“-Zeile und drückt **die rechte Maustaste**, dann erscheint dieses zugehörige Menü.

Wollen wir also kürzer oder länger simulieren lassen, dann tragen wir im Kästchen „**Simulationsende“** die neue Zeit ein (…sie wird automatisch in die Anweisung übernommen..), drücken auf OK und anschließend auf den **„RUN“-Button mit dem laufenden Männchen.**

Aufgabe:
Bitte mal mit einer Stoppzeit von 50 Millisekunden simulieren und nur die Spannung am linken Kollektor (= V(n002) bei Transistor Q2) darstellen!



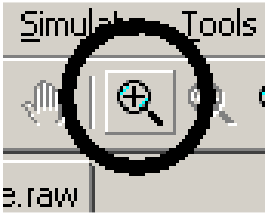
So sollte das dann aussehen! Wie man sieht, wird bei SPICE sehr viel Wert auf die Analyse des Verhaltens der Schaltung nach dem Start gelegt. Das ist kein Wunder, denn die Firma hat es ja zur Analyse von Schaltnetzteilen entwickelt und da ist diese Information außerordentlich wichtig.

Will man jedoch nur die Verläufe im „eingeschwungenen Zustand“ sehen, dann greift man z. B. zur Zoom-Funktion:

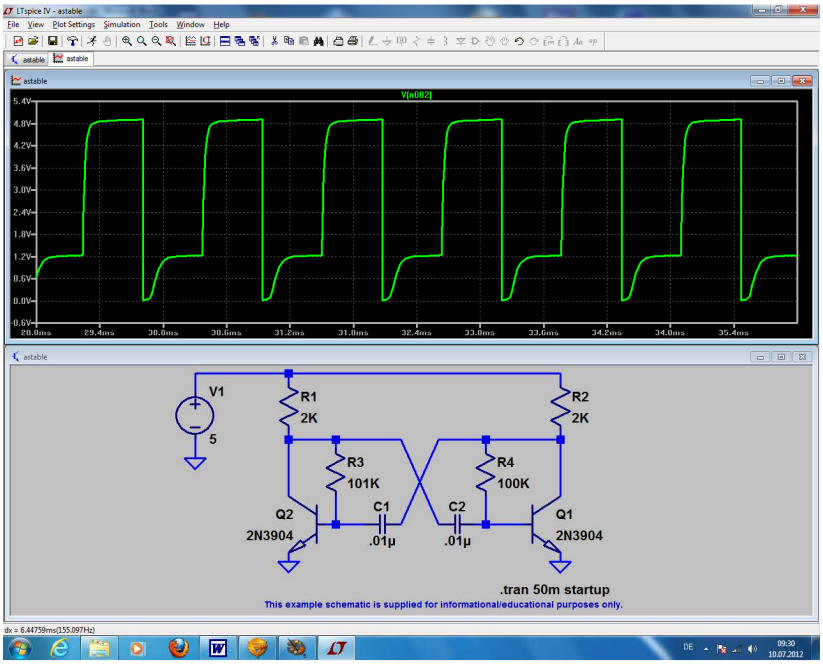
Aufgabe:
Stellen Sie nur den Spannungsverlauf von ca. 30 bis 35Millisekunden dar.

Achtung: da gibt es zwei Möglichkeiten!

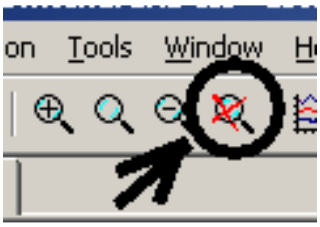
Lösung 1:



a) Wir klicken auf den „**Vergrößerungs-Button** und ziehen anschließend bei gedrückter linker Maustaste ein Rechteck um den Spannungsverlauf von 30…..35 Millisekunden.



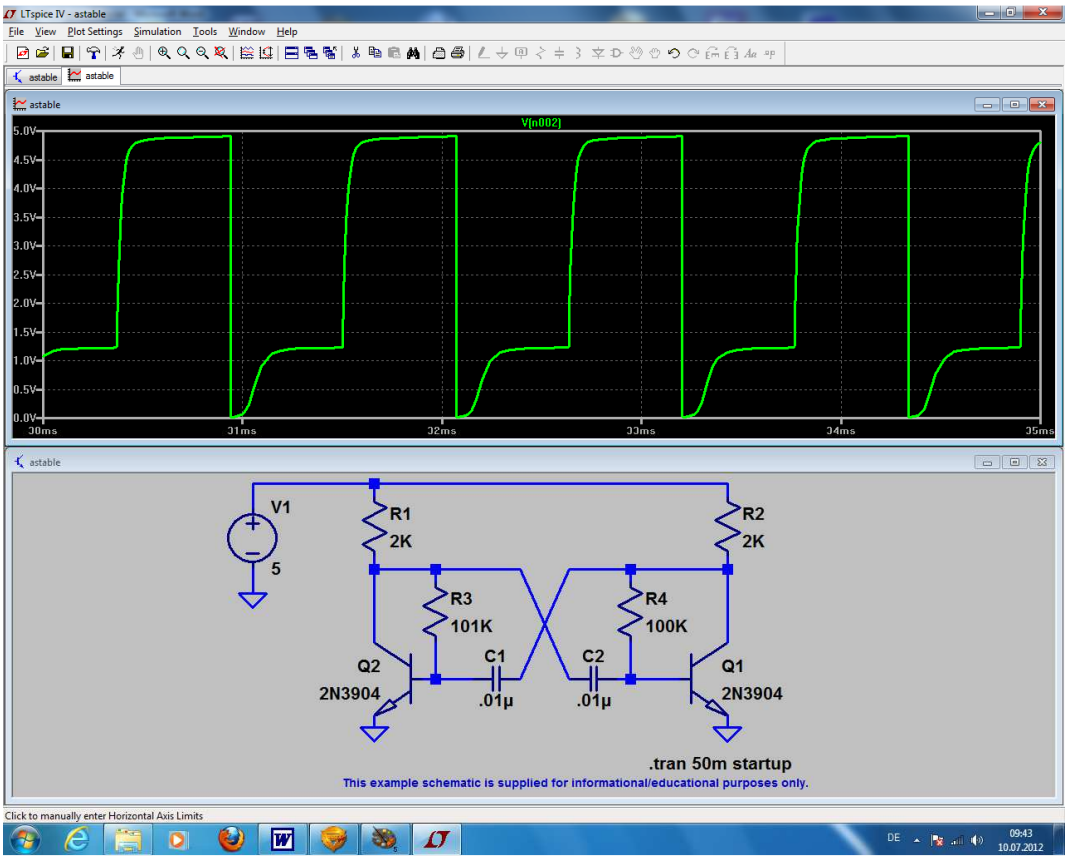
b) Nach dem Loslassen der linken Maustaste muss der gewünschte Ausschnitt erscheinen.



c) Will man wieder zur ursprünglichen **Voll-Darstellung** zurückkehren, dann klickt man einfach auf die „**Gesamt-Taste**“

Lösung 2:
Wenn man mit dem Cursor genau auf die Zahlenteilung der waagrechten Achse fährt, dann verändert sich der Cursor und wird zu einem **LINEAL**. Klickt man mit der **linken Maustaste** darauf, so öffnet sich ein Menü, in dem man die **Startzeit (30ms), die Skalenteilung („tick“ = 1ms) und die Stoppzeit (35ms)** eintragen kann:

Mit „OK“ zeigt sich folgendes Bild des Bereiches von 30 bis 35ms mit einem Tick von 1ms:

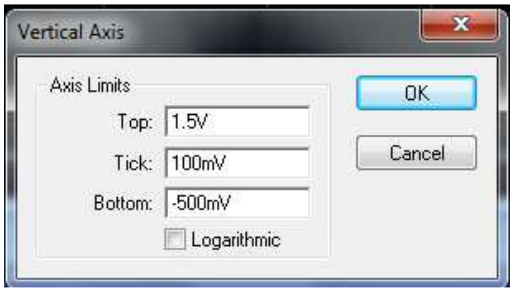


4.6. Änderung des dargestellten Spannungs- oder Strombereichs

Oft interessieren bei einer Schaltungsentwicklung bestimmte Details bei den Signalverläufen. So möchte man sich z. B. die negative Spitze unseres Signals genauer ansehen. Da gibt es zwei Möglichkeiten:

a) Man arbeitet mit der eben besprochenen **Zoom-Funktion** oder

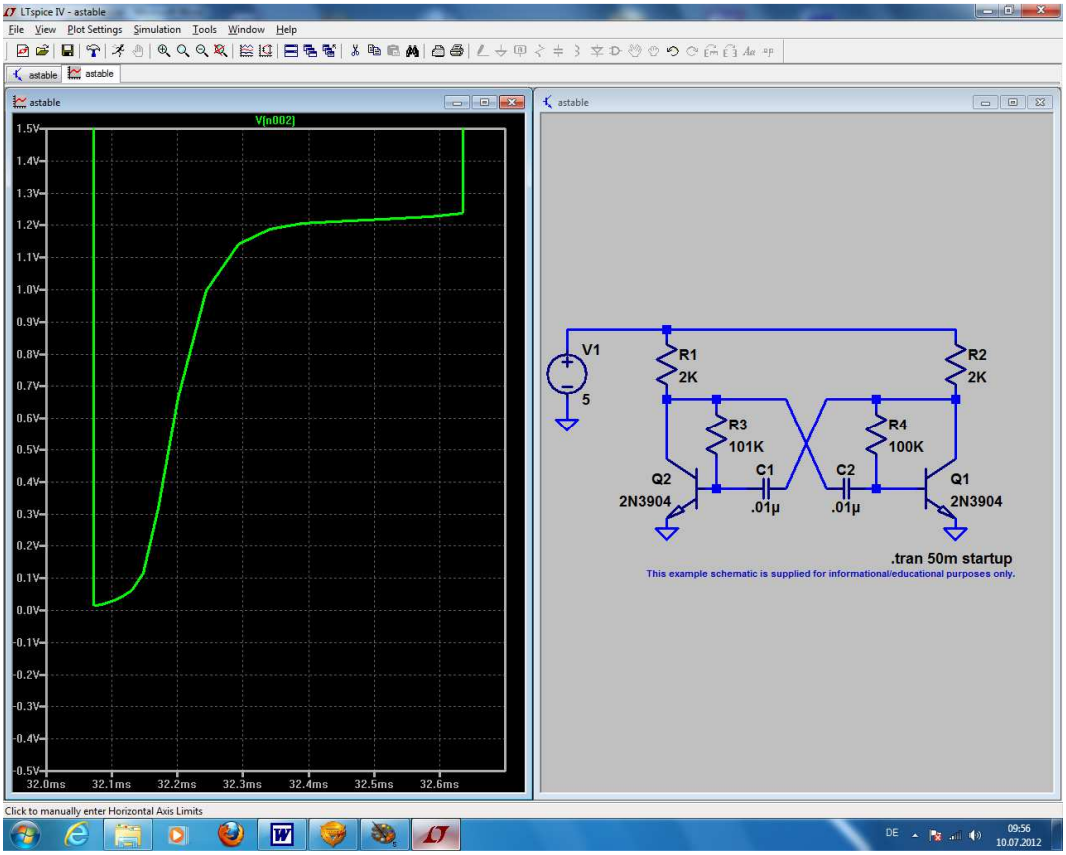
b) Man geht wieder nach der oben stehenden Lösung 2 vor und fährt erst auf die Skalenteilung der waagrechten Achse. Nach Eingabe des neuen Zeitbereichs (hier: von 32ms bis 32,7ms, Tick = 0.1ms) klickt man OK. Dann stellt man den Cursor auf die Skalenteilung der **senkrechten Achse** und wartet auf das Auftauchen des Lineals. Nach einem linken Mausklick taucht nun das Menü der senkrechten Achse auf und wartet auf die Eingabe des Benutzers.



c)

Mit diesen Einträgen wird der Bereich von -0,5V bis +1,5V und einem Tick von 100mV dargestellt.

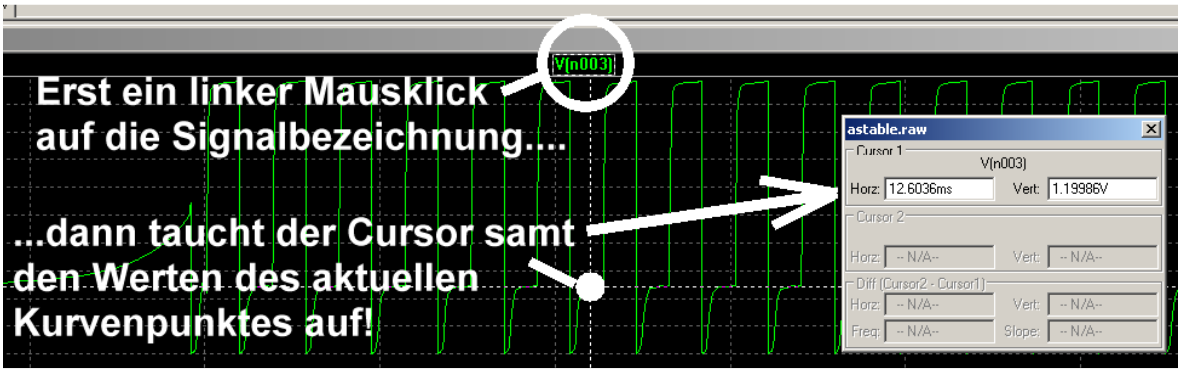
So wird das Ergebnis schließlich aussehen, wenn man zusätzlich unter **„Window“ auf „Tile vertically“** umstellt.



4.7. Cursor-Einsatz

4.7.1. Verwendung eines Cursors

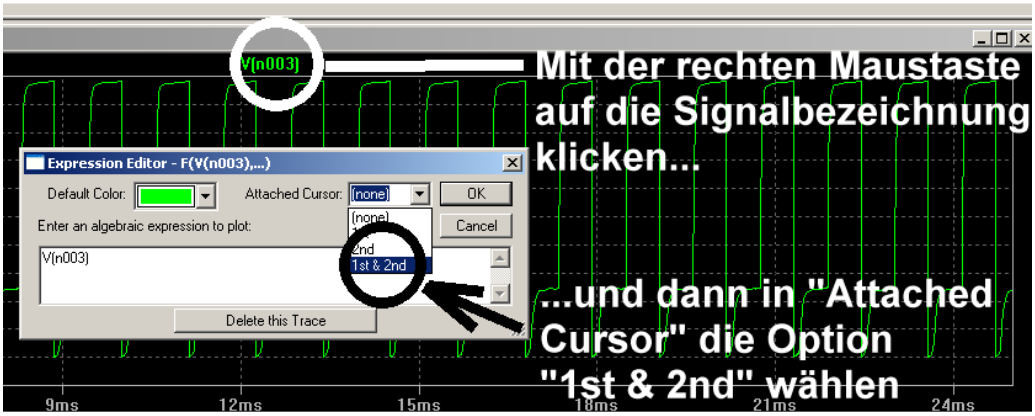
Oft möchte man eine Ergebniskurve „abfahren“, um an bestimmten Zeitpunkten den dort gültigen Spannungs- oder Stromwert zu ermitteln. Das geht so:

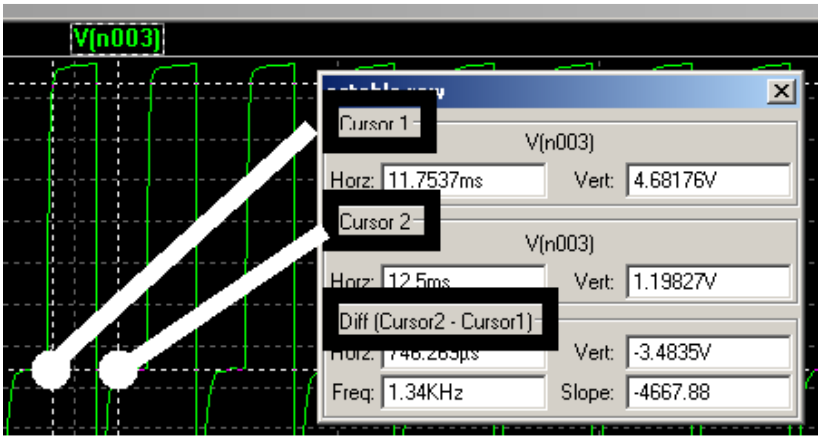


Fährt man nun die Maus genau auf das „Fadenkreuz“, dann wird nicht nur die Cursor-Nummer (hier: „1“) eingeblendet. Zusätzlich lässt sich nun das Fadenkreuz durch „Ziehen bei gedrückter Maustaste = DRAG“ entlang der Kurve verschieben. Damit kann an jeder Stelle der Kurve die Zeit sowie der genaue Amplitudenwert aus dem eingeblendeten Fenster entnommen werden.

4.7.2. Verwendung eines zweiten Cursors

Dazu klickt man mit der **RECHTEN Maustaste** auf die Signalbezeichnung und geht im auftauchenden Fenster in das Menu „**Attached Cursor“**

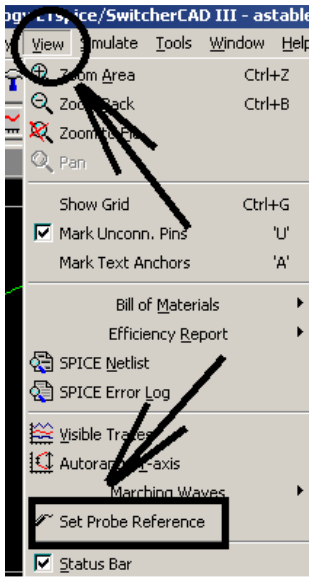




Die beiden Cursors lassen sich nun getrennt verschieben. Ihre aktuellen Werte sowie die Differenz werden wieder in einem Fenster eingeblendet.

4.8. Differenzmessungen

Grundsätzlich werden alle Spannungen gegen den Referenzpunkt „Masse“ gemessen und dargestellt. Da kämpft man dann mit einem altbekannten Problem: wo steckt das „Differential-Voltmeter“, mit dem die Spannung direkt zwischen zwei Bauteil-Anschlüssen ermittelt werden kann?

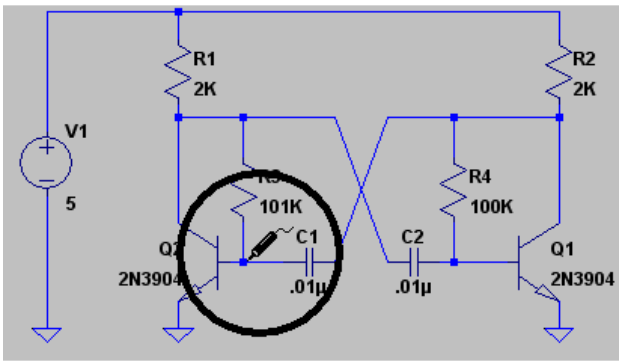


Nun, bei LTSpice geht das etwas anders: man muss zuerst bewusst einen neuen Referenzpunkt in die Schaltung setzen und dann am anderen Bauteilanschluss klicken!

Das geht so:

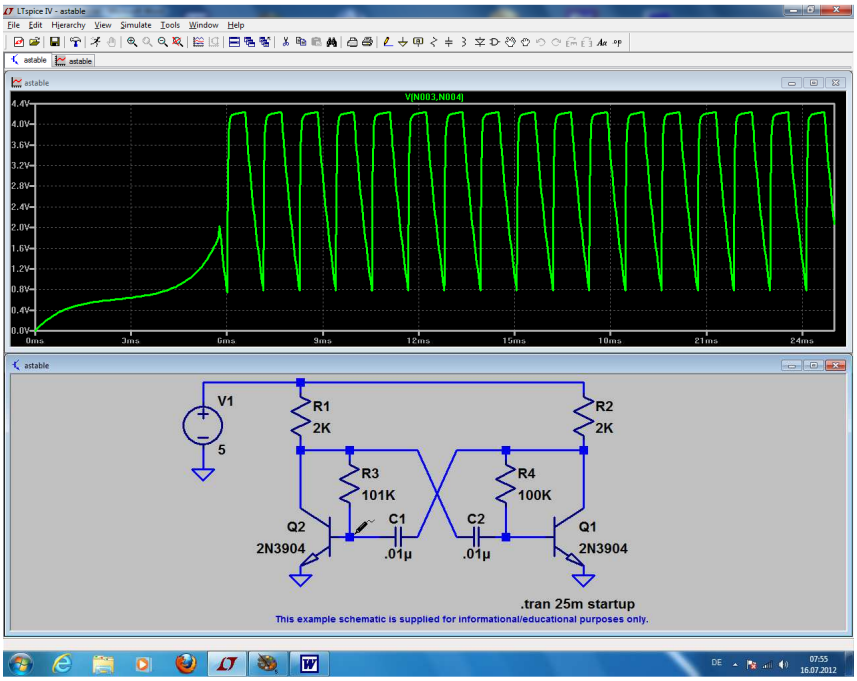
Man öffnet das Menü „**View**“ und findet darin die Funktion „**Set Probe Reference**“, die angeklickt wird.

Nehmen wir nun einmal an, dass uns die Spannung am Kondensator C1 (= an der Basis des linken Transistors) interessiert.



Nach „OK“ hängt plötzlich eine Prüfspitze am Cursor, die man direkt am Basisanschluss des linken Transistors durch Anklicken dieses Punktes absetzt.

Fährt man nun den Cursor auf den zweiten Anschluss des Kondensators (= Kollektor des rechten Transistors) und klickt auf die Leitung, dann ist man bereits am Ziel.
Falls gewünscht, kann man sich nun wieder interessierende Details herauszoomen.



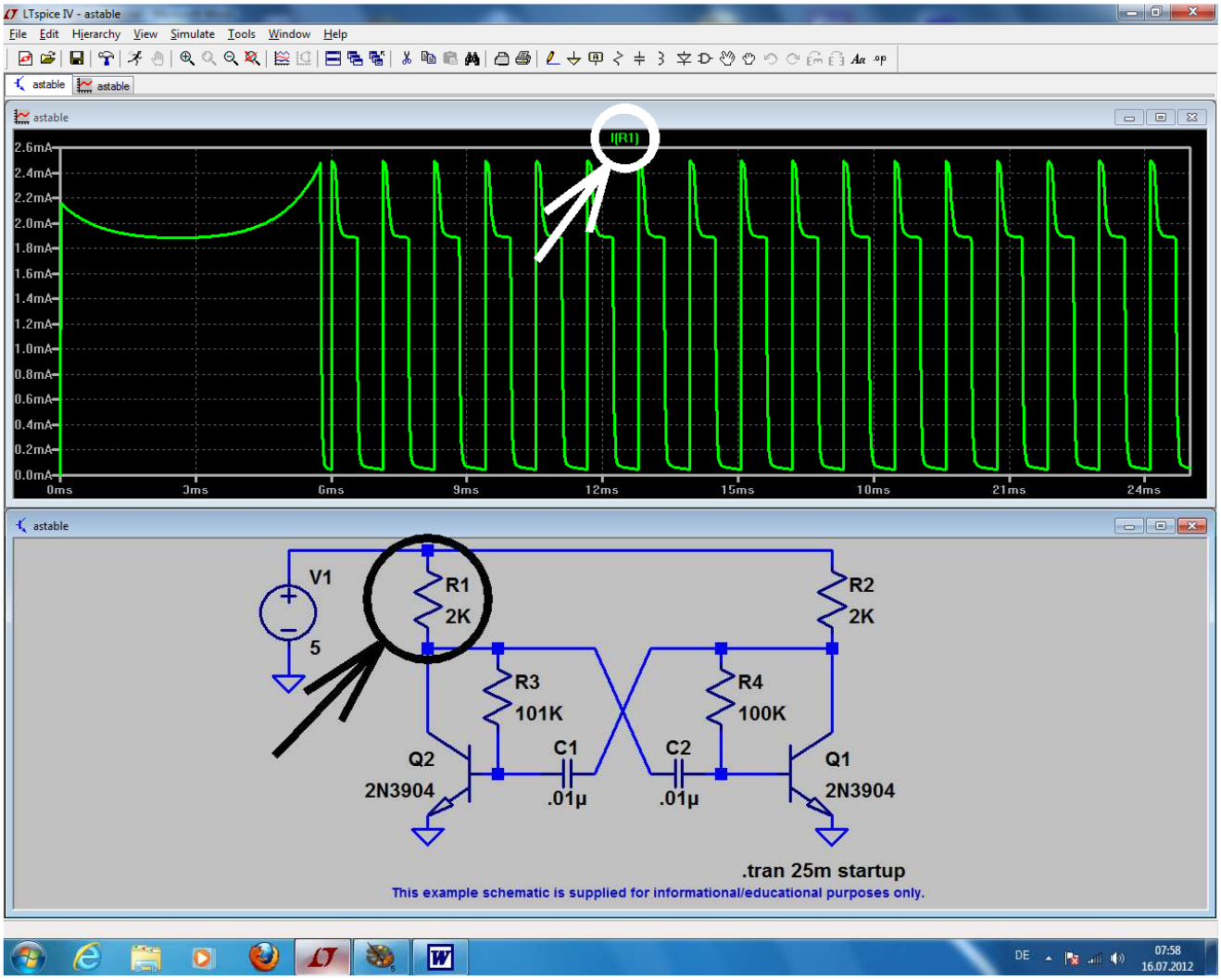
Übrigens:
Den Reference-Point bringt man durch einen Druck auf die Escape-Taste zum Verschwinden, wenn man wieder „normal“ (= gegen Masse) messen möchte.

4.9. Strom-Messungen

Dazu fährt (bei aktiviertem Schaltplanfenster) den Cursor **exakt auf das Schaltzeichen des Bauteils**, dessen Strom man simuliert haben möchte. Der Cursor nimmt dann plötzlich **das Aussehen einer Stromzange** an und ein linker Mausklick reicht, um die Anzeige des Stromverlaufs auszulösen.

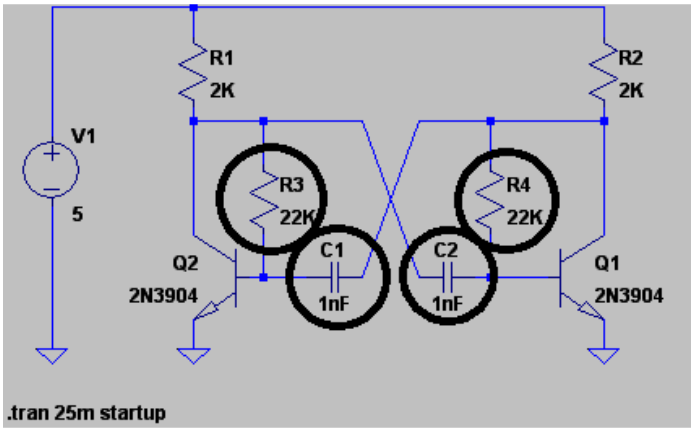
Allerdings ist es meist sinnvoll, den Strom in einem eigenen Diagramm (= new plot pane) darzustellen…schon wegen der anderen Einheit „Ampere"…

Beispiel:
Stromverlauf im Widerstand R1 (= Kollektorwiderstand des linken Transistors):



4.10. Änderung von Bauteilwerten

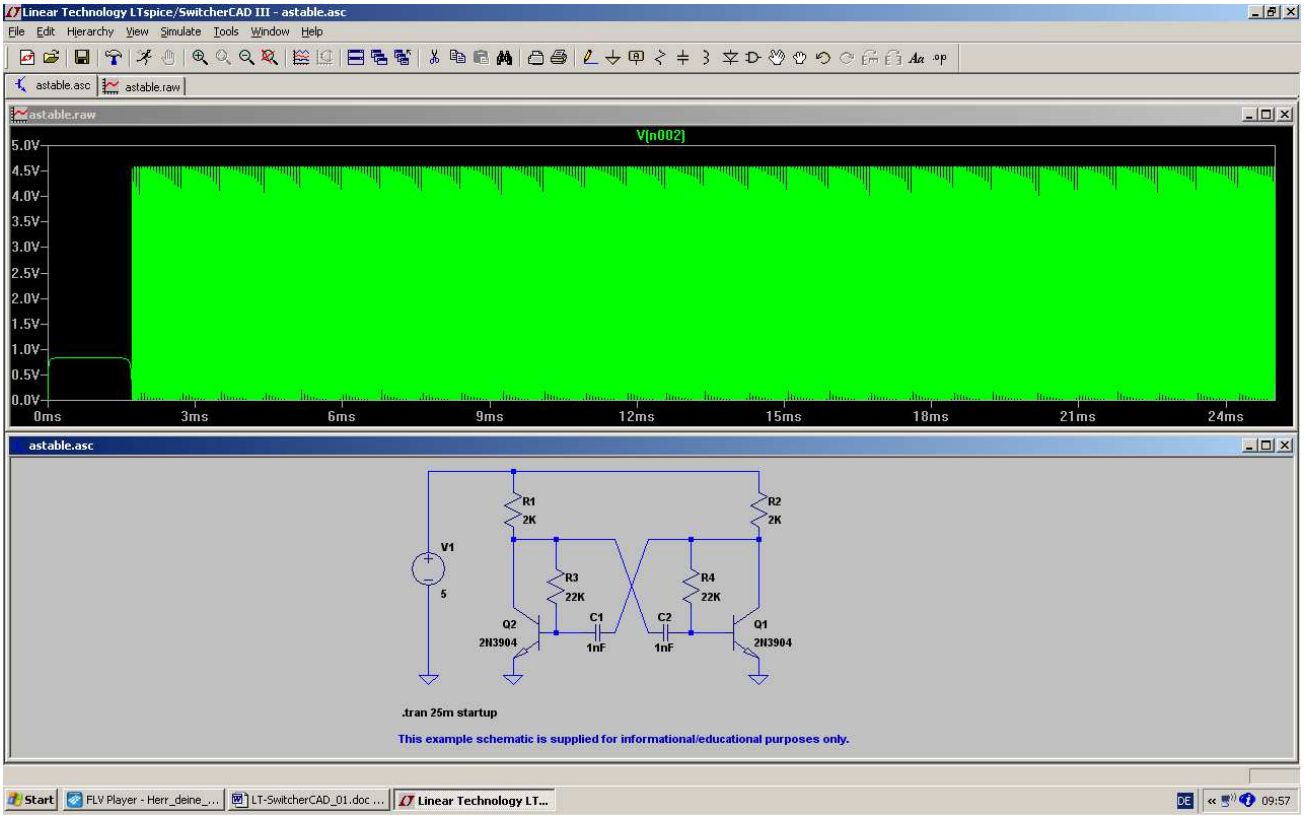
Nun wollen wir die Simulation mit anderen Bauteilwerten wiederholen und **ändern beide Kondensatoren auf je 1000pF sowie die beiden Basiswiderstände auf je 22kΩ ab**.



Dazu stellt man bei jedem Bauteil **den Cursor exakt auf die „WERTANGABE“** und drückt dann **die rechte Maustaste**. Im auftauchenden „Property Menu“ trägt man dann den korrigierten Wert ein und bestätigt mit OK.

So muss die Schaltung schließlich aussehen:

Jetzt wird die Simulation wiederholt (= Klick auf das rennende Männchen…) und schon bekommen wir folgendes Bild für die Kollektorspannung des linken Transistors:

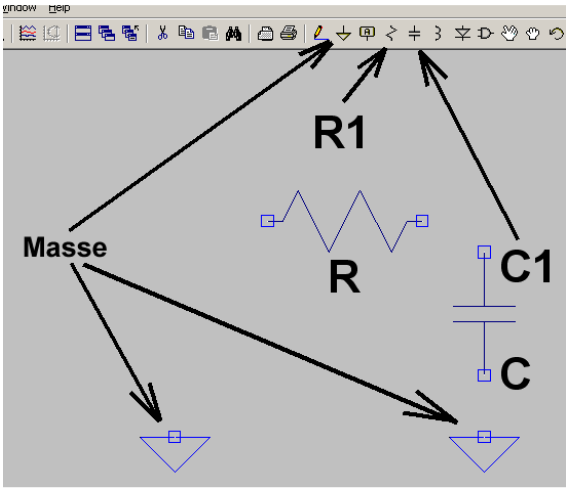


Da muss man nun natürlich mit der **Zoom-Funktion** drangehen oder gleich die Simulationszeit ändern, wenn man Details sehen oder die neue Frequenz messen möchte.

5. Erstes eigenes Projekt: RC - Tiefpass

5.1. Zeichnen des Stromlaufplans mit dem Editor

Wir holen uns zuerst unter „File“ ein neues Schematic-Blatt und speichern es in einem eigenen neuen Ordner (z. B. „Training“) unter einem eigenen Namen (z. B. „RC-LPF_01“).



Dann drücken wir den „**Button mit dem Widerstand**“, holen uns dort unser Widerständchen ab (...es hängt jetzt am Cursor...) **drehen** es mit dem Tastenkommando

<CTRL> + <R>

um 90 Grad und setzen es in der Schaltung ab. Anschließend klicken wir mit der **rechten Maustaste** kurz auf den Bildschirm (…oder drücken „**ESCAPE**“...), denn erst dadurch verschwindet das Widerstandssymbol am Cursor wieder.
Jetzt wiederholen wir die Prozedur mit dem **Kondensator** sowie **zwei Massezeichen** und haben dann den nebenstehenden Stand.

Dazu zwei Tipps:

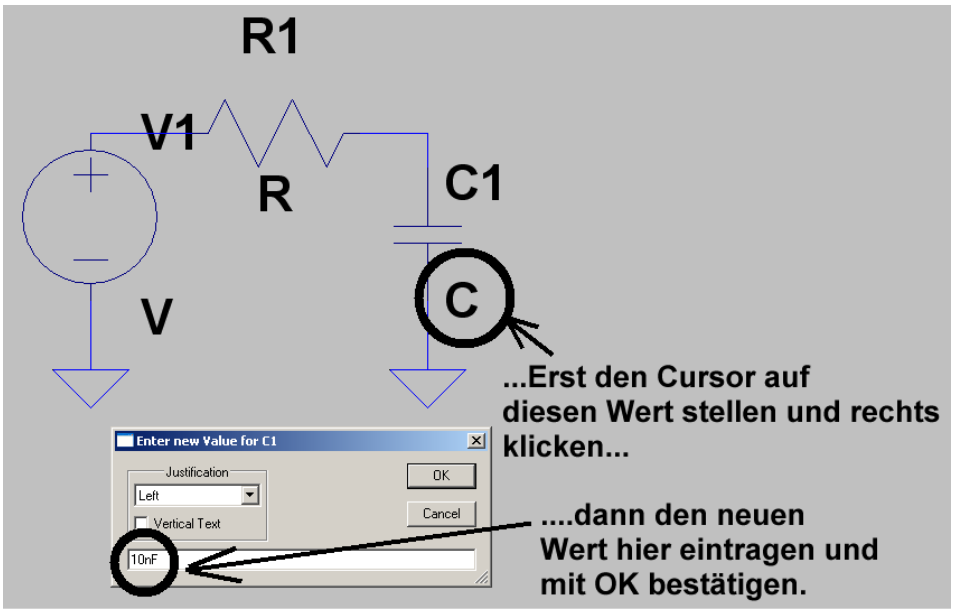
a) Möchte man ein Bauteil (oder später eine Verbindungsleitung) wieder **löschen**, dann drückt man erst auf die **Taste F5.** Sofort erscheint die **schon bekannte Schere als Cursor**! Fährt man damit direkt auf das Symbol und klickt links, dann verschwindet das betreffende Teil wieder vom Schirm.

b) Soll ein Bauteil dagegen auf dem Bildschirm **verschoben oder gedreht werden** (= MOVE), dann ist ein Druck auf die **Taste F7** erforderlich. Der Cursor wird zu einer **Hand,** die man auf das zu verschiebende Bauteil stellt. Klickt man dann mit der linken Maustaste, so ändert das Teil seine Farbe und hängt plötzlich am Cursor! So kann es nun beliebig mit der Maus verschoben werden und ein neuer linker Klick setzt es an der neuen Position ab.

Bitte dran denken: aus solchen Sonderfunktionen kommt man immer mit der „ESCAPE“-Taste wieder zurück zum Hauptmenü!

Jetzt fehlt uns noch die Spannungsquelle und die ist leider in der Menüleiste nicht aufgeführt. Da hilft nur die „**Bauteiltaste**“ in der Menüleiste.
In der auftauchenden Component-List (…darin ist alles aufgeführt, womit man arbeiten kann…) gehen wir zu „**voltage**“ und bestätigen mit OK. Dann hängt die Quelle am Cursor und kann in die Schaltung eingesetzt werden.
Jetzt wollen wir verdrahten und benötigen dafür die **WIRE-Funktion**. Bitte auf die **Taste mit dem Bleistift** klicken und genau hinschauen: auf dem Bildschirm muss nun ein **Fadenkreuz** zu sehen sein! Klickt man nun auf einen Bauteilanschluss und rollt dann die Maus, dann spult sich ein Verbindungsdraht ab. Man zieht ihn bis zum nächsten Bauteilanschluss und klickt erneut. Dann sollte die Verbindung hergestellt sein.
Wichtig:
Für eine „abgeknickte Verbindung“ muss man erst an der Knickstelle einmal klicken. Nur dann lässt sich der Rest der Leitung in einer anderen Richtung verlegen.

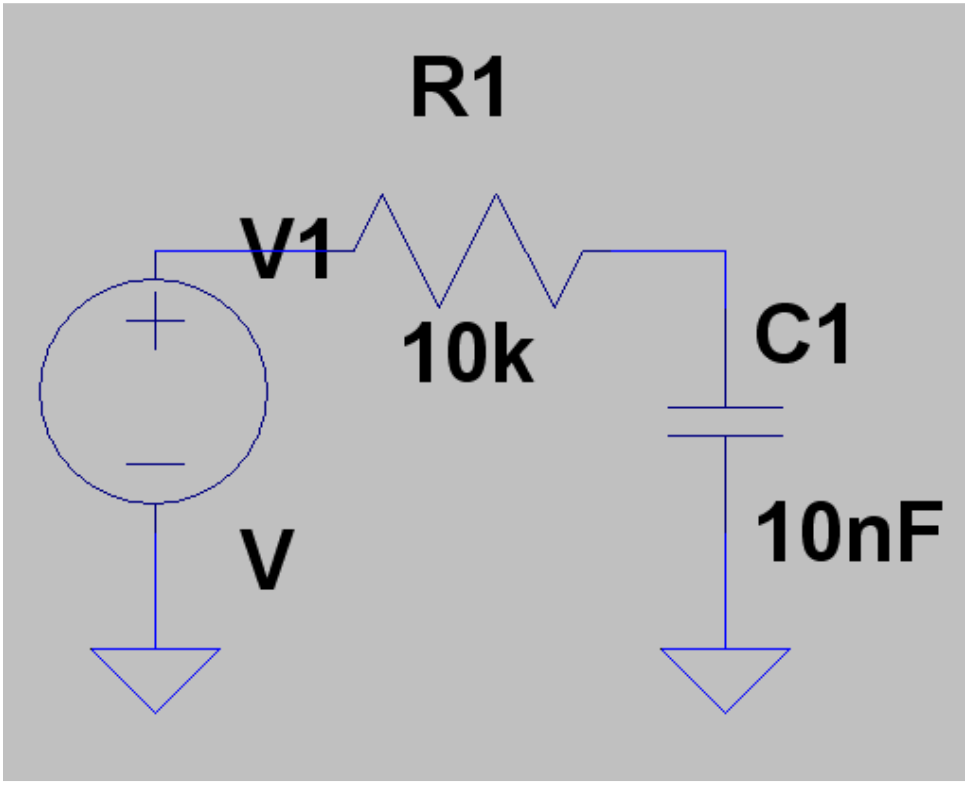
5.2. Zuweisung neuer Bauteilwerte



Die Zuweisung oder Änderung eines Bauteilwertes (**10nF**) ist hier am Kondensator C1 demonstriert:

Erst auf das Wertefeld mit dem Cursor fahren, dann rechts klicken und schließlich den neuen Wert eingeben.

Bitte auf dieselbe Weise dem Widerstand R1 einen Wert von 10 Kilo-Ohm (= 10k) zuweisen!



So sollte das jetzt aussehen.

Jetzt fehlt nur noch die **Programmierung der Spannungsquelle**.

Das wird erst dann erledigt, wenn wir wissen, in welchem Bereich

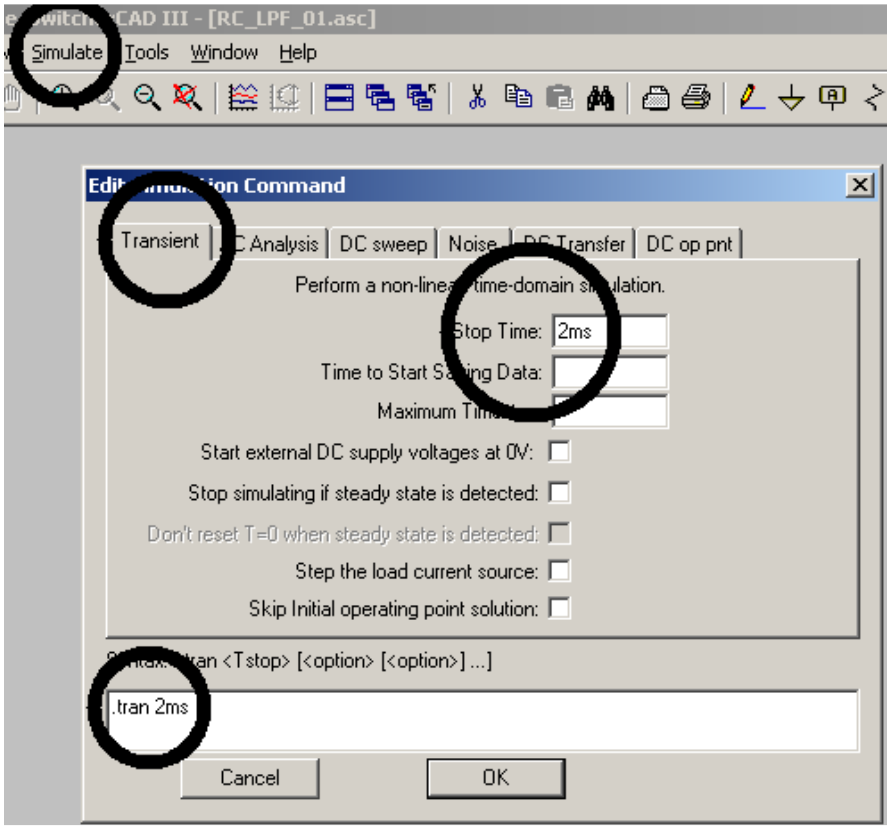
(= time domain oder frequency domain)

wir simulieren wollen.

5.3. Untersuchung von einmaligen Vorgängen im Zeitbereich (Time Domain)

5.3.1. Die Sprungantwort

Dazu programmieren wir zuerst eine „**Simulation im Zeitbereich**“, also in der **Time Domain (= Transient Simulation). Außerdem lassen wir die Eingangsspannung in sehr kurzer Zeit von 0V auf +1V springen.**

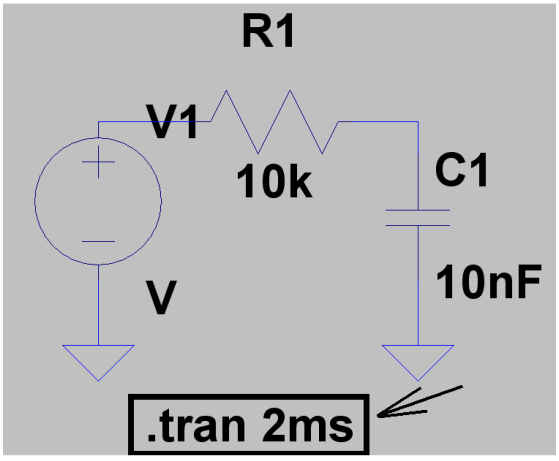


Hinter dem **Simulate-**Menü versteckt sich die Zeile „**Edit Simulation Command**“.

Wenn wir darauf klicken, öffnet sich das nebenstehende Transient-Menü. Darin tippen wir eine Stop Time von 2 Millisekunden ein und prüfen, ob das korrekt in das Anweisungsfenster als

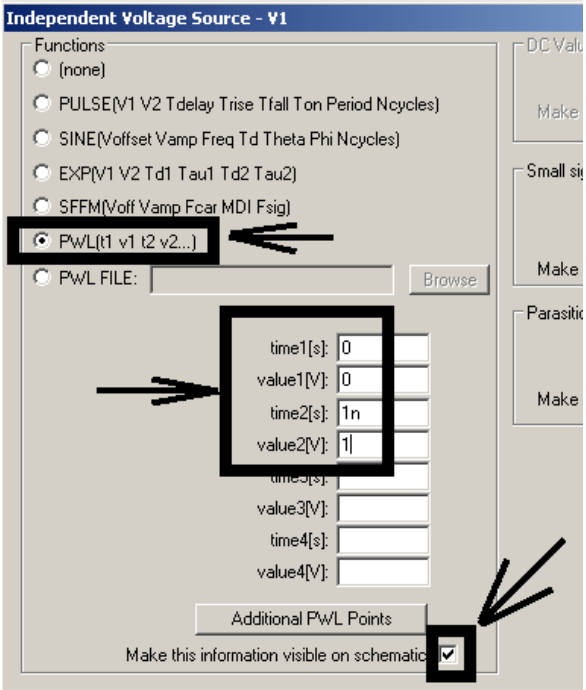
.tran 2ms

übernommen wurde. Bitte schließlich mit OK bestätigen.



Dieser Eintrag **„tran 2ms“** hängt anschließend am Cursor und muss im Schaltplan abgesetzt werden.

Jetzt fehlt nur noch die passende Eingangsspannung, nämlich ein **idealer Sprung von Null Volt auf +1 Volt**. **Dazu klicken wir mit der RECHTEN MAUSTASTE auf das Symbol der Spannungsquelle und öffnen anschließend das „ADVANCED“ – Menü.**

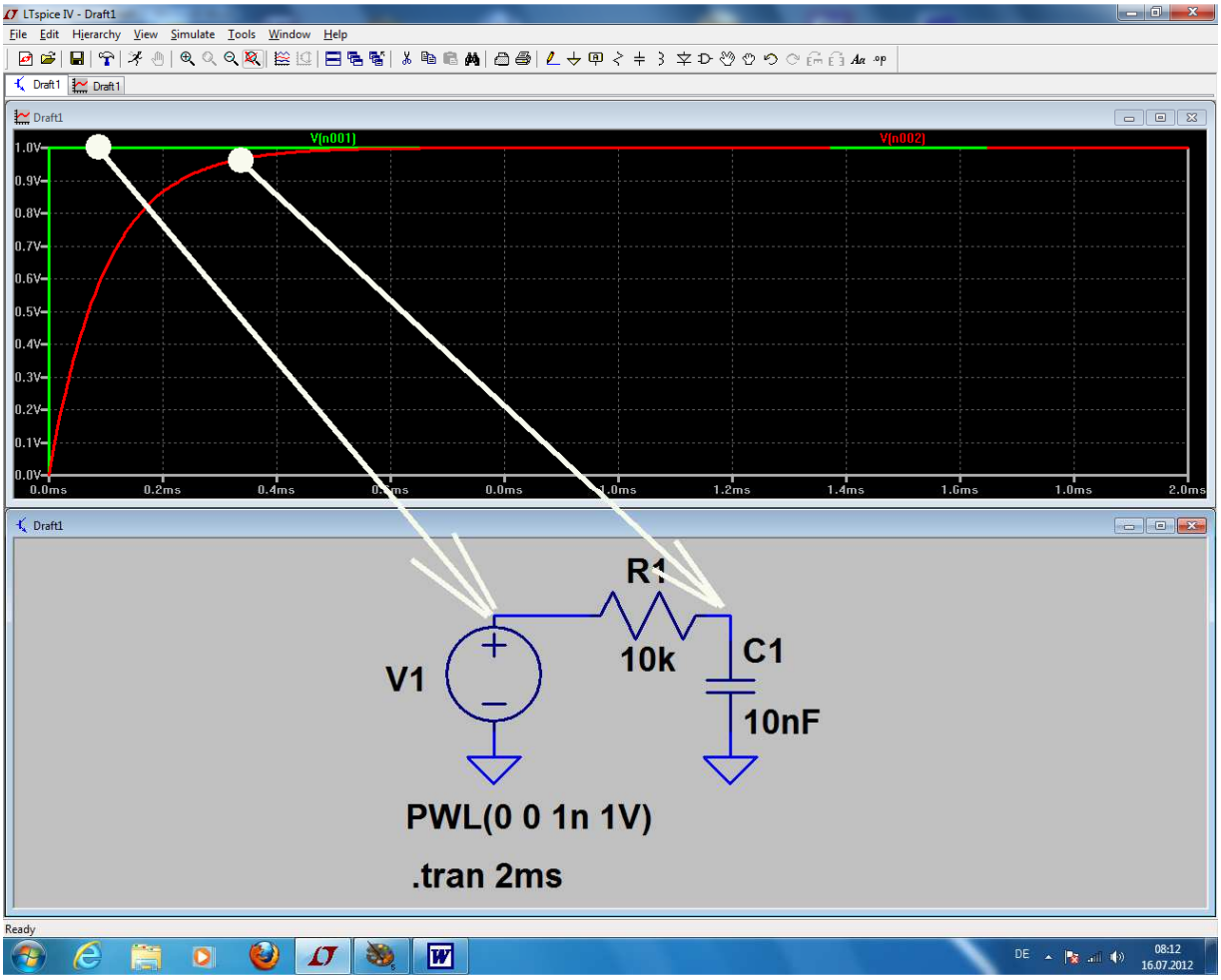


Wir setzen den Spannungsverlauf aus lauter geraden Stücken zwischen mehreren vorgegebenen Punkten zusammen. Der zugehörige Fachausdruck lautet:

PWL = piecewise linear

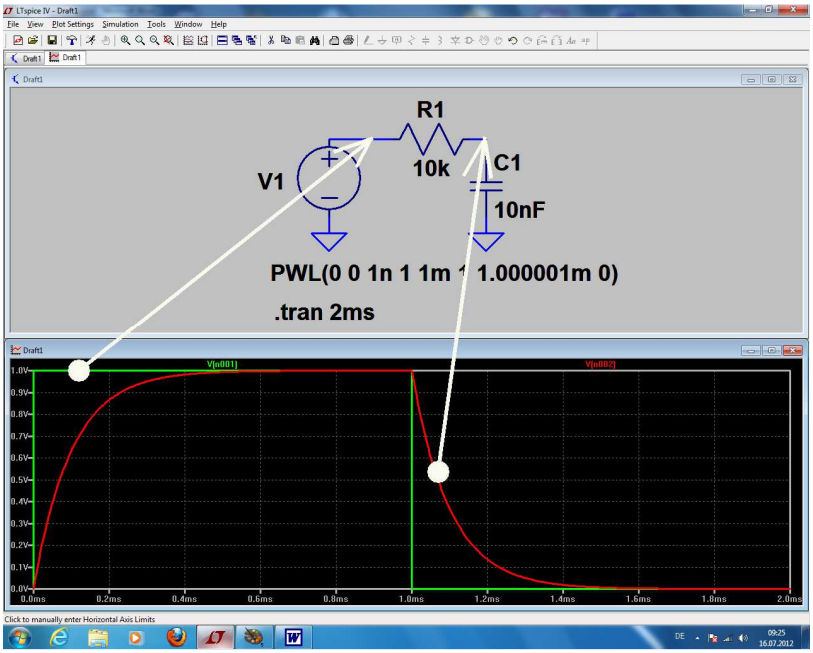
und er wird im linken Menü aktiviert.
Beim Nullpunkt wählen wir Null Volt und nach einer Nanosekunde soll die Spannung auf +1V angestiegen sein (…und auf diesem Wert bleibt sie anschließend). Diese Wertepaare trägt man in die auftauchende Tabelle ein und macht noch zum Schluss die Anzeige dieser Werte sichtbar.
Dann folgt „OK“.
Wer möchte, kann sich anschließend noch diese Zeile auf dem Bildschirm passend hinschieben (Wissen Sie es noch? Erst F7 drücken, dann mit der linken Maustaste auf die Zeile klicken. Sie hängt jetzt am Cursor und kann neu platziert werden. Schließlich mit einem neuen linken Klick absetzen und dann mit ESCAPE die MOVE-Funktion verlassen).

Nun kommt der Druck auf die Run-Taste mit den laufenden Männchen, das Anklicken von Eingangs- und Ausgangspunkt in der Schaltung und das Bewundern des Ergebnisses:



5.3.2. Ein- und Ausschaltvorgang

Kein Problem, denn dazu erweitern wir einfach unsere PWL-Liste um 2 Wertepaare. Sie muss dann so aussehen:



0 0 (Null Volt beim Startpunkt)

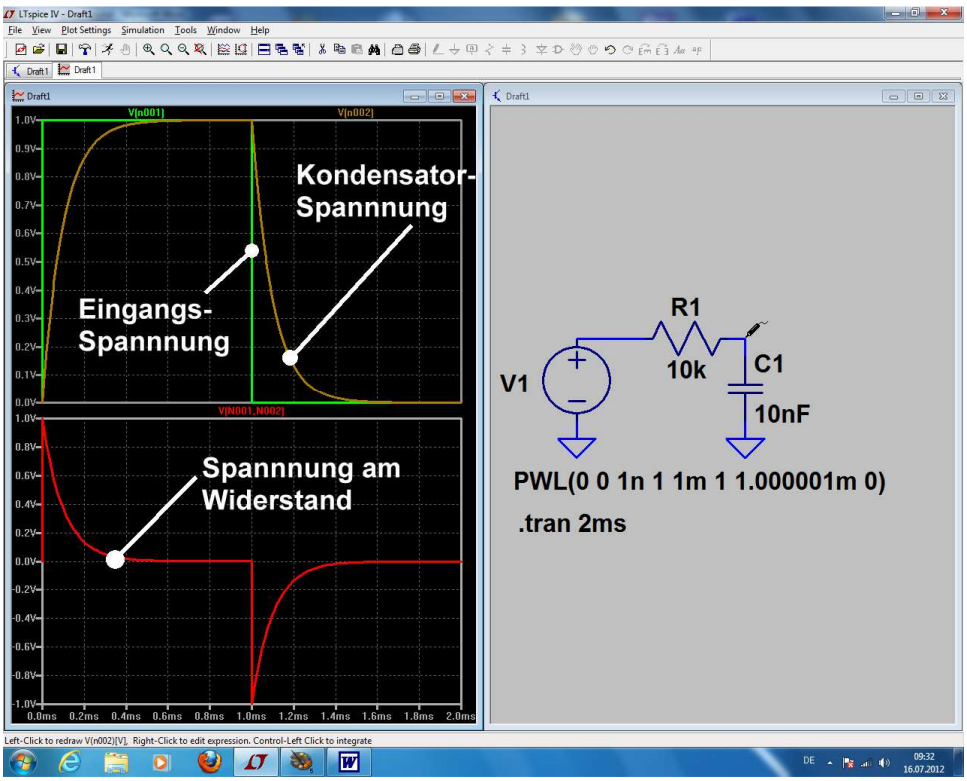
1n 1 (1 Volt nach einer Nanosekunde)

1m 1 (immer noch 1 Volt nach einer Millisekunde)

1.000001m 0 (eine Nanosekunde später wieder auf Null Volt zurück)

Zusatzaufgabe:
Stellen Sie zusätzlich die Spannung am Widerstand in einem getrennten Diagramm dar.

Lösung: wir klicken zuerst auf das Diagramm mit der rechten Maustaste und holen uns ein weiteres Diagramm mit „Add plot pane“.



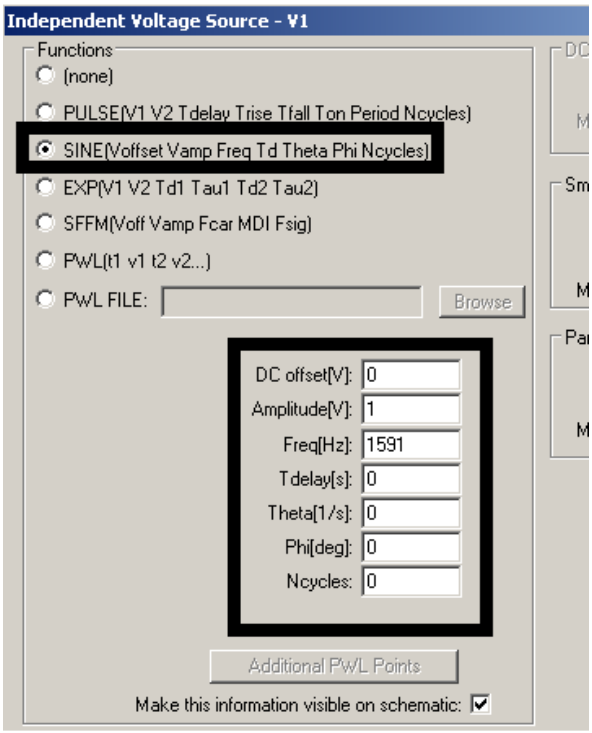
Dann setzen wir einen „**Reference Point**“ (…findet sich hinter „View“ als „Set Probe Reference“) am Ausgang (= Verbindung zwischen Widerstand und Kondensator). Tippt man dann an die Verbindung von Spannungsquelle und Widerstand, so sollte die gesuchte Spannung am Widerstand im Diagramm auftauchen. Und besonders hübsch wird das Ganze mit der Option „Tile Vertically“ hinter „Window“.

5.4. Periodische Signale am Eingang

5.4.1. Sinussignal mit f = 1591 Hz

Wir steuern den Tiefpass etwa bei seiner Grenzfrequenz an und speisen ihn dort mit einer Amplitude von 1V Spitzenwert.
Übrigens wird f_{grenz} so berechnet:

$$f_{Grenz} = \frac{1}{2 \bullet \pi \bullet R \bullet C} = \frac{1}{2 \bullet \pi \bullet 0{,}1ms} = 1591Hz$$



Dazu muss wieder das Symbol der Spannungsquelle rechts angeklickt und auf „Sinusbetrieb“ umgestellt werden.

Außerdem sind folgende Einträge nötig:

DC Offset = Null

Amplitude = 1 Volt (= Spitzenwert)

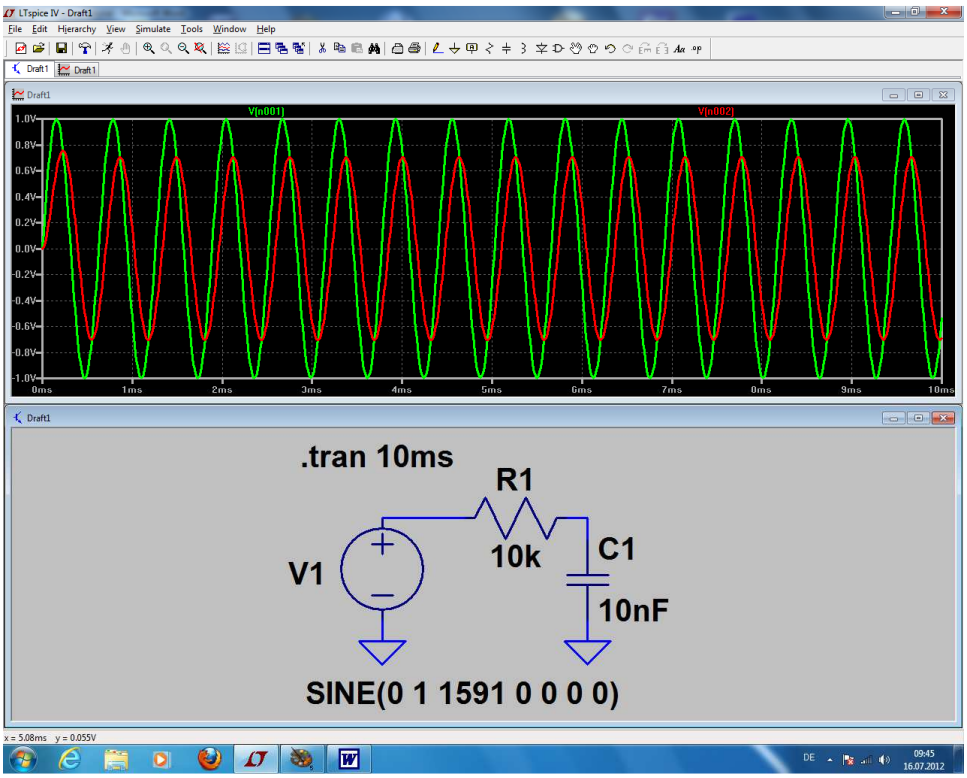
Frequenz = 1591 Hz

Tdelay = Null (= Startverzögerung)

Theta = Null (= Dämpfungsfaktor, Null ergibt Dauerschwingung)

Phi = Null (= Startphase der Schwingung)

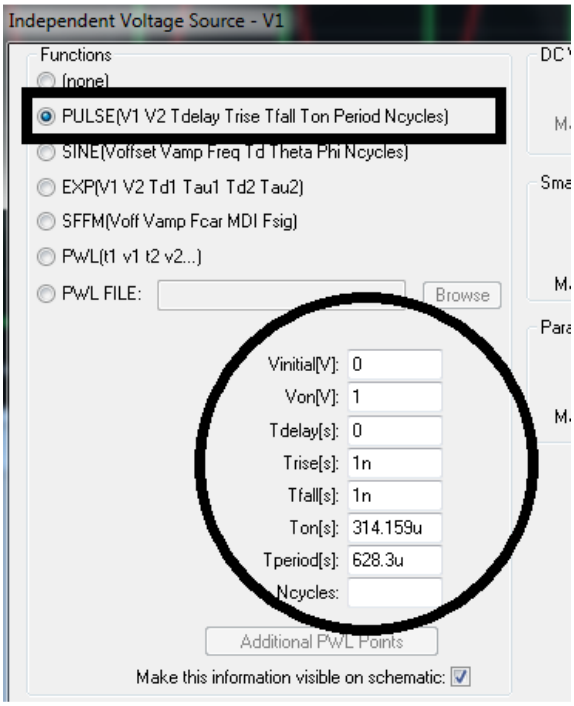
Ncycles = Null (ist dann wichtig, wenn ein „Burst“ aus wenigen Schwingungen programmiert werden soll)



Als Nächstes geht man in das „Simulate“-Menü, öffnet die Zeile „Edit Simulation Command“ und gibt einen Stopp bei 10 Millisekunden vor. **Nach der Simulation sieht man sehr schön die (zur Grenzfrequenz gehörende) Abnahme der Amplitude auf 70% und 45 Grad Phasen-Nacheilung bei der Ausgangs-spannung.**

5.4.2. Rechtecksignal mit f = 1591 Hz

Auch hier ist der erste Schritt wieder das Programmieren der Spannungsquelle.



Wir brauchen „PULSE“ und folgende Einträge in den Feldern:

Vinitial = Null (= Minimalwert der Spannung)

Von = 1 Volt (= Maximalwert der Spannung)

Tdelay = Null (= Startverzögerung)

Trise = 1 Nanosekunde (= Anstiegszeit)

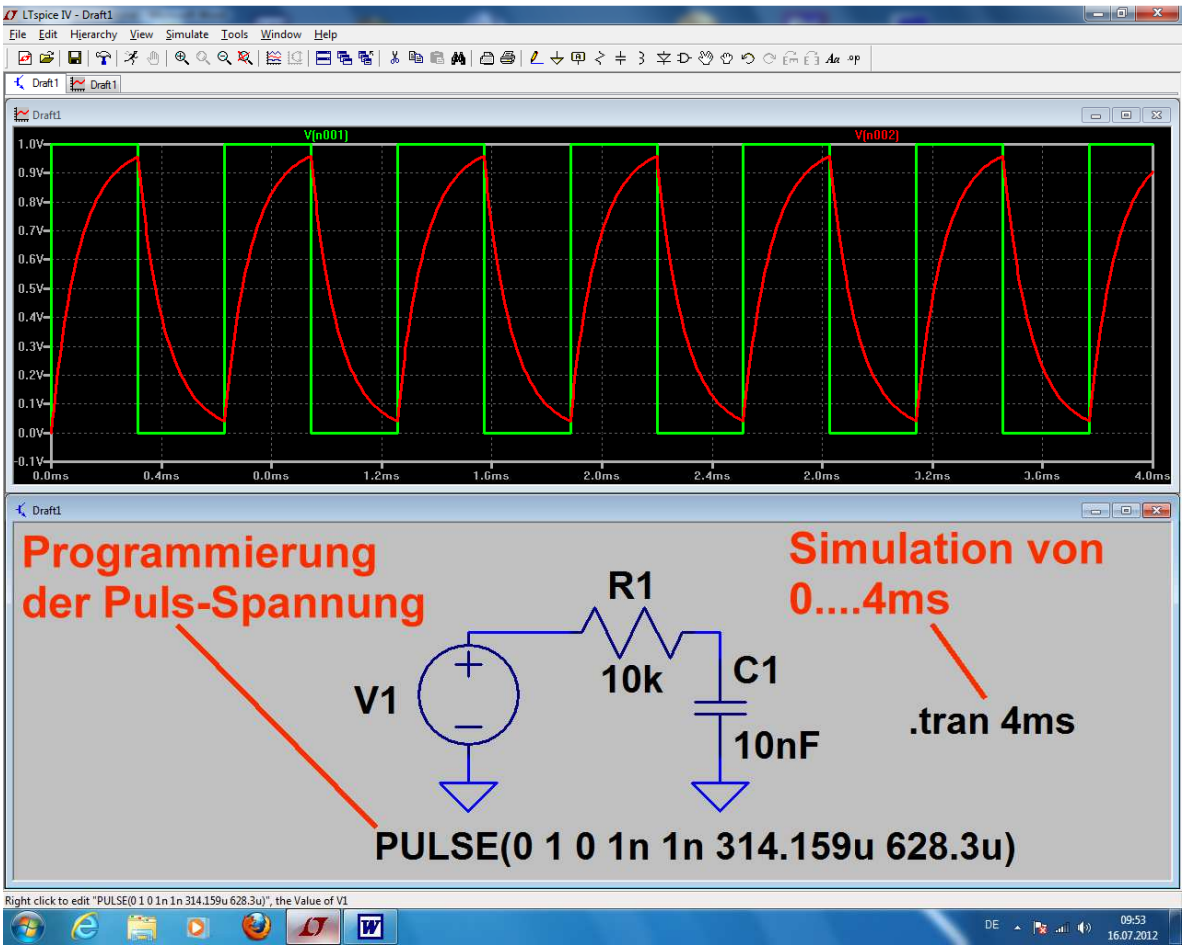
Tfall = 1 Nanosekunde (= Abfallzeit)

Ton = 314.159 Mikrosekunden (= Pulsdauer)

Tperiod = 628.3 Mikrosekunden(= Periodendauer)

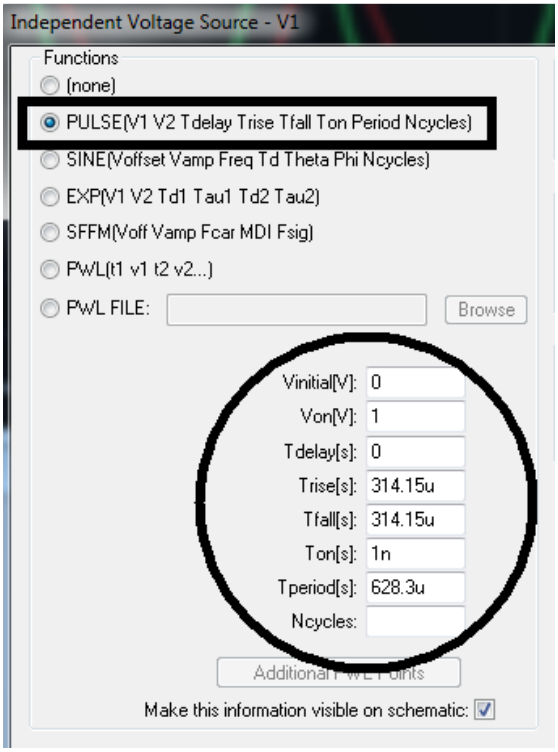
Ncycles bleibt ohne Eintrag (= Zahl der Zyklen, wenn ein Burstsignal erzeugt werden soll)

Wenn wir noch (über „Simulate“ und „Edit Simulation Command“) eine Simulationszeit von 0 bis 4 Millisekunden vorgeben, erhalten wir folgendes Bild:



5.4.3. Dreiecksignal mit f = 1591 Hz

Da wird es etwas schwieriger, denn dafür müssen wir die **Pulsquelle etwas zweckentfremden.** Wählt man nämlich die Anstiegs- und Abfallzeit sehr groß und die Pulsdauer extrem kurz, dann muss logischerweise ein Dreiecksignal entstehen.



Also programmiert man die Spannungsquelle folgendermaßen:

Vinitial = Null (= Minimalwert der Spannung)

Von = 1 Volt (= Maximalwert der Spannung)

Tdelay = Null (= Startverzögerung)

Trise = 314.15 Mikrosekunden (= Anstiegszeit)

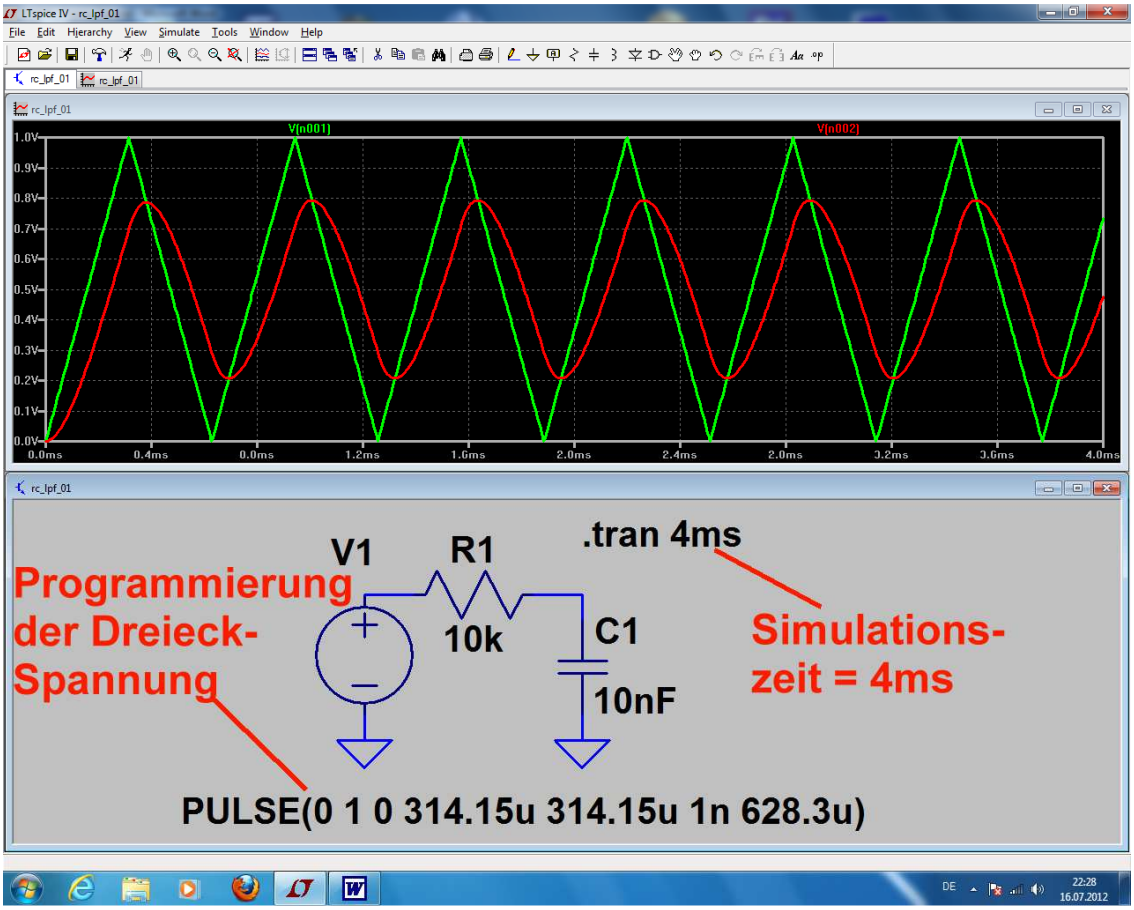
Tfall = 314.15 Mikrosekunden (= Abfallzeit)

Ton = 1 Nanosekunde (= Pulsdauer)

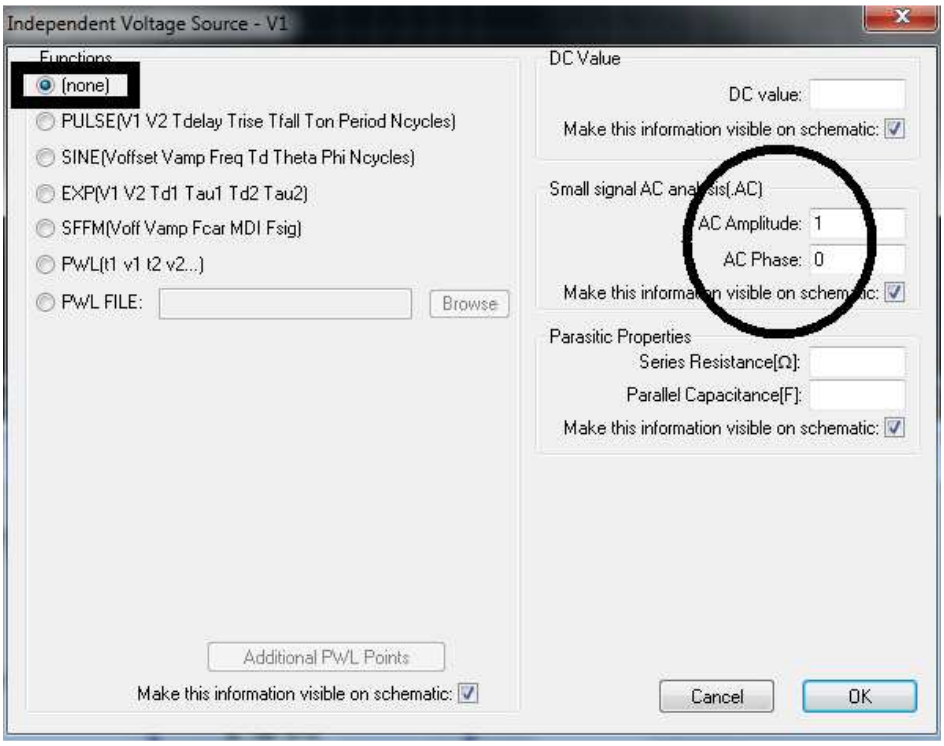
Tperiod = 628.3 Mikrosekunden (= Periodendauer)

Ncycles bleibt ohne Eintrag (= Zahl der Zyklen, wenn ein Burstsignal erzeugt werden soll)

Und so sieht das anschließend aus, wenn wieder die Zeit von 0….4 Millisekunden dargestellt wird:



5.5. AC-Sweep zur Ermittlung des Frequenzganges



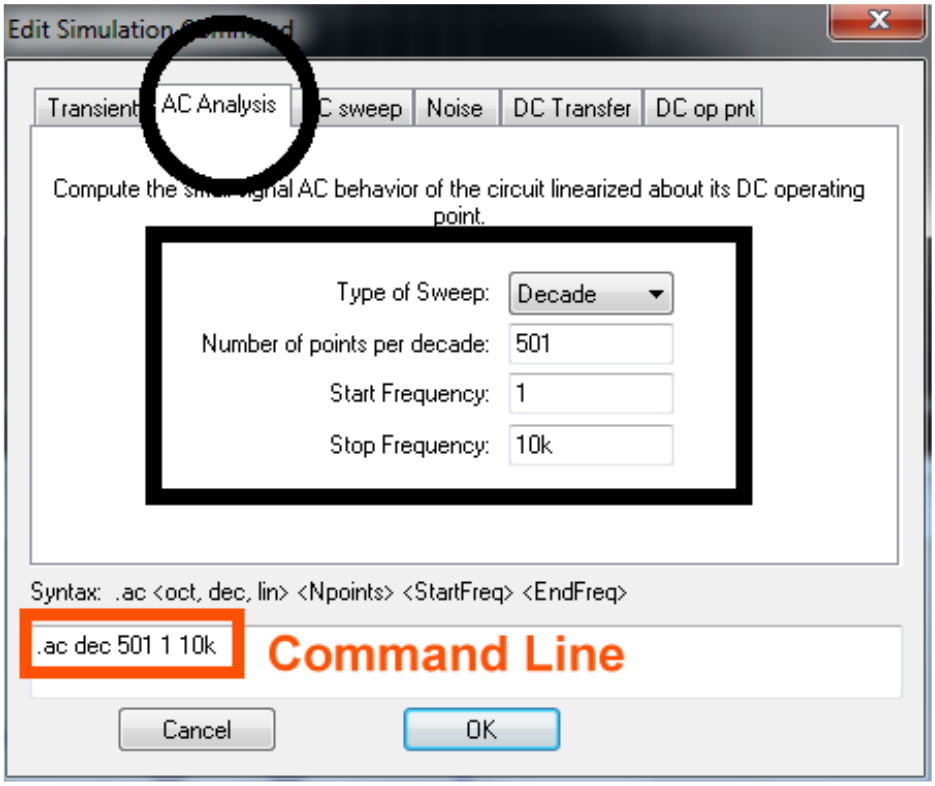
1. Schritt:
Wir stellen die Spannungsquelle auf **„AC Analysis“** um und geben vor:

AC Amplitude = 1 Volt

AC Startphase = Null

Außerdem beseitigen wir die Programmierung der Spannungsquelle für Simulationen im Zeitbereich durch die Aktivierung von

Function = none



2. Schritt:
Wir rufen unter „**Simulate**“ die Zeile **„Edit Simulation Command“** auf.

Im nebenstehenden Menü schalten wir gleich auf „**AC Analysis**“ um und geben folgende Dinge vor:

Dekadischer Sweep

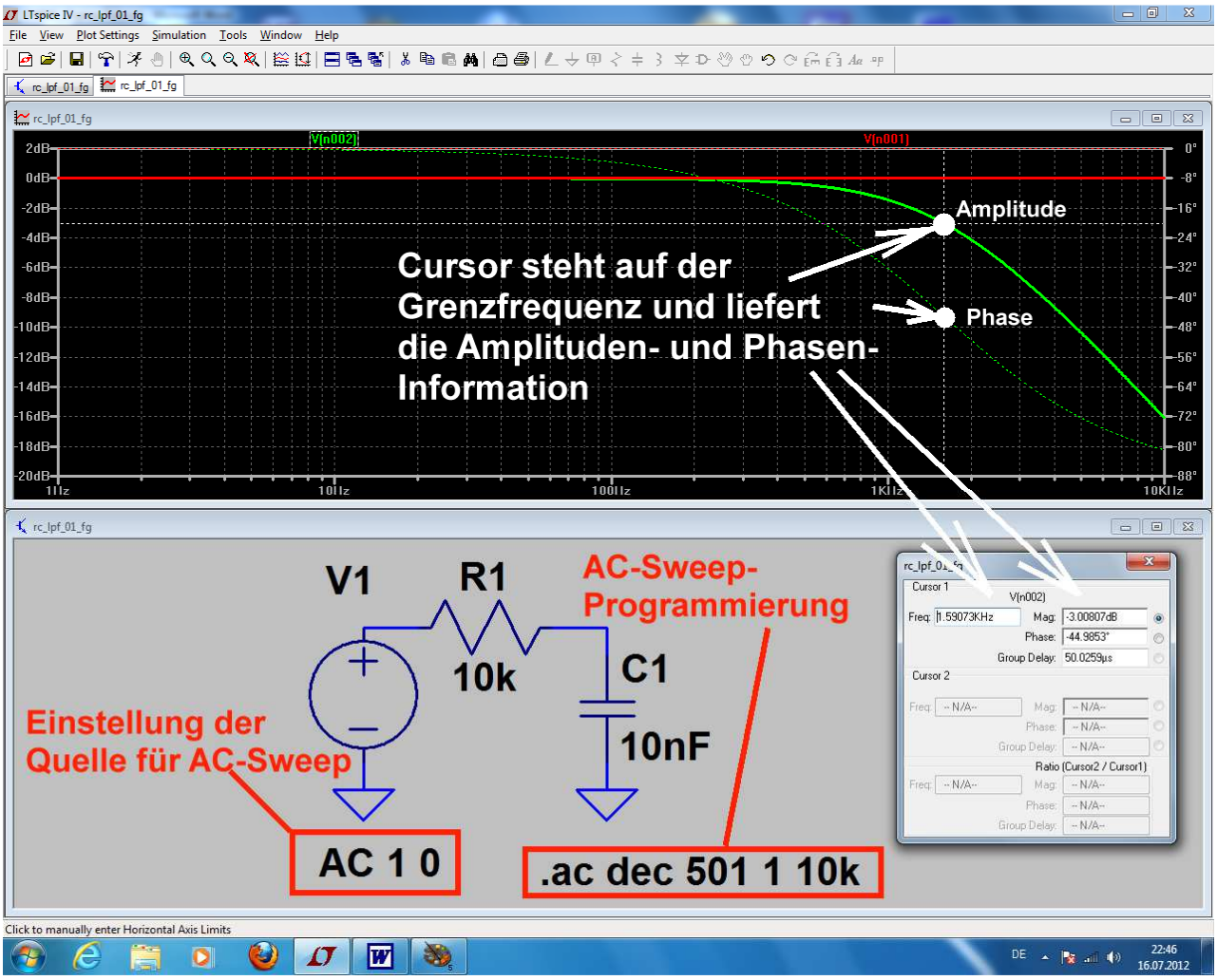
501 Punkte pro Dekade

Startfrequenz = 1Hz

Stoppfrequenz = 10kHz

Bitte die korrekte Übernahme der Vorgaben in die Kommandozeile (command line) nachprüfen!

So einen Bildschirm müssen wir dann nach der Simulation vor uns haben, wenn wir zusätzlich einen Cursor zur Kontrolle der Grenzfrequenz einsetzen.



Wen die Phasenanzeige stört:
Bitte auf die Skalenteilung der rechten senkrechten Achse klicken und im auftauchenden Menü

„Don't plot phase"

aktivieren.

Damit haben wir das erste eigene Beispiel geschafft!

5.6. Wirkleistung, Blindleistung, Scheinleistung

Besonders in der Leistungselektronik interessieren oft diese drei Größen. Nehmen wir einfach unseren Tiefpass aus dem vorigen Kapitel nochmals her und speisen ihn wieder mit derselben Sinusspannung (**U_{max} = 1V / f = 1591Hz**). Dann simulieren wir wieder von 0….4ms und lassen uns die Eingangsspannung anzeigen.

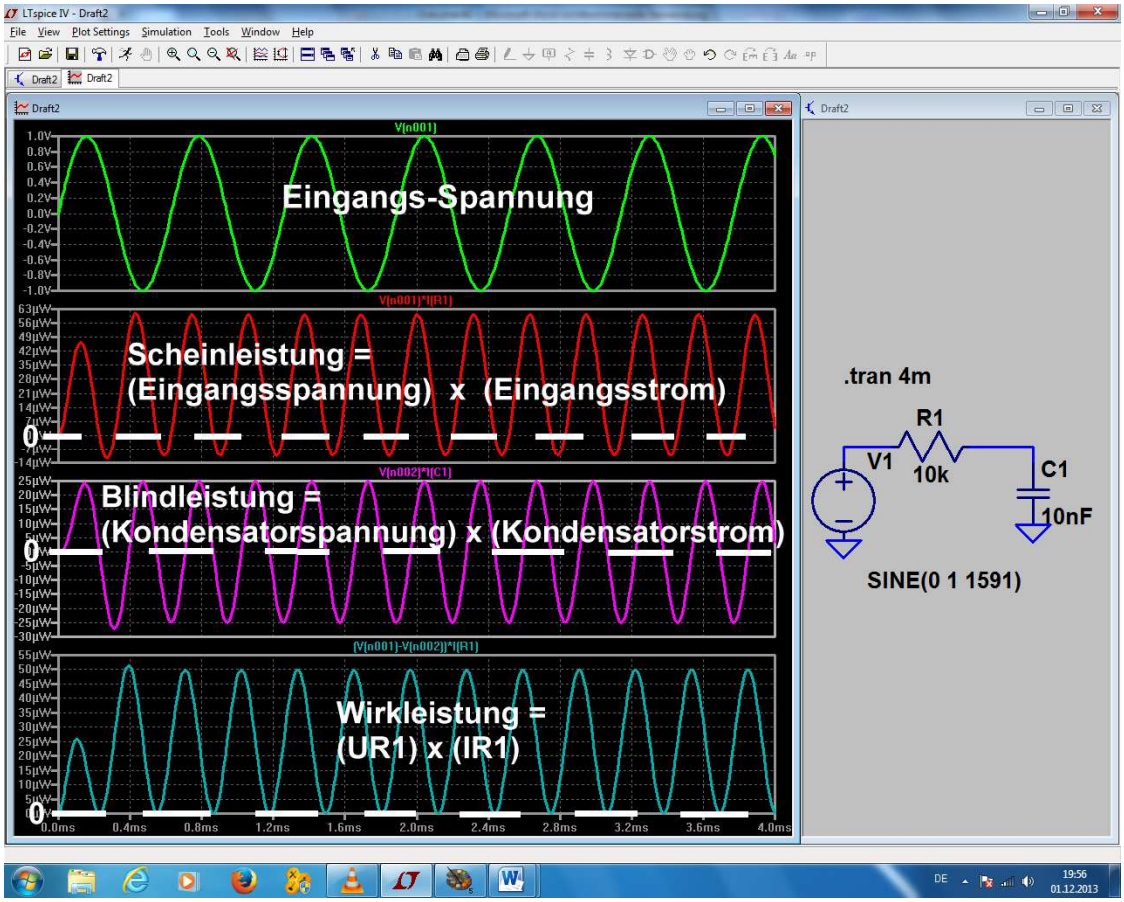
Nun holen wir uns zusätzlich über „**rechten Mausklick auf das Diagramm / Add Plot Pane**“ insgesamt 3 zusätzliche leere Ergebnisdiagramme. In eines dieser neuen Diagramme wird rechts geklickt und „**Add Trace**“ gewählt. Da findet sich in der linken unteren Hälfte des auftauchenden Menüs eine Zeile, in der wir **Verknüpfungen und Formeln eintragen können.**

Für die **Scheinleistung** gilt: **V(n001) * I(R1)**

Für die **Blindleistung** gilt: **V(n002) * I(C1)**

Für die **Wirkleistung** gilt: **U(R1) x I(R1) = (V(n001) – V(n002)) x I(R1)**

So sieht das Ergebnis aus:



Aus gegebenem Anlass (…mit meinen Studenten):
Wenn in diesem Diagramm z. B. die Wirkleistung mit negativen Werten ausgegeben wird, dann muss man nur den Widerstand im Schaltbild von seinen Zuleitungen lösen, das Schaltzeichen mit <CTRL> + <R> um 180 Grad drehen und dann wieder neu verdrahten. Hängt mit der Spannungs- und Stromdefinition im Modell zusammen…..

6. FFT (= Fast Fourier Transformation) mit LTSpice

6.1. Zusammenfassung der Grundlagen

Oft interessiert nicht nur der zeitliche Verlauf der einzelnen Spannungen (oder Ströme) in der Schaltung. Sobald nämlich ihre Kurvenform nicht mehr rein sinusförmig ist, gibt es außer der Grundwelle noch „Oberwellen“, also Signale mit doppelter, dreifacher, vierfacher Frequenz usw. Oder es handelt sich um ein „Gemisch“ oder um ein moduliertes Signal usw. und dann will man doch wissen, was da so alles drin steckt. Die **FFT** berechnet ein solches Frequenzspektrum aus dem zeitlichen Verlauf des Signals.

Allerdings ist eine ganze Reihe an Vorüberlegungen bereits für die Simulation im Zeitbereich nötig, wenn die FFT anschließend befriedigende und korrekte Werte liefern soll:

==========

Die Startfrequenz des simulierten Frequenzspektrums UND die Breite der dargestellten Linien UND der kleinste mögliche Frequenzschritt im Spektrum werden durch die **Simulationszeit in der Time Domain** bestimmt:

Maximaler Frequenzschritt = Startfrequenz = Linienbreite = 1 / Simulationszeit

Außerdem muss bei periodischen Signalen die **Simulationszeit immer ein ganzzahliges Vielfaches der Periodendauer des simulierten Signals sein** (…“es darf keine Periode angeschnitten werden“). Wird die Forderung verletzt, dann erhält man plötzlich zusätzliche „gelogene“ Linien und / oder ein starkes „Grundrauschen“ (Fachausdruck: Leakage).
(Zusatzbemerkung für ganz obergenaue Spezialisten: genau genommen muss dieser Wert der Simulationszeit noch um einen Timestep gekürzt werden, denn sonst wird schon wieder eine „neue Periode angeschnitten“. Diesen Fehler merkt man aber nur, wenn sehr wenige Perioden für die FFT benützt werden oder der Time Step sehr grob ist).

==========

Der „**Maximum Time Step**“ im Zeitbereich erfordert ebenfalls mehrere Überlegungen:

a) Er soll so klein sein, dass die simulierte Kurve nicht „eckig“ aussieht.

b) Die zugehörige „**minimale Samplefrequenz = 1 / Maximum Timestep“** muss **mindestens doppelt so hoch** sein wie die allerhöchste im simulierten Signal vorkommende Oberwellen-Frequenz. Erst dann ist die „Shannon-Bedingung“ erfüllt und man braucht nicht mit „Aliasing-Effekten“ zu rechnen.

c) Für die FFT muss eine bestimmte **Mindestmenge „echter“ Samples** zur Verfügung stehen, denn das bestimmt (**neben dem Maximum Timestep und der dadurch festgelegten Samplefrequenz) die höchste im Spektrum dargestellte Frequenz UND den Dynamikbereich der Amplitudenanzeige**. Deshalb macht es keinen Sinn, viel mehr Samples für die Verarbeitung in einer FFT vorzugeben als tatsächlich simuliert wurden (…gibt bisweilen Probleme und gelogene Details beim Spektrum und die Interpolation des Programms hat viel Arbeit..)

d) Diese Anzahl echter Samples kann leicht durch die Beziehung

Echte Samplezahl = Simulationsdauer / Maximum Timestep

ermittelt und kontrolliert werden.

==========

LTSpice nimmt bei den errechneten Samples im Zeitbereich meist eine **Datenkompression** vor -- und **genau das wollen wir nicht**, weil wir ja möglichst viele „echte“ Werte für die FFT brauchen. Diese Datenkompression müssen wir durch eine selbst zu schreibende SPICE-Direktive

.option plotwinsize=0

abschalten.

==========

Die für eine FFT verwendete **Anzahl an Samples muss immer eine Zweierpotenz** sein. Ist kein Problem, denn LTSpice listet uns automatisch die gemäß dieser Bedingung zugelassenen Samplemengen zur Auswahl in einem Menü auf. Und immer muss man im Hinterkopf behalten:
Mehr Samples für die FFT geben bessere Amplitudenauflösung und höhere Grenzfrequenz, erhöhen aber entsprechend die Rechenzeit und die Größe der Ergebnisdatei……und setzen genügend „echte“ berechnete Samples voraus.
==========
Noch etwas für die Spezialisten: die Samplerate ist in LTSpice nicht immer konstant, denn wir geben ja nur den „Maximum time step“ vor und sorgen damit für eine bestimmte Mindest-Abtastrate. Bei kniffligen Signalstellen wird SPICE diese Abtastrate automatisch und ohne Vorwarnung erhöhen und wir merken das im Spektrum z. B. an unerklärlichen zusätzlichen Linien mit kleiner Amplitude oder an einem erhöhten „Grundrauschen bei hohen Frequenzen“. (Fachausdruck: z. B. „Rechen-Artefarkte“).
Der Effekt entspricht einer nicht genau vorhersehbaren zusätzlichen FM und wird gemindert, je kleiner wir den „Maximum Time Step“ bei der Simulation wählen. Dann steigt bei der Simulation der Samplevorrat entsprechend an, aber die zeitlichen Schwankungen beim Abstand von z. B. zwei aufeinander folgenden Werten sind kleiner. Das hilft umso mehr, je kleiner die Anzahl der für die FFT benützten Samples im Vergleich zu dem bei der Simulation angelegten Wertevorrat gewählt wird -- und deshalb sollte man möglichst alle simulierten Samples bei der FFT benützen.
==========
Und noch ein Tipp für die Praxis:
In den meisten Fällen reicht ein Anzeige-Umfang von 80dB aus -- ganz selten müssen es mal 100dB oder mehr sein. Die erforderliche Messtechnik zur praktischen Untersuchung einer Schaltung wird nämlich bereits bei den Werten ab 80dB recht aufwendig und teuer…..

Bitte bedenken, wie das bei einer Pegelangabe in dB so läuft:

-20dB	**entsprechen**	**Faktor 0,1**	**= 10%**
-40dB	**entsprechen**	**Faktor 0,01**	**= 1%**
-60dB	**entsprechen**	**Faktor 0,001**	**= 1 Promille**
-80dB	**entsprechen**	**Faktor 0,0001**	**= 0,1 Promille**

usw.

6.2. Dazu ein praktisches Beispiel:

Wir wollen das Spektrum eines symmetrischen Rechteck-Puls-Signals untersuchen und geben folgende Eigenschaften vor:

Frequenz = 10kHz
Periodendauer = 0,1 Millisekunde = 100 Mikrosekunden
Pulsdauer = 50 Mikrosekunden
Minimalspannung = Null Volt
Maximalspannung = +1V
Pulslänge = 500µs
Anstiegs- und Abfallzeit je 1Nanosekunde.

Mit einer **Simulationszeit von 20ms und einem „Maximum Timestep“ von 100ns** erfüllen wir folgende Forderungen und Bedingungen:

a) Die Simulationszeit von 20ms ist ein ganzzahliges Vielfaches der Periodendauer von 0,1ms.

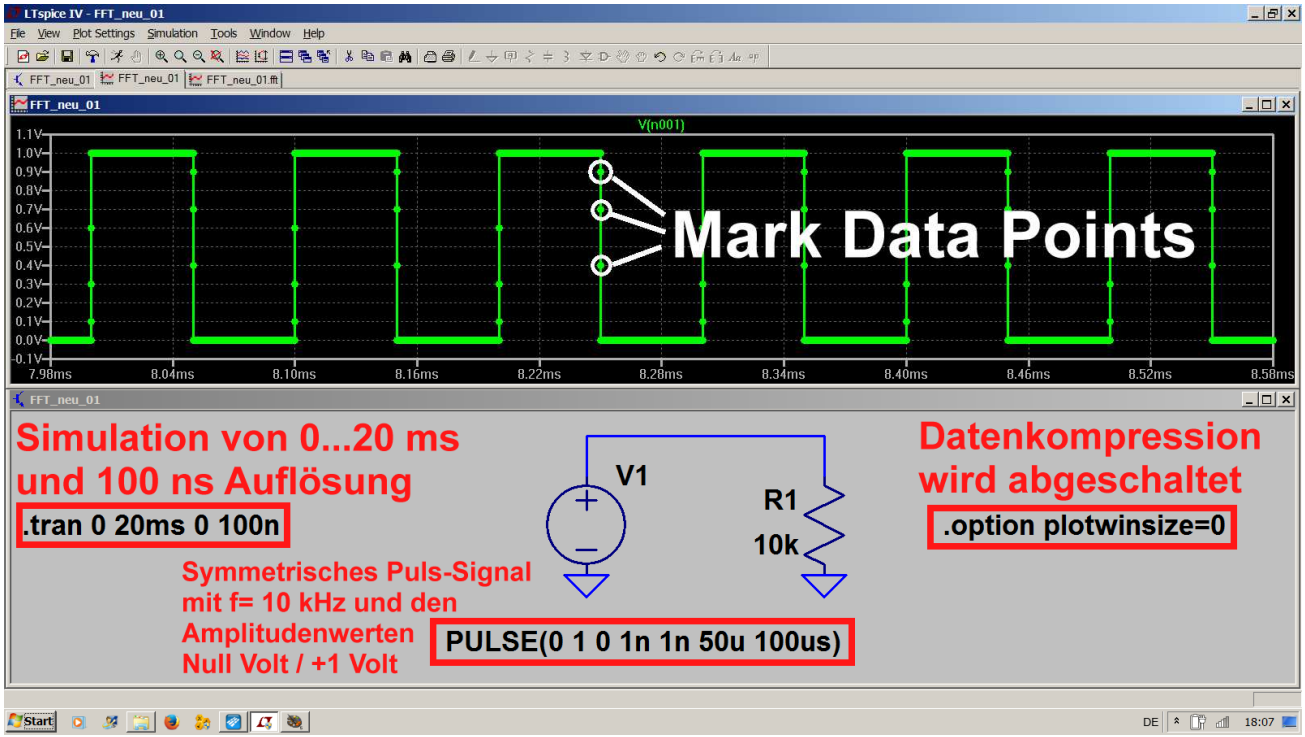
b) Der Maximum Timestep von 100ns ergibt eine minimale Samplefrequenz von 1/100ns = 10MHz. Die Schmerzgrenze = Shannon-Grenze = höchste zulässige Oberwellenfrequenz würde dann bei 5MHz liegen.

c) Die Anzahl der „echten vorhandenen Samples“ erhält man nun als 20ms / 100ns = 200 000 Samples. Also können wir anschließend im FFT-Menü gefahrlos mit

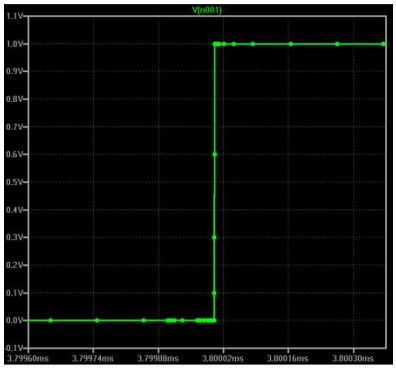
„131 072 sampled data points in time“

beginnen.

Jetzt zeichnen wir die Schaltung, programmieren die Spannungsquelle, legen Simulationszeit samt Timestep fest und schreiben die Direktive zum Abschalten der Daten-Kompression **(.option plotwinsize=0):**

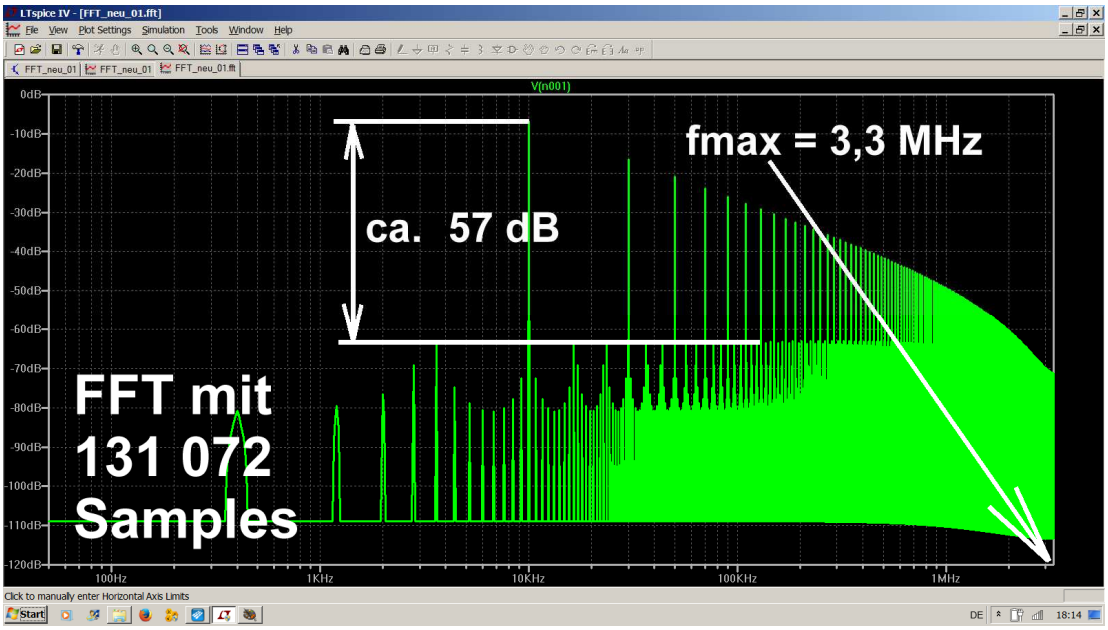


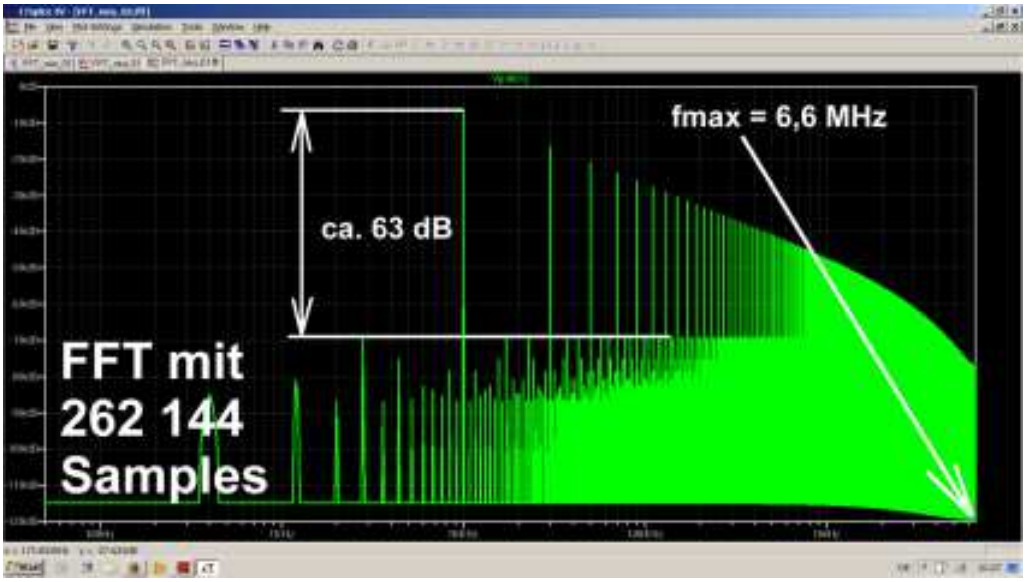
Wenn man das Ergebnis zoomt und anschließend nach einem rechten Mausklick auf das Diagramm „**Mark Data Points**“ aktiviert, dann sieht man sehr schön die einzelnen „Sample-Punkte“. Allerdings sieht man auch, wie die Samplefrequenz dem Kurvenverlauf angepasst wird und deshalb nicht konstant bleibt. Da die Anstiegs- bzw. Abfallzeit je 1ns beträgt, wird an den Punkten „0,1ns / 0.3ns / 0,6ns“ gerechnet und deshalb haben wir hier die vorhin erwähnte **Frequenzmodulation** der Samplefrequenz, die zu unerwünschten Ergebnissen führen dürfte.



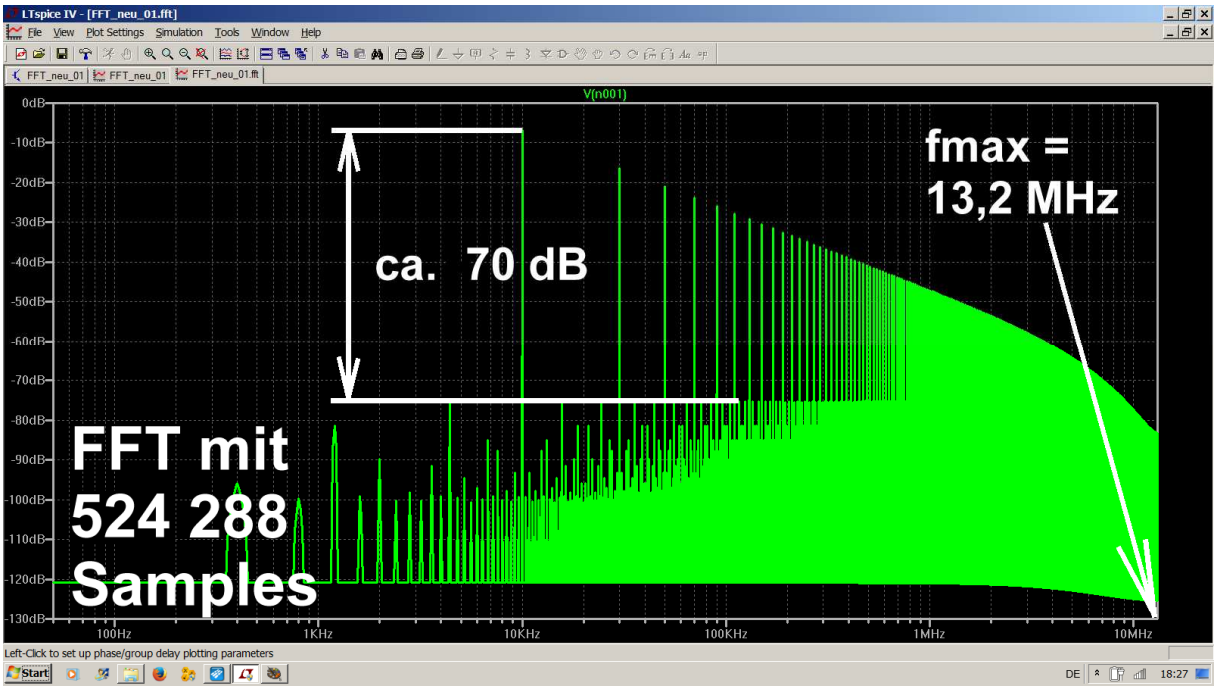
Sehr gut sieht man das im nebenstehenden Bild, bei dem der Kurvenverlauf sehr stark gezoomt wurde und dadurch die ungleichen zeitlichen Abstände bei den einzelnen Samples ganz deutlich zu erkennen sind.

Starten wir nun die FFT (= rechter Mausklick auf das Diagramm, dann „**View**“ und „**FFT**“). Wir stellen einen Vorrat von **131 072 Samples** ein und erhalten damit folgendes Spektrum. Der „SFR“ (= Spurious Free Range) beträgt etwa 57 dB bei einer maximal angezeigten Frequenz von ca. 3,3 MHz.



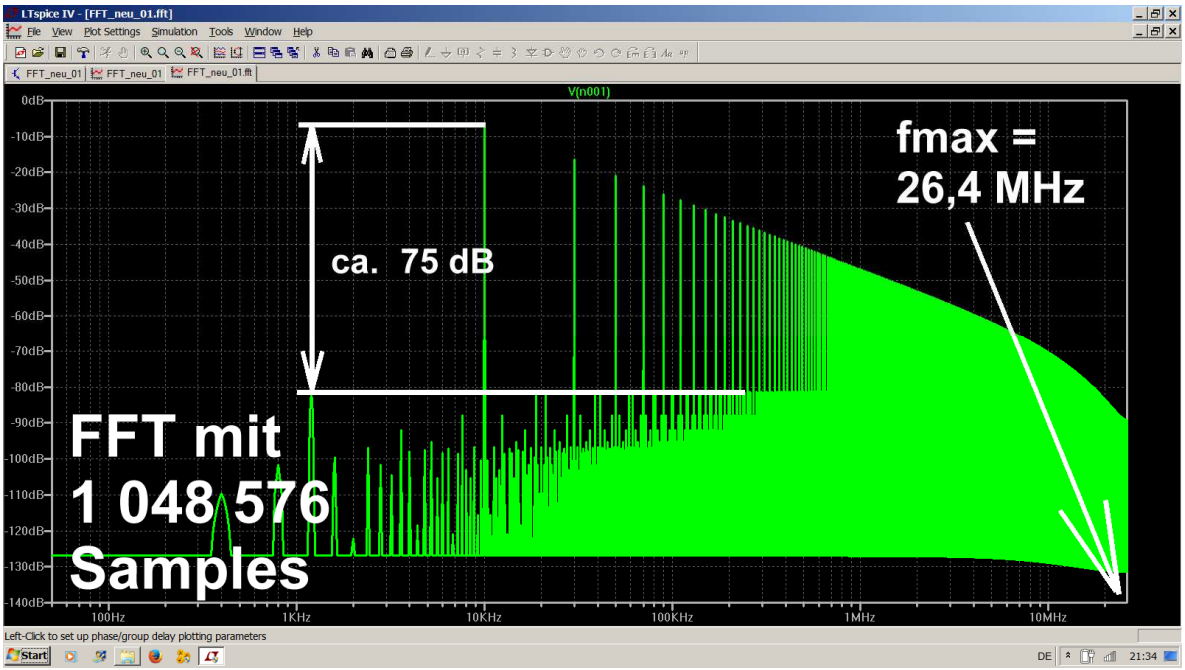


Wiederholen wir jetzt die FFT-Berechnung mit 262 144 Werten und steigern dadurch den SFR auf ca. 63 dB. Angezeigt wird nun bis ca. 6,6 MHz.



Also machen wir weiter, verlassen uns auf die gute Interpolation und sehen uns das Ergebnis bei 524 288 zur FFT verwendeten Werten an. Nun haben wir einen SFR von ca. 70 dB und eine höchste Frequenz von ca. 13,6 MHz im Diagramm:

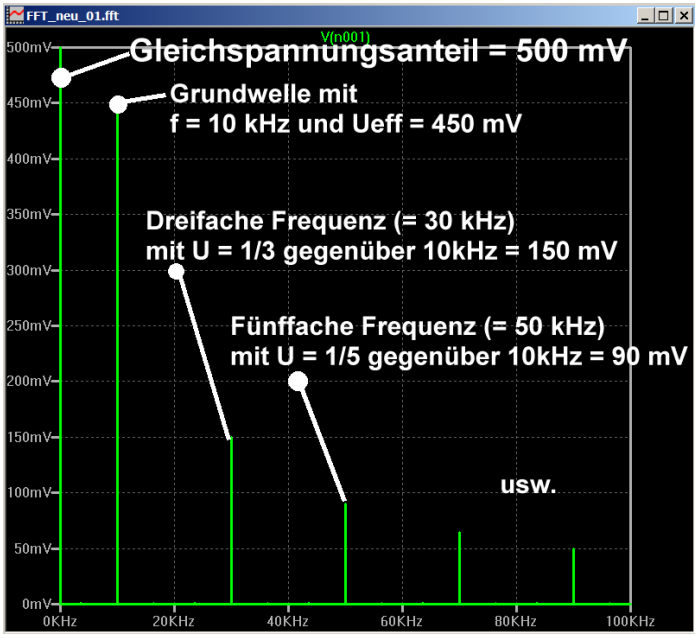
Wenn wir uns noch weiter steigern wollen, dann geht das vernünftig nur noch über mehr berechnete Samples. Das bedeutet längere Simulationszeit und / oder kleinere Time Steps. (Bitte aber daran denken: nur eine Verlängerung der Simulationszeit ergibt eine bessere Frequenzauflösung bei der waagrechten Diagramm-Achse).
Deshalb folgt hier noch das Ergebnis für 1 048 288 Samples -- aber zuvor wurde eine neue Simulation für 0…20ms, aber mit einem maximalen Timestep von nur 50ns durchgeführt. Jetzt bekommen wir ca. 75 dB beim SFR und eine höchste Frequenz von 26,4 MHz bei der waagrechten Achse.



So könnte man das weiter treiben…..

Oft braucht man eine lineare Darstellung beim Spektrum, um die Amplituden der einzelnen Linien exakt vergleichen zu können. Dabei ist es auch sinnvoll, den dargestellten Frequenzbereich zu verkleinern.

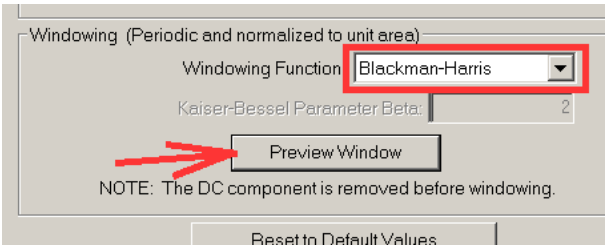
Unser Pulssignal ist ein gutes Beispiel dazu, denn im symmetrischen Rechteck gibt es nur Ungeradzahlige Oberwellen (= dreifache, fünffache, siebenfache…Grundfrequenz). Ihre Amplituden nehmen exakt mit dem Grad der Oberwelle ab.
Also fahren wir mit dem Cursor auf die Zahlenteilung der senkrechten Achse, bis das Lineal auftaucht. Ein linker Mausklick genügt und im auftauchenden Menü können wir „Linear“ wählen.
Ebenso machen wir es bei der waagrechten Frequenzachse. Dort wählen wir eine lineare Darstellung von 0….100kHz mit einem Tick von 10kHz. Und Siehe da……:

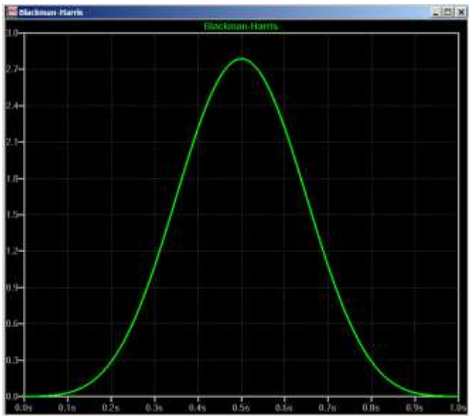


Achtung:
Die errechneten und dargestellten Spannungsamplituden sind stets Effektivwerte!

Übrigens:
Wer sich zusätzlich für die Phase interessiert, der klicke auf die rechte senkrechte Achse und dort auf „Phase“.

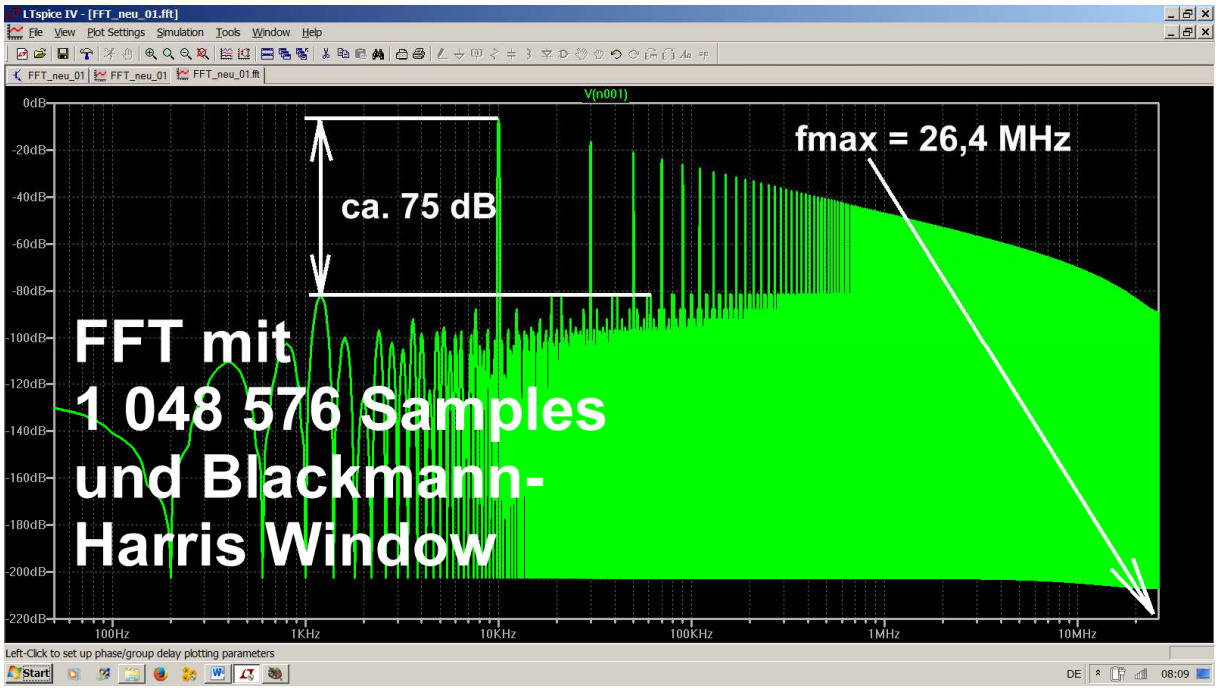
Nun noch einige weitere Tipps für die Zukunft:
Die dargestellten Spannungsverläufe beginnen und enden stets sehr abrupt (z. B.: vor dem Start = nix, dann plötzlich die volle Rechteckspannung. Beim Ausschalten folgt auf den Signalverlauf wieder „nix“). Diese plötzlichen „Schaltvorgänge“ erzeugen zusätzliche Störspektren, die sich dem gewünschten Nutzspektrum überlagern und man kann sie durch Verwendung einer **„Window“-Funktion** (die mit dem Signalverlauf multipliziert wird!) abschwächen.
Ebenso werden „Leakage-Effekte“ (= Simulationszeit und Periodendauer bilden kein ganzzahliges Verhältnis) weniger wirksam und das sollte insgesamt zu einer Verbesserung der Spektraldarstellung führen.





Sehr bewährt hat sich die „**Blackmann-Harris Windowing Function“** und man kann ihren Verlauf durch einen Klick auf „Preview Window“ im FFT-Menü sichtbar machen:
Man sieht, dass der Anfang und das Ende fast für 10% praktisch zu Null gesetzt werden und das gibt einen sehr sanften Einstieg bzw. Ausklang.

Doch nehmen wir uns das letzte simulierte Spektrum mal vor, legen „Blackmann-Harris“ auf den Signalverlauf und warten auf das Ergebnis. Dauert jetzt natürlich nochmals länger mit der Berechnung….



Bitte vergleichen:
Man sieht, dass der SFR-Bereich und die maximale dargestellte Frequenz gleich geblieben sind (….denn das hängt direkt mit der verarbeiteten Zahl von 1 048 576 Samples zusammen).
Deutlich tiefer gesunken ist die untere Pegelgrenze bei der senkrechten Achse, denn sie beträgt etwa -200dB statt -125dB.
Allerdings wurde die Zahl der Samples, die etwas zum Ergebnis beitragen, durch das Blackmann-Harris Window kleiner und das ist gleich bedeutend mit einer **kürzeren Simulationszeit**. **Die direkte Folge ist dann sofort eine geringere Frequenzauflösung** (= höhere Startfrequenz / breitere Linien) gemäß der Formel

Frequenzauflösung = 1 / Simulationszeit.

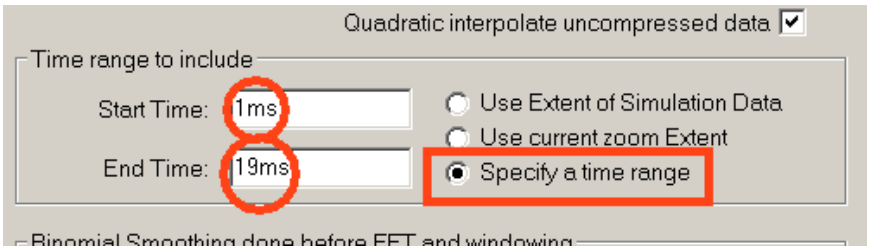
Die einzige wahre Lösung zur echten weiteren Verbesserung ist halt immer noch:

Rauf mit der Simulationszeit (….notfalls bis zur Minute…) und
Runter mit dem Time Step (…notfalls bis zur Picosekunde…)

Dazu ein superschneller Rechner, eine riesige (externe?) Festplatte und viel Zeit + Geduld.

==========

Noch ein wichtiger Hinweis:
Bei aktiven Schaltungen (Verstärker, Oszillatoren, Schaltnetzteile usw.) beobachtet man oft deutliche Einschwingvorgänge. Interessiert nur der Anfang ODER der nachfolgende „Dauerstrichbetrieb“, dann legt man über das gewünschte Stück des Signalverlaufs ein **Zeitfenster und blendet den nicht interessierenden Teil aus.**



So würde ein **Zeitfenster von 1ms bis 19ms** aussehen

Auf eine immer wieder gestellte Frage wollen wir noch eingehen:

Wie legt LTSpice in seinem berechneten FFT-Spektrum die höchste angezeigte Frequenz fest?

Dazu brauchen wir nochmals unser voriges Beispiel mit seinen Einstelldaten:

Simulationszeit = 20ms und Maximum Time Step = 50ns ergeben eine maximale Sample – Frequenz von f_{sample}= 1/50ns = 20MHz

Und der Rest geht so:
Beim Übergang auf die FFT benützen wir jedoch 1 048 0576 Samples für eine gesamte Simulationszeit von 20ms. Deshalb gehört nun zu jedem einzelnen Sample-Wert ein

Timestep = 20ms / 1 048 576 = 19,073486ns

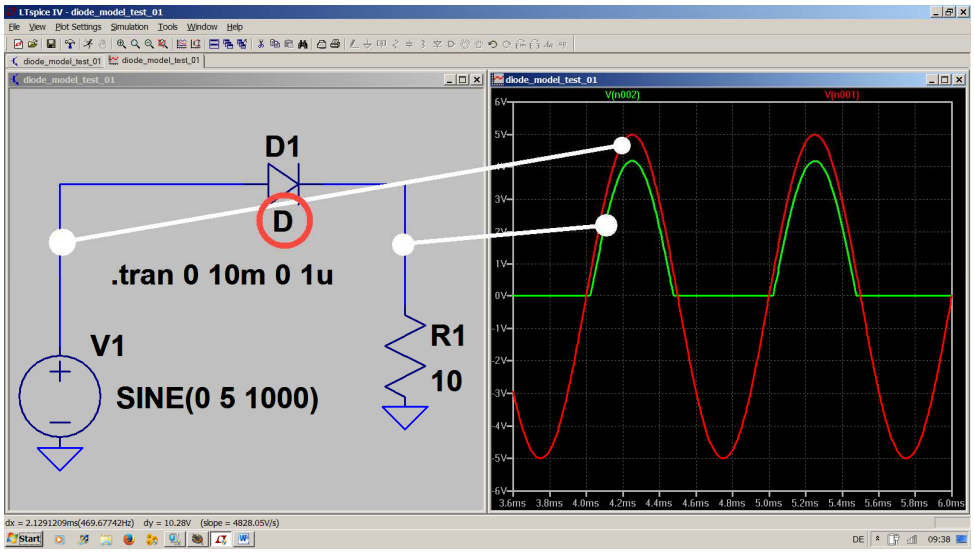
Das entspricht aber einer „**scheinbaren Samplefrequenz“ von 1/19,073486ns = 52,4288MHz**

Die Shannon-Bedingung sagt aber, dass wir beim FFT-konvertierten Signal nur bis höchstens zur halben Samplefrequenz gehen dürfen, um falsche Darstellungen (= Aliasing) zu vermeiden. Das wären dann 26,2144MHz.

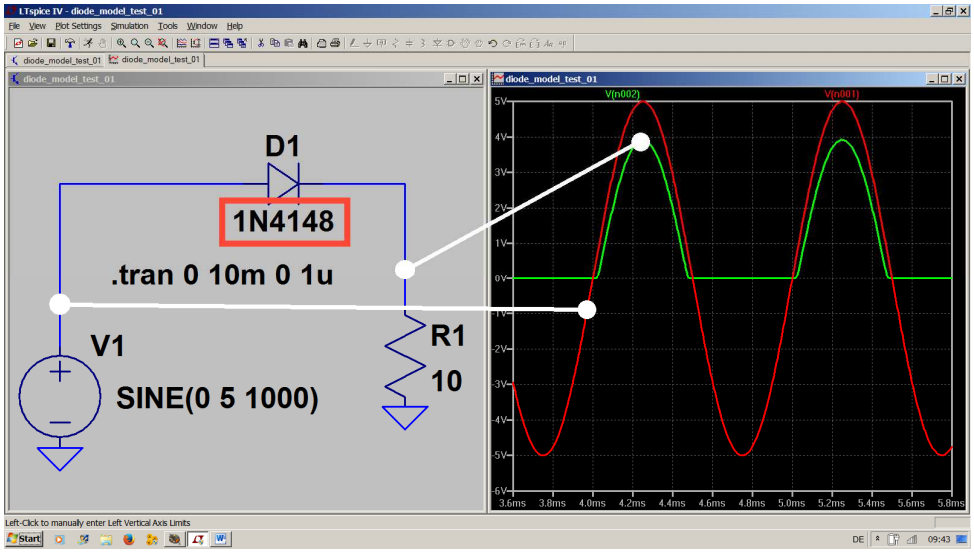
Bitte mal im simulierten Spektrum überprüfen…und da finden wir fast genau diesen Wert….Siehe da, das war doch nicht schwierig, oder?

7. Einbindung fremder Bauteil-Modelle (=Third-party Models)
7.1. Verwendung anderer Dioden

7.1.1. Was bietet LTSpice selbst als Vorrat?



Dazu nehmen wir diese einfache Testschaltung mit nur einer Diode. Dieses Bauteil findet sich in der Menüleiste als Button mit einem Diodensymbol. Jetzt speist ein 1kHz-Sinus-Signal (Spitzenwert = 5V) einen Lastwiderstand von 10Ω über eine ideale Diode.



Möchte man einen anderen Diodentyp verwenden, so erreicht man die Vorratsliste über einen **rechten Mausklick auf das Diodensymbol in der Schaltung, gefolgt von „Pick new diode“**.

Wählt man die 1N4148 als Standard-Kleinleistungstyp und klickt auf die entsprechende Zeile in der Liste, dann wird dieser Wunsch direkt in das Diodensymbol übernommen.

7.1.2. Beschaffung eines Diodenmodells aus dem Internet

Für den in der Schaltung auftretenden Spitzenstrom (ca. 0,5A) ist die 1N4148 etwas schwach auf der Brust und man sollte auf eine Leistungsdiode aus der 1N4000-Serie -- etwa die 1N4004 -- ausweichen. Also startet man die Suchmaschine und gibt z. B. ein:

1N4004 spice model

Wir öffnen eine passende Fundstelle (hier: **www.onsemi.com/pub/Collateral/1N4004RL.LIB**), markieren darin nur den Text für das SPICE-Modell und fügen ihn in die leere Seite eines Text-Editors ein. So sollte das dann aussehen:

```
*************************************
*      Model Generated by MODPEX     *
*Copyright(c) Symmetry Design Systems*
*         All Rights Reserved        *
*    UNPUBLISHED LICENSED SOFTWARE   *
*   Contains Proprietary Information *
*      Which is The Property of      *
*     SYMMETRY OR ITS LICENSORS      *
*    Modeling services provided by   *
* Interface Technologies www.i-t.com *
*************************************
.MODEL D1n4004rl d
+IS=5.31656e-08 RS=0.0392384 N=2 EG=0.6
+XTI=0.05 BV=400 IBV=5e-08 CJO=1e-11
+VJ=0.7 M=0.5 FC=0.5 TT=1e-09
+KF=0 AF=1
* Model generated on October 12, 2003
* Model format: PSpice
```

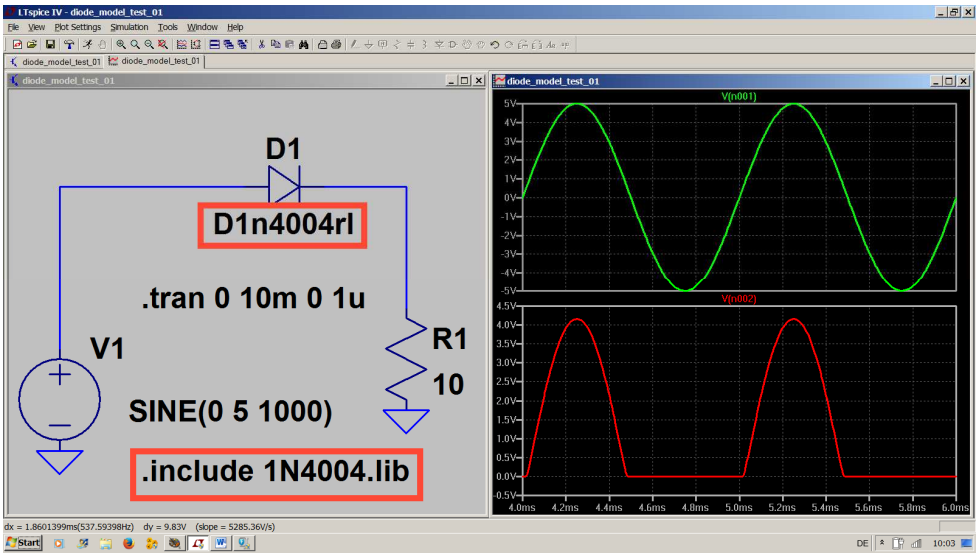
Jetzt wird dieses File in LTspice, und zwar im Ordner **LTSpice / lib / sub** als „**1N4004.lib**“ abgelegt. Da warten allerdings einige hübsche kleine Fallen auf uns:

a) Der Texteditor (z. B. Notepad) muss zuvor mit **Administrator-Rechten** geöffnet werden, sonst können wir in LTSpice nichts speichern.

b) Beim Abspeichern muss als Dateityp unbedingt „**alle Dateien**“ angegeben werden! Die **Grundeinstellung ist nämlich „Textfiles“ (*.txt)** und beim Speichern würde dann aus unserer Datei „**1N4004.lib.txt**“ gemacht. Damit kann LTSpice nichts anfangen und wir erhalten eine Fehlermeldung beim späteren Aufruf.

c) Die genaue Modellbezeichnung finden wir in der obigen Datei als **„.MODEL D1n4004rl**“. Genau diese Bezeichnung (**D1n4004rl**) müssen wir auch in LTSpice beim Aufruf verwenden, sonst…..

7.1.3. Erste Simulation mit dem neuen Modell: direkter Aufruf über seine Library



Dazu klicken wir mit der rechten Maustaste auf den Diodentyp (unterhalb des Diodensymbols) und tragen die genaue Modellbezeichnung

„D1n4004rl“

ein.

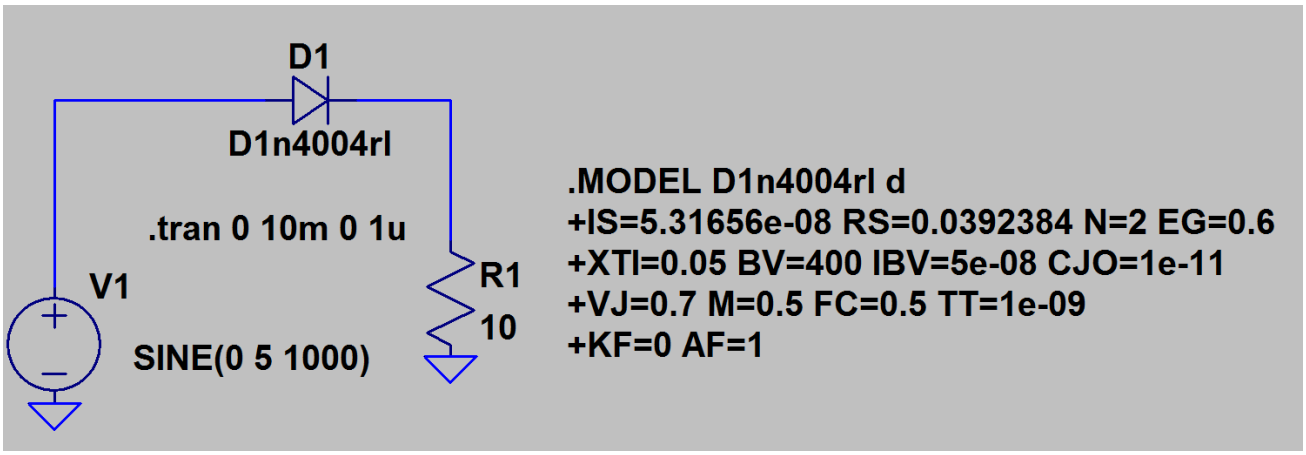
Anschließend klicken wir zusätzlich in der Menüleiste auf **„.op“**, um das passende Spice-Kommando

.include 1N4004.lib

zum Zugriff auf unser Diodenmodell zu geben.

7.1.4. Zweite Simulation mit dem neuen Modell: direkte Einbindung der Modellbeschreibung in das Projekt

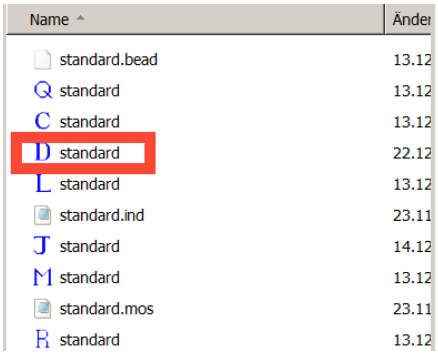
Dazu öffnen wir unseren Texteditor und laden das File „1N4004.lib“. **Darin wird der eigentliche Modelltext ausgeschnitten und in die Zwischenablage kopiert.**
Erst jetzt gehen wir wieder zu unserem LTSpice-Projekt und klicken auf **„.op“**. Der in der Zwischenablage enthaltene Modelltext wird nun eingefügt, alles mit OK bestätigt und dann sollte der Bildschirm so aussehen: /



Die Simulation ergibt dann exakt dasselbe Ergebnis.

7.1.5. Dritte Simulation mit dem neuen Modell: Übernahme in die Standard-Dioden-Bibliothek von LTSpice

Wenn wir wissen, dass wir dieses Bauteil immer wieder benötigen, dann können wir es in die **Standard-Diodenbibliothek von LTSpice einbinden**. Es steht dann -- Siehe Kapitel 7.1.1. -- nach einem Rechtsklick auf das bereits im Schaltplan-Editor abgesetzte Schaltzeichen zur Verfügung. Allerdings gehört zu dieser Prozedur etwas Aufwand….:



Schritt 1:
Wir öffnen im Datei-Explorer unser LTSpice mit **Administrator-Rechten**, gehen in den lib-Ordner und öffnen darin den „**cmp**“-Ordner.

Anschließend kommen wir mit einem Doppelklick auf „**D standard**“ an dem im Augenblick gültigen Diodenvorrat heran.

So sieht dann der Bildschirm aus:

```
* Copyright © 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Linear Technology Corporation.   All rights reserved.
*
* Mike Engelhardt
*
.model 1N914 D(Is=2.52n Rs=.568 N=1.752 Cjo=4p M=.4 tt=20n Iave=200m Vpk=75 mfg=OnSemi type=silicon)
.model 1N4148 D(Is=2.52n Rs=.568 N=1.752 Cjo=4p M=.4 tt=20n Iave=200m Vpk=75 mfg=OnSemi type=silicon)
.model MMSD4148 D(Is=2.52n Rs=.568 N=1.752 Cjo=.64p M=.4 tt=5n Iave=200m Vpk=100 mfg=Onsemi type=silicon)
.model 1N5817 D(Is=31.7u Rs=.051 N=1.373 Cjo=190p M=.3 Eg=.69 Xti=2 Iave=1 Vpk=20 mfg=OnSemi type=Schottky)
```

Schritt 2:
Jetzt nehmen wir wieder unseren Texteditor (z. B. Notepad) und öffnen mit Administratorrechten die in Kapitel 7.1.2. von uns imPfad „**lib / sub**“ gespeicherte Modellbeschreibung der Diode 1N4004 (….es war das File „**1N4004.lib**“):

```
*************************************
*      Model Generated by MODPEX    *
*Copyright(c) Symmetry Design Systems*
*         All Rights Reserved       *
*    UNPUBLISHED LICENSED SOFTWARE  *
*   Contains Proprietary Information *
*      Which is The Property of     *
*     SYMMETRY OR ITS LICENSORS     *
*    Modeling services provided by  *
* Interface Technologies www.i-t.com *
*************************************
.MODEL D1n4004rl d
+IS=5.31656e-08 RS=0.0392384 N=2 EG=0.6
+XTI=0.05 BV=400 IBV=5e-08 CJO=1e-11
+VJ=0.7 M=0.5 FC=0.5 TT=1e-09
+KF=0 AF=1
* Model generated on October 12, 2003
* Model format: PSpice
```

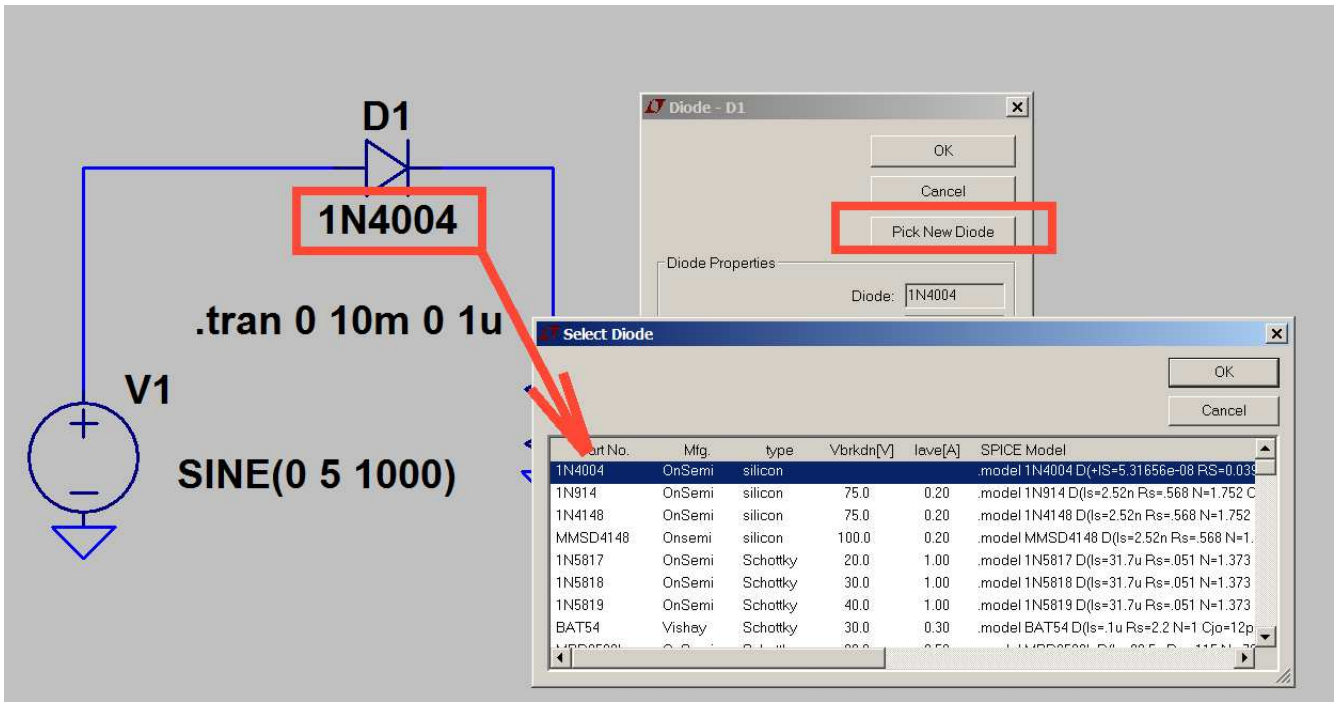
Wir benötigen nur den eigentlichen, markierten Modelltext, passen ihn aber zuerst an die Form an, die in der Standard-Diodenbibliothek üblich ist. Dazu sind folgende Änderungen erforderlich -- sie sind in roter Farbe markiert:

```
.model 1N4004
D(
+IS=5.31656e-08 RS=0.0392384 N=2 EG=0.6
+XTI=0.05 BV=400 IBV=5e-08 CJO=1e-11
+VJ=0.7 M=0.5 FC=0.5 TT=1e-09
+KF=0 AF=1
mfg=OnSemi
type=silicon
)
```

Schritt 3:
Dieses Modell-File kopiert man nun in die erste Zeile der Dioden-Standardbibliothek, bringt die gesamte Information in einer einzigen Zeile unter und kann dann die geänderte Bibliothek wieder speichern. Sie sollte folgende Form haben und mit der 1N4004 beginnen:

```
* Copyright © 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Linear Technology Corpora
*
* Mike Engelhardt
*

.model 1N4004 D(+IS=5.31656e-08 RS=0.0392384 N=2 EG=0.6 +XTI=0.05 BV=400 IBV=5e-08 CJO=1e-11 +VJ=0
.model 1N914 D(Is=2.52n Rs=.568 N=1.752 Cjo=4p M=.4 tt=20n Iave=200m Vpk=75 mfg=OnSemi type=silicon
.model 1N4148 D(Is=2.52n Rs=.568 N=1.752 Cjo=4p M=.4 tt=20n Iave=200m Vpk=75 mfg=OnSemi type=silic
```

Schritt 4:
Jetzt kommt der Schlusstest und den hatten wir bereits in Kapitel 7.1.1 geübt:



Damit klappt auch eine neue Simulation.

7.2. Übernahme neuer Bipolar-Transistoren

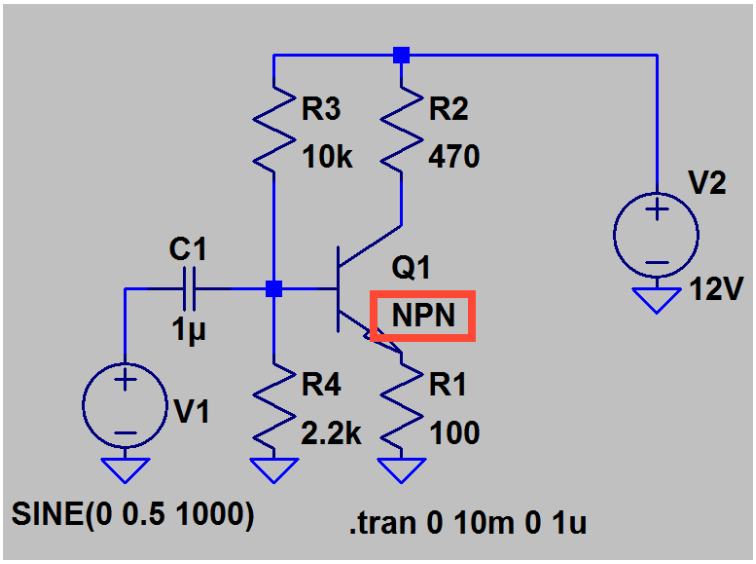
7.2.1. Ersatz des Standard-Typs durch ein Exemplar aus dem Vorrat

Im Prinzip funktioniert das ähnlich einfach wie bei den Dioden (Siehe Kapitel 7.1.1.), aber:

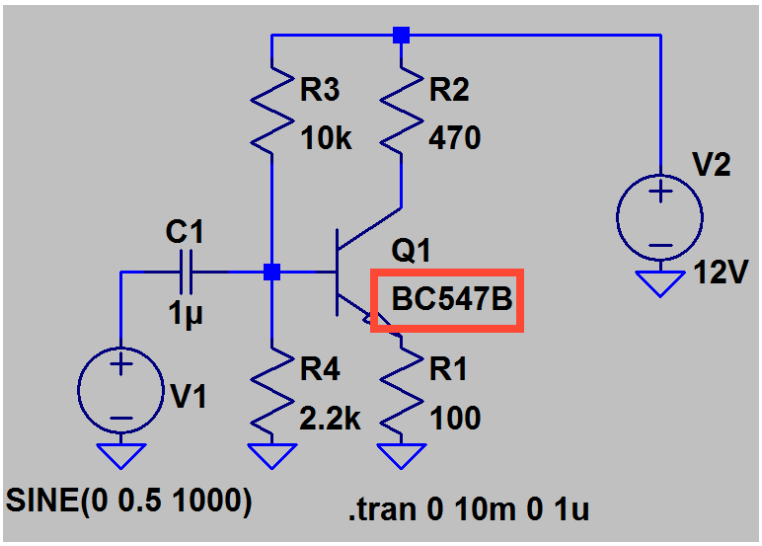
a) Wir müssen **npn- und pnp-Typen** unterscheiden und

b) im Teile-Vorrat in der Menüleiste (…= der Button mit dem UND-Symbol drauf…) werden uns z. B. gleich 4 unterschiedliche Symbole für den npn-Transistor angeboten. Drei davon sind nur leicht verschiedene Symboldarstellungen , aber bei der vierten Version kann auch noch die durch die Transistorherstellung auftretende „Bulk"-Diode zwischen Substrat und Gehäuse einbezogen werden. Ist aber eher etwas für Spezialisten oder HF-Anwendungen und deshalb kann man sich meist mit der Variante „**npn**" begnügen.

Klickt man dann „mit rechts" z. B. auf ein solches npn-Symbol in der eigenen Schaltung, dann werden in gewohnter Form alle zur Verfügung stehenden npn-Transistoren aufgelistet und es reicht wieder ein Mausklick zur Umstellung.



So würde ein einfacher Verstärker in Emitterschaltung unter Verwendung des Symbols „npn" aussehen….



…und hier wurde der npn-Transistor in bekannter Weise (= Rechtsklick auf das Schaltzeichen dann „pick new transistor") durch den Typ „BC547B" aus dem LTSpice-Vorrat ersetzt.

7.2.2. Beschaffung eines Transistormodells aus dem Internet

Das ist öfters nötig als man denkt, denn in der Vorratsbibliothek finden sich z. B. keine modernen HF- oder Mikrowellen-Transistoren. Auch benötigt man bisweilen eine sehr rauscharme Version und da muss die LTSpice-Bibliothek passen.

Nehmen wir mal an, dass wir auf den rauscharmen Typ „BC849“ angewiesen sind. Die Internet-Recherche bietet dann eine passende Infineon-Bibliothek unter der Adresse

https://www.infineon.com/dgdl/S_AFBJT%5B1%5D.lib?fileId=db3a3043156fd5730115a3f6c81509a8

an und daraus holen wir (wie bereits geübt) das Modell für den BC849. Es sieht so aus:

```
.MODEL BC849 NPN (IS=45.000F NF=1.010 BF=516.544 VAF=74.000 IKF=0.708
+ ISE=55.668F NE=2.567 NR=1.015 BR=7.745 VAR=14.000 IKR=1.000
+ ISC=1.084P NC=4.063
+ RB=9.000 IRB=0.100M RBM=4.500
+ RE=0.350 RC=1.445
+ CJE=13.050P VJE=0.690 MJE=0.375 FC=0.750
+ CJC=4.100P VJC=0.750 MJC=0.420 XCJC=0.650
+ TF=0.620N TR=2.5N PTF=1.000
+ XTF=68.000 VTF=1.000 ITF=0.720
+ XTB=1.400 EG=1.110 XTI=3.200
+ KF=10.000F AF=1.000)
```

Das wird als „**BF849.lib**“ in der LTSpice-Bibliothek im Ordner „**sub**“ gespeichert und dann haben wir - wie bei unserer Diode im letzten Kapitel -- wieder die drei bekannten Möglichkeiten:

a) **Erste Simulation** mit dem neuen Modell: **direkter Aufruf über seine Library** (Siehe Kapitel 7.1.3)

b) **Zweite Simulation** mit dem neuen Modell: **direkte Einbindung der Modellbeschreibung** in das Projekt auf dem Bildschirm (Siehe Kapitel 7.1.4)

c) **Dritte Simulation** mit dem neuen Modell: **Übernahme in die Standard-npn-Transistor-Bibliothek von LTSpice** (Siehe Kapitel 7.1.5). Aufruf durch rechten Mausklick auf das Schaltzeichen und „**pick new transistor**“.

7.3. Richtiger Umgang mit Sub-Circuits (*.sub)

Das ist ein alltägliches Problem, denn etwas aufwendigere Schaltungen (Beispiel: Operationsverstärker) enthalten oft viele Bauteile oder Baugruppen. Hier hilft man sich damit, einzelne Teile innerhalb dieses Modells unter einem eigenen Namen als Modelle oder auch als „Subcircuits“ zusammen zu fassen. Das Gesamtmodell ruft dann in seiner Modellbeschreibung die beteiligten Gruppen auf und so entsteht ein besser überschaubares Gesamtgebilde.

Es beginnt immer mit **„*subckt <name>“**“und hört mit **„.ends“** auf.

Das komplette Bauteil wird jedoch unbedingt als ***.sub** in der „sub“-Library von LTSpice gespeichert. Von dort kann es (u. U. auch aus einer dort extra angelegten *.lib- Modellsammlung) geholt werden!

7.3.1. Beispiel: Verwendung des Operationsverstärkers TL072

7.3.1.1. Beschaffung des Spice Modells aus dem Internet

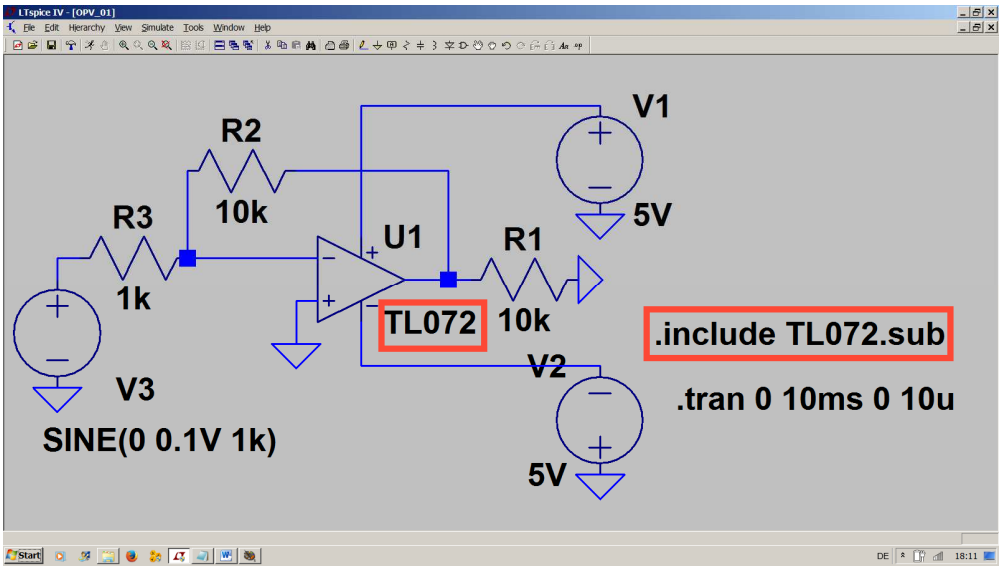
Das ist eine einfache Übung, denn dazu reicht der Aufruf „**TL072 spice model**“ bei der Suchmaschine. Wie immer, muss man einen Texteditor mit Administratorberechtigung öffnen und dann das eigentliche Modell aus der gewählten html-Seite herausholen und in eine neue Editorseite hineinkopieren. Gespeichert wird das Ergebnis als „**TL072.sub**“ im Bibliothekspfad von LTSpice (= LTC / LTSpiceIV / lib / sub). **Bitte wie üblich den Editor vorher auf „all Files“ schalten, um das Speichern als „*.sub.txt“ – File zu verhindern….**
So sollte das File aussehen:

```
* TL072 OPERATIONAL AMPLIFIER "MACROMODEL" SUBCIRCUIT
* CREATED USING PARTS RELEASE 4.01 ON 06/16/89 AT 13:08
* (REV N/A)      SUPPLY VOLTAGE: +/-15V
* CONNECTIONS:   NON-INVERTING INPUT
*               | INVERTING INPUT
*               | | POSITIVE POWER SUPPLY
*               | | | NEGATIVE POWER SUPPLY
*               | | | | OUTPUT
*               | | | | |
.SUBCKT TL072    1 2 3 4 5
*
  C1   11 12 3.498E-12
  C2    6  7 15.00E-12
  DC    5 53 DX
  DE   54  5 DX
  DLP  90 91 DX
  DLN  92 90 DX
  DP    4  3 DX
  EGND 99  0 POLY(2) (3,0) (4,0) 0 .5 .5
  FB    7 99 POLY(5) VB VC VE VLP VLN 0 4.715E6 -5E6 5E6 5E6 -5E6
  GA    6  0 11 12 282.8E-6
  GCM   0  6 10 99 8.942E-9
  ISS   3 10 DC 195.0E-6
  HLIM 90  0 VLIM 1K
  J1   11  2 10 JX
  J2   12  1 10 JX
  R2    6  9 100.0E3
  RD1   4 11 3.536E3
  RD2   4 12 3.536E3
  RO1   8  5 150
  RO2   7 99 150
  RP    3  4 2.143E3
  RSS  10 99 1.026E6
  VB    9  0 DC 0
```

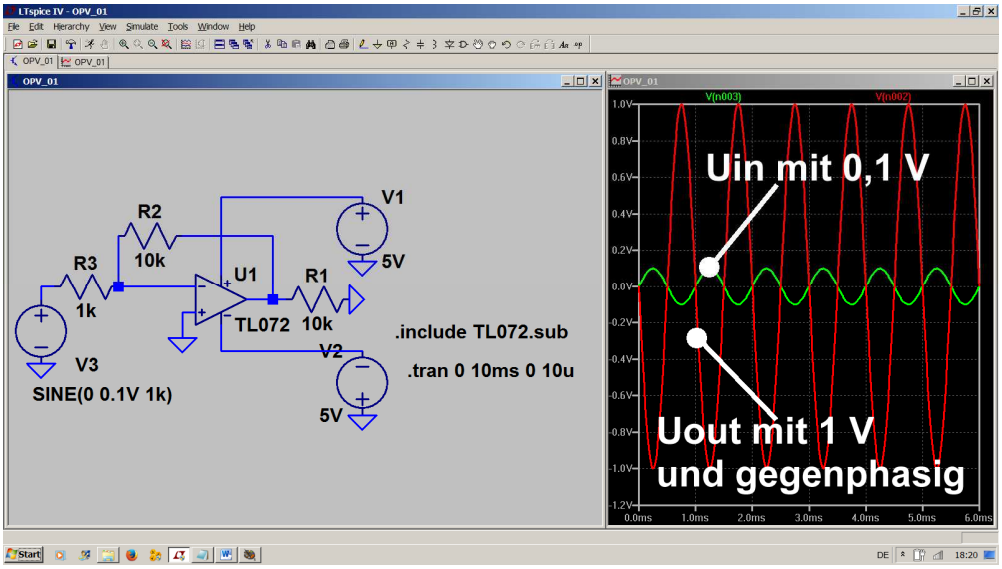
```
 VC    3 53 DC 2.200
 VE   54  4 DC 2.200
 VLIM  7  8 DC 0
 VLP  91  0 DC 25
 VLN   0 92 DC 25
.MODEL DX D(IS=800.0E-18)
.MODEL JX PJF(IS=15.00E-12 BETA=270.1E-6 VTO=-1)
.ENDS
```

7.3.1.2. Simulation einer einfachen Verstärkerschaltung mit Modellaufruf über „include“

Es handelt sich um eine invertierende Schaltung mit zehnfacher Verstärkung, einem Eingangswiderstand von 1k und zwei Versorgungsspannungen (+5V und -5V).
Benützt wird das „**opamp2**“-Schaltzeichen aus dem Bauteilvorrat und darin wird gleich der neue Name „**TL072**“ eingetragen.
Dann schreibt man noch die Spice-Direktive **„include TL072.sub“**, programmiert die Eingangs-Spannungsquelle (Sinus mit 0,1 V / 1 kHz), legt noch die Simulationsvorgaben im Zeitbereich (= Transient Simulation für 10 ms mit Timestep = 10 µs) fest und dann kann es losgehen.

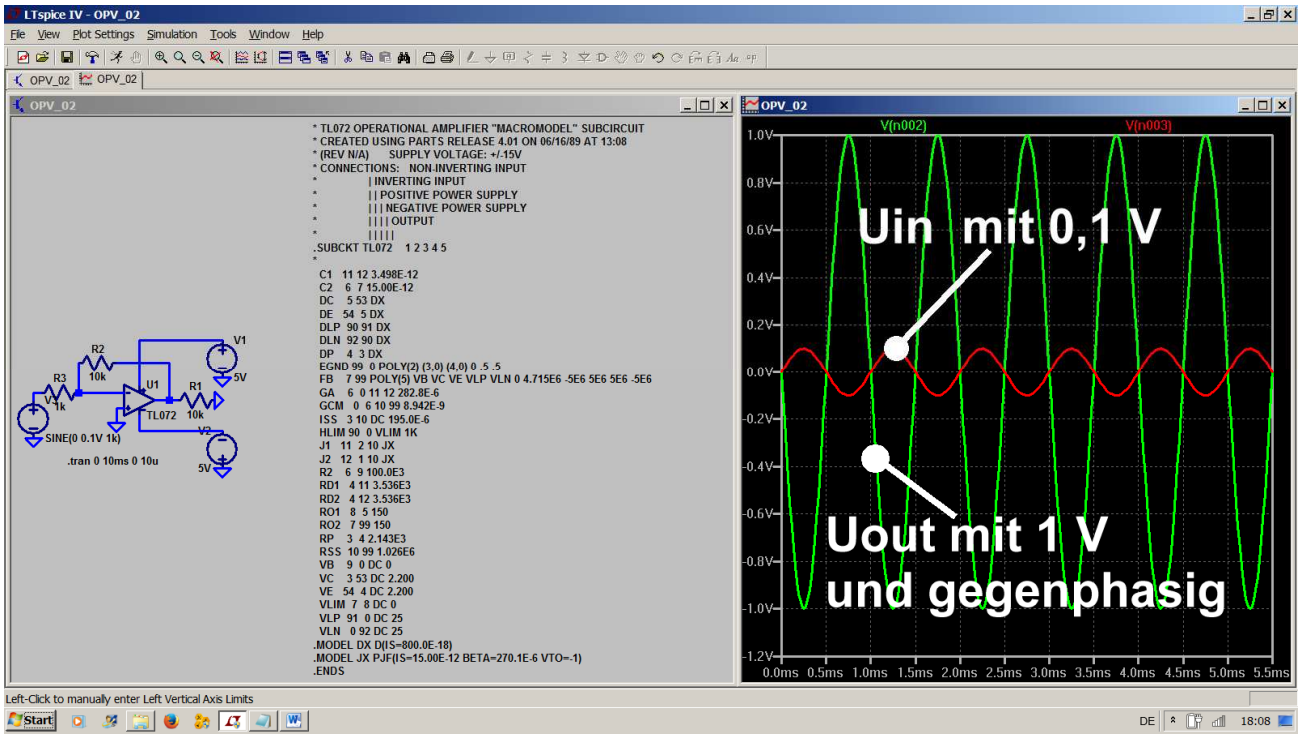


So sollte das Ergebnis aussehen:



7.3.1.3. Simulation mit Modellbeschreibung im Editor-Schaltplan

Das kennen wir ja nun schon zum Genüge und das macht auch keine Schwierigkeiten:



7.3.2. Die Krönung: Erstellung eines kompletten neuen Symbols mit *.subckt als Modellbeschreibung

Dazu wollen wir uns mal einen rauscharmen Dual-Gate-MOSFET (Typ: BF998) für eine Empfänger-Eingangsstufe vornehmen, der garantiert nicht in LTSpice enthalten ist.

Schritt 1:

```
*   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!

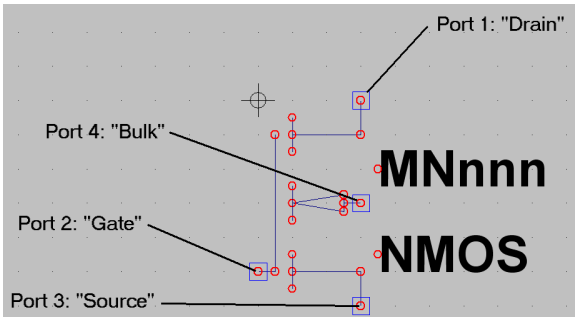
Das geht natürlich mit der Beschaffung dieses Modells aus dem Internet los (…Eingabe in die Suchmaschine: „**bf998 spice model**“) und das Ergebnis zeigt dieses Bild. Es sieht zwar schön aus und die Anschlüsse sind sauber durchnummeriert, aber darin gibt es **eine ganz bösartige Tücke:**

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.sub**“ im LTSpice-Bibliotheksordner „**lib / sub**“ gespeichert!

Schritt 2:

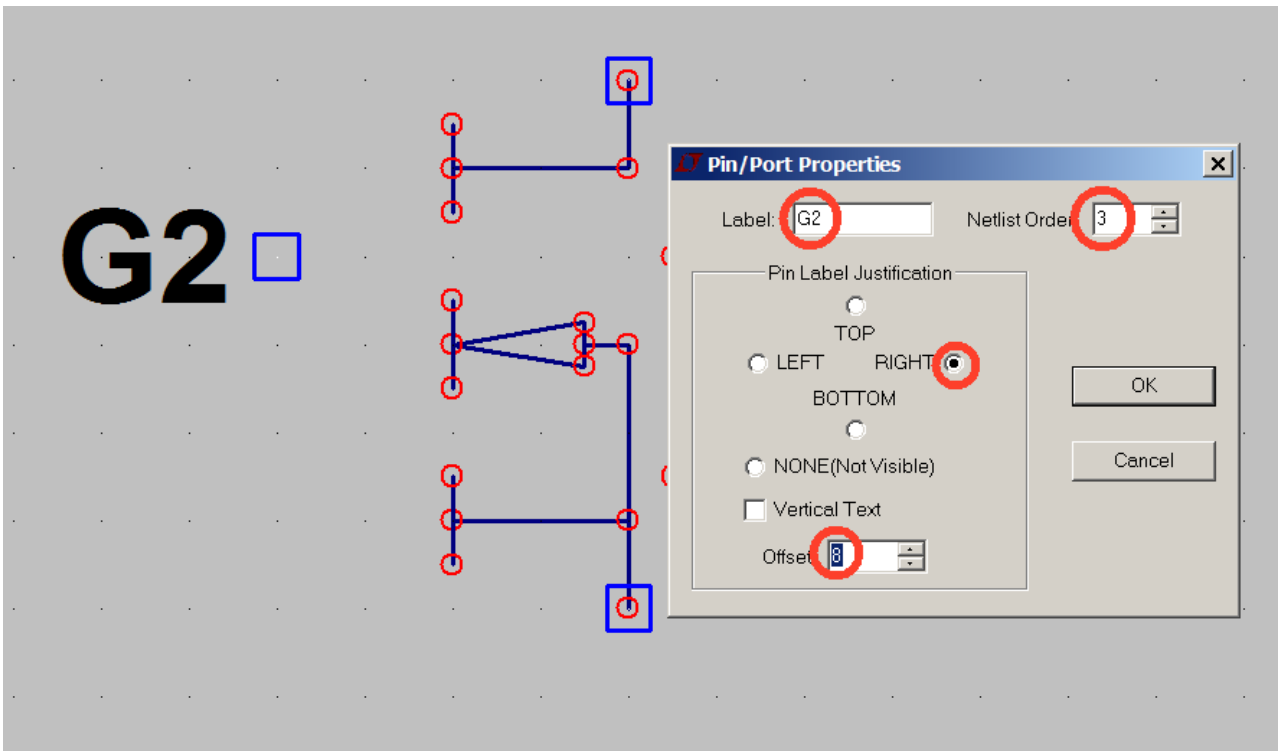
Nun wollen wir zuerst das Symbol für einen Dualgate-MOSFET erstellen, gehen dazu in das Menü „File“ und klicken auf „**New Symbol**“. Die zum Zeichnen erforderlichen Werkzeuge (Linie, Kreis, Rechteck etc.) gibt es unter „**Draw**“ und über dem Bildschirm liegt ein Punkraster zur Erleichterung der Zeichenarbeit.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



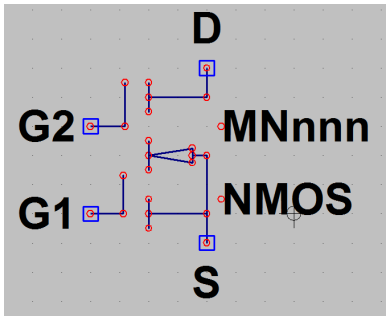
Dann hat man die Wahl: **entweder** löscht man dieses Symbol ganz vom Schirm (= erst die Taste F5 drücken, auf das Symbol mit der linken Maustaste klicken, schließlich mit „Escape“ wieder zurück) und zeichnet den Dualgate-MOSFET komplett neu.
Oder man ändert das eben geladene Symbol (nmos4.asy) für die eigenen Bedürfnisse ab und speichert es anschließend als „**BF998.asy“ im LTSpice-Bibliotheksordner „lib / cmp“**

Schritt 3:
Wir wählen den leichteren zweiten Weg und ändern einfach das Schaltzeichen. Dazu benötigen wir die Löschfunktion für nicht benötigte Linien (= F5) sowie das DRAW-Menü, hinter dem sich die Zeichenmöglichkeiten für Linien, Kreis etc. verbergen.
Beginnen wir mit dem Löschen des kompletten Gates samt Anschluss, verbinden aber dafür den Source-anschluss mit dem Substrat (= Bulk) durch eine neue Linie. Dabei darf man nicht vergessen, den Bulk-Anschluss (= kleines Quadrat) ebenfalls zu löschen!
Die kleinen roten Kreise markieren dabei die Enden von Linien und sind sowohl Zeichenhilfen wie auch später Fangpunkte beim Einsatz des Bauteils.

Schritt 4:
Die eigentlichen Bauteilanschlüsse (korrekte Bezeichnung: „Ports“ = kleine Rechtecke) 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 (hier: G2), die Port / Pin – Nummer laut SPICE-Modell (hier: 3) und die Position der Anschlussbezeichnung neben dem Symbol (= Left / Offset = 8). Dabei gilt beim BF998: **Source = Port 1 / Drain = Port 2 / Gate 2 = Port 3 / Gate 1 = Port 4**.

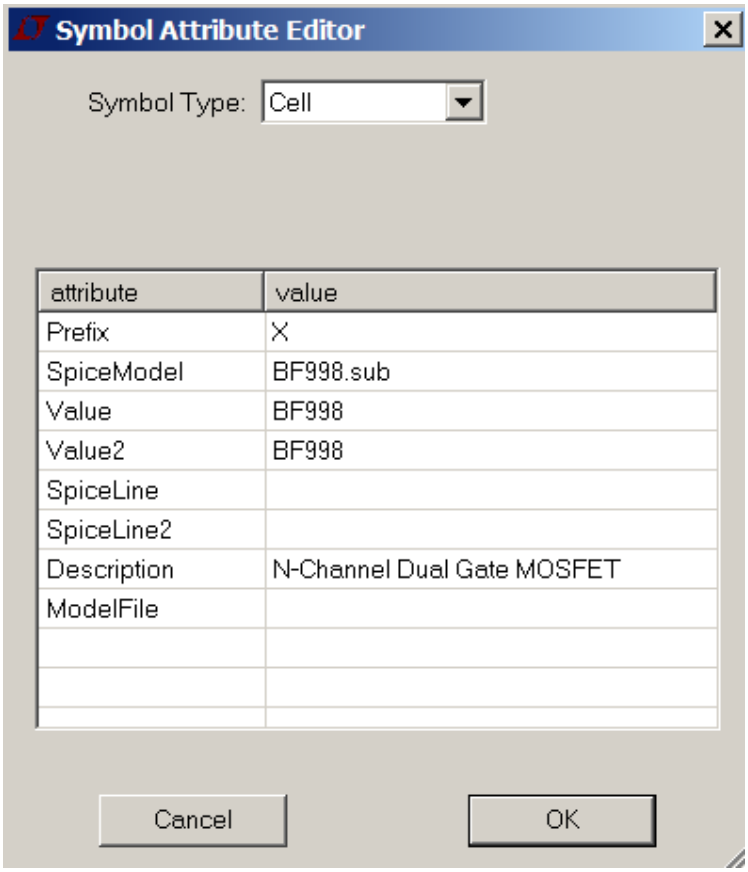


Nach dem Absetzen kann dieser Anschluss für Gate 2 (= Netlist Order = 3) noch mit F7 verschoben und an die richtige Stelle gebracht werden. **Und diese Prozedur wird nochmals für Gate 1 (Netlist Order = 4) wiederholt.**



Schritt 5:
Jetzt zeichnet man noch mit „DRAW / Lines“ die nötigen Gate-Anschlüsse und kontrolliert anschließend bei ALLEN Anschlüssen die korrekte Nummerierung („Netlist Order“), nachdem man jeweils mit der rechten Maustaste auf den entsprechenden Anschluss geklickt hat:
Source = 1
Drain = 2
Gate 2 = 3
Gate 1 = 4

Übrigens: der Offset wird bei allen Anschlüssen auf „8“ eingestellt!



Schritt 6:
Jetzt wird das fertig gestellte Symbol erst mal im Bibliothekspfad

„SpiceIV / lib / sym“

als
„**dgate_MOSFET.asy**“

für weitere spätere Aktionen gespeichert.

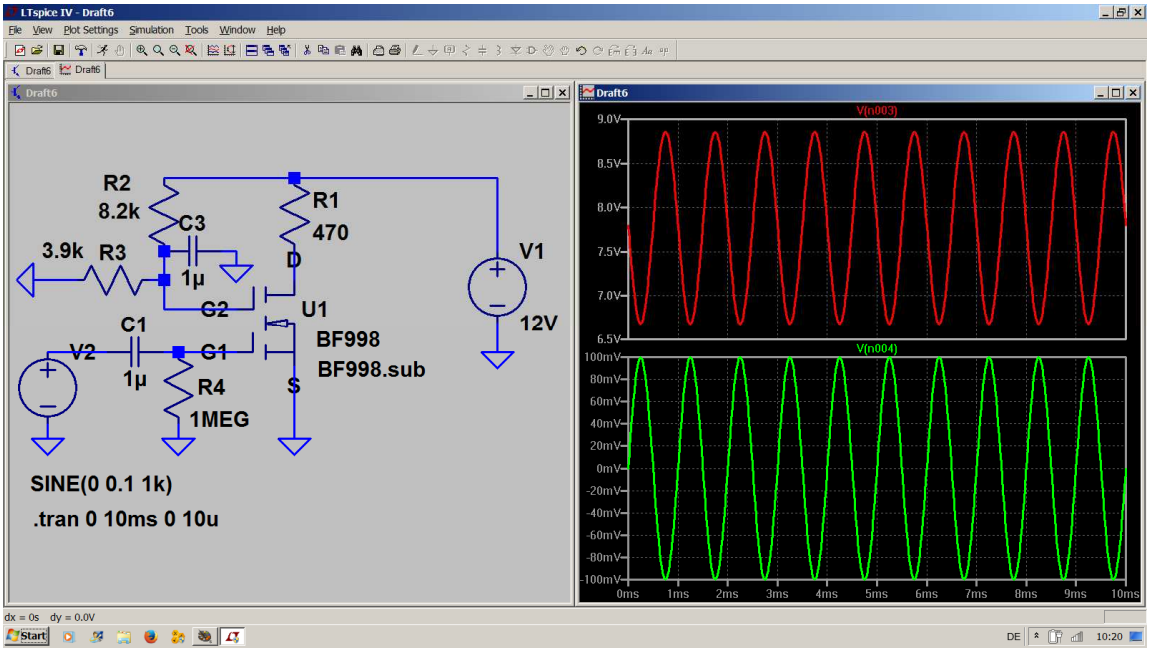
Erst jetzt machen wir einen echten BF998 daraus, indem wir unter „Edit“ das Menü „**Attributes / Edit Attributes**“ öffnen und diese Liste (Zeile für Zeile) korrekt ausfüllen.

Aber dann haben wir es fast schon geschafft.

Schritt 7:
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.sub**“ sichtbar wird. Nun kann man sich im „**lib / sub**“ – Ordner einen weiteren Ordner mit dem Namen Dualgate-MOSFET anlegen und endlich das gesamte neue Teil als „**BF998.asy**“ darin speichern.
(Wer möchte, kann auch auf den neuen zusätzlichen Ordner verzichten und einfach nur in „lib / sub“ speichern).

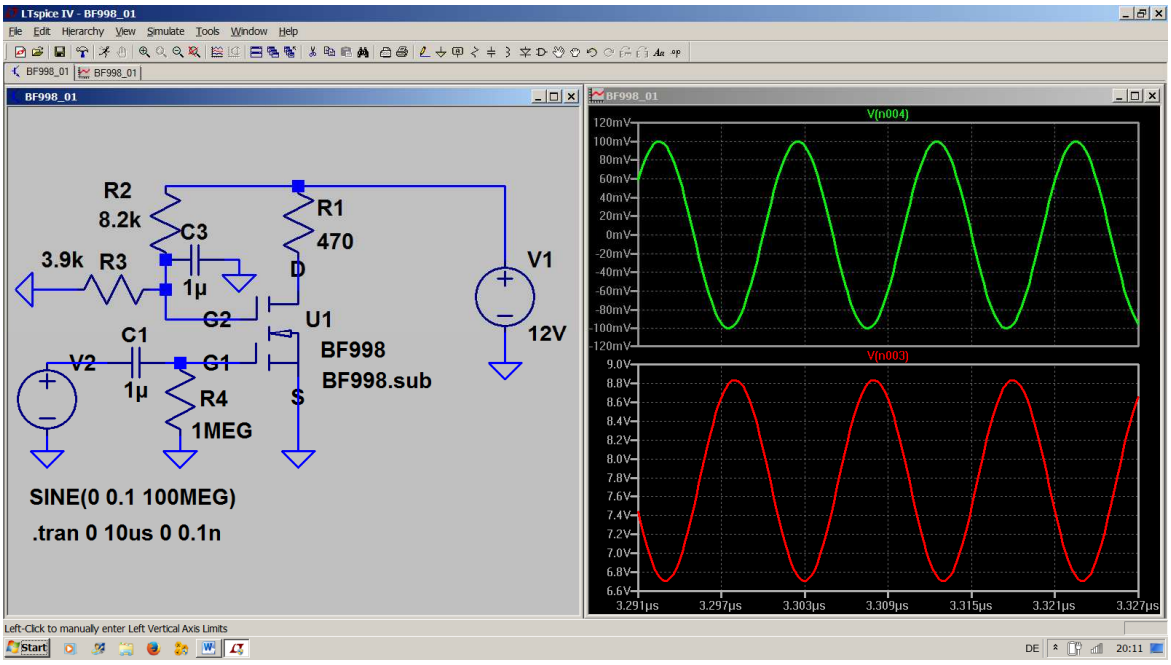
Fertig!

Wir wollen jetzt noch prüfen, ob das auch wirklich funktioniert. Dazu erstellen wir uns eine einstufige Verstärkerschaltung, bei der Gate 2 eine Vorspannung von +4 V, Gate 1 jedoch von Null Volt erhält. Die Betriebsspannung sei +12 V.
Gate 1 wird mit einer 1 kHz-Sinusspannung (Spitzenwert = 100 mV) angesteuert.



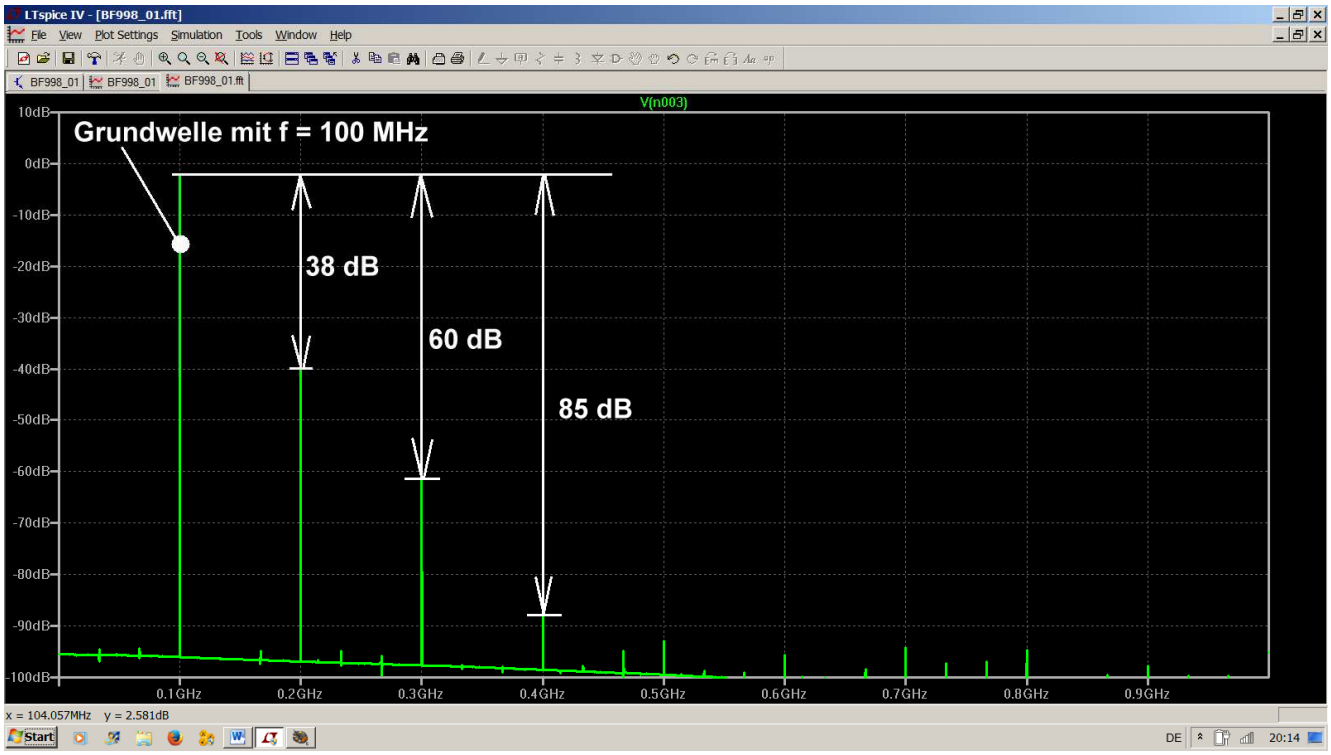
Das Ausgangssignal am Drain-Anschluss ist exakt gegenphasig und die Verstärkung beträgt etwa V = 15fach. Klappt also!

Nun wollen wir mal sehen, was die Schaltung kann und erhöhen die Frequenz auf 100MHz (…bitte natürlich auch die Simulationszeit auf 10µs und den Timestep auf 0,1ns reduzieren….):



Man sieht, dass die Verstärkung fast gleich geblieben ist (= ca. 14…15fach). Aber das Ausgangssignal wird durch die Laufzeiten im MOSFET etwas verzögert (…gut zu sehen, wenn man den Höchstwert beim Eingangssignal mit dem Tiefstwert des Ausgangssignals vergleicht).

Nun sehen wir uns mal die Qualität des Ausgangssignals an, unterziehen es einer FFT mit 262 144 Samples und messen die Dämpfungen der erzeugten Oberwellen gegenüber der Grundwelle im Spektrum. Der interessierende Frequenzbereich sei 0….1GHz.



Na also!
Wenn man sich zusätzlich daran erinnert, dass

40 dB Dämpfung einem Klirrfaktor von 1 %,

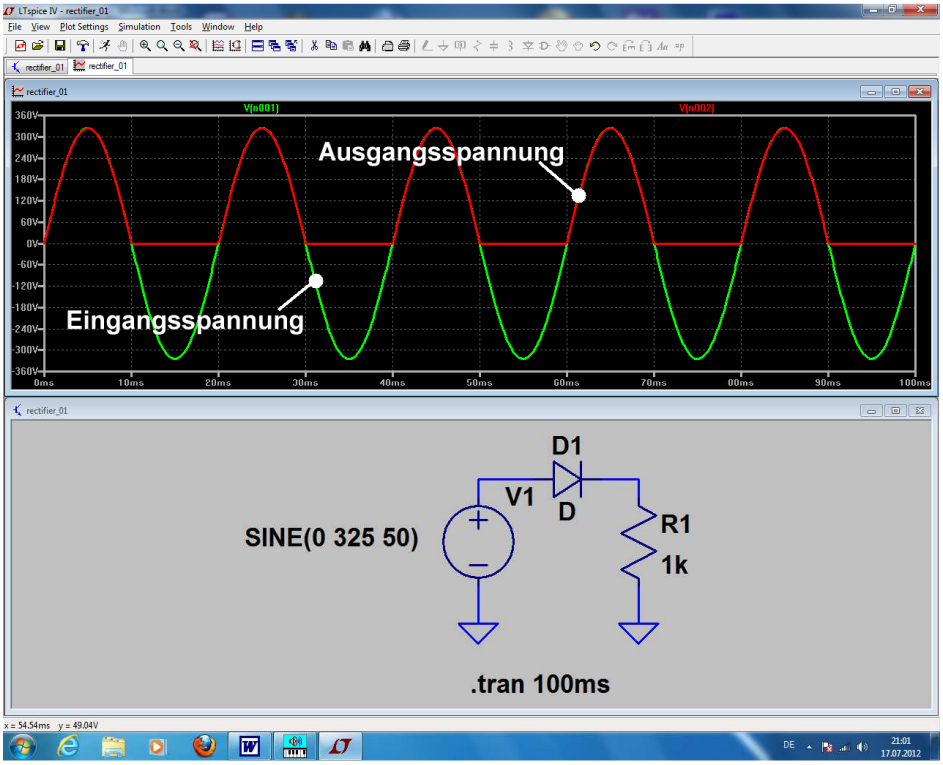
60 dB Dämpfung dagegen einem Klirrfaktor von 0,1 % = 1 Promille,

80 dB Dämpfung sogar einem Klirrfaktor von 0,01 % = 0,1 Promille

entsprechen…

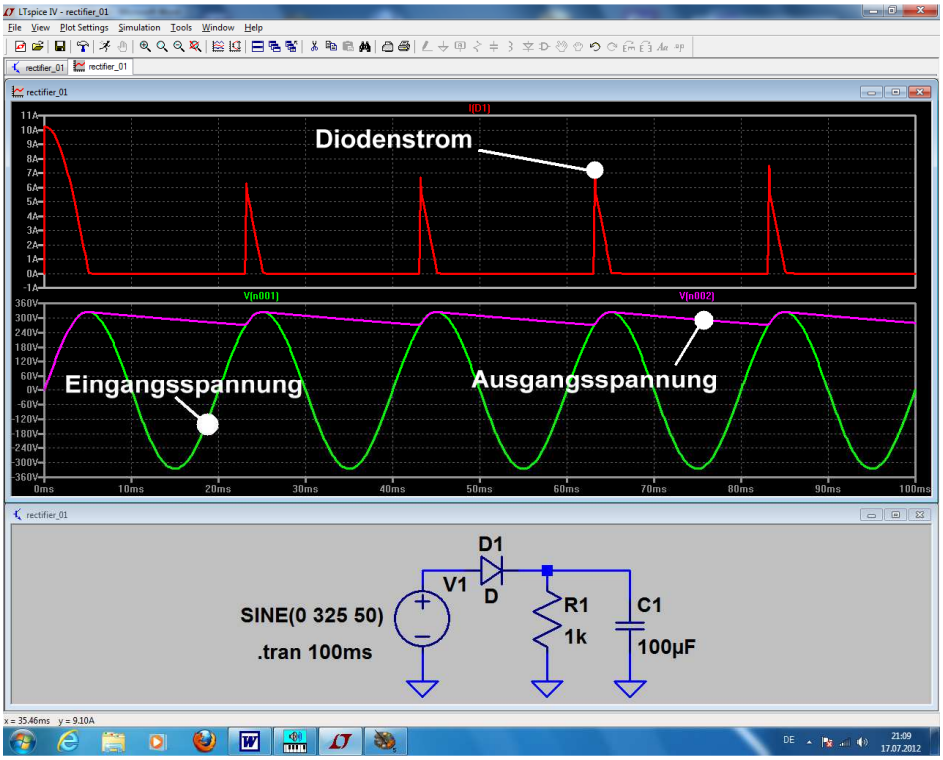
8. Zweites Projekt: Gleichrichtung

8.1. Einpuls-Gleichrichter ohne Trafo



Das darf nun kein Problem mehr sein: erst die Schaltung aus Spannungsquelle, Diode und Widerstand (1k) zusammenstellen, dann eine sinusförmige Eingangsspannung (50 Hz / Spitzenwert = 325 V) programmieren und eine Simulationszeit von 0 bis 100 Millisekunden vorsehen. Nach der Simulation sollte dieser Bildschirm zu sehen sein.

Aufgabe: Schalten Sie am Ausgang noch einen Ladekondensator von 100 Mikrofarad parallel und simulieren Sie zusätzlich den Strom in der Diode (= fahren Sie dazu mit dem Cursor auf den linken Anschluss der Diode, bis sich der Cursor zur Stromzange verändert). Stellen Sie den Strom in einem eigenen Diagramm dar.

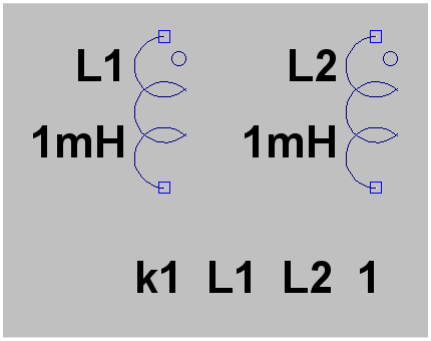


8.2. Einpuls-Gleichrichter mit Transformator

8.2.1. Ein idealer Trafo muss her!

Leider enthält die LTSpice-Bibliothek keine speziellen Transformatormodelle, sondern nur die aus dem originalen SPICE-Manual stammende, einfache (= aber deshalb sehr gut verständliche) Lösung für **den idealen Transformator**. Sie sieht so aus:

Man trägt die vorgesehene Anzahl an Wicklungen mit ihren Induktivitätswerten in die Schaltung ein und schreibt zusätzliche Spice-Direktiven für die magnetischen Kopplungsfaktoren „kn“ zwischen den einzelnen Wicklungen.



Beispiel für einen idealen Übertrager mit 2 Wicklungen und dem Übersetzungsverhältnis 1:1

a) Als Schaltzeichen wird „**ind2**“ aus der Symbolbibliothek geholt und platziert. Es enthält außer der Induktivität (= Wicklung) noch die bekannte und bei Trafos übliche **Kennzeichnung des Wicklungsanfangs.**

b) Die Zeile „***k1 L1 L2 1***“ sagt: „**die magnetische Kopplung zwischen der Wicklung L1 und der Wicklung L2 beträgt „1“ = „100%“.**

c) **Achtung:**
Eine direkte Eingabe des Übersetzungsverhältnisses ist bei SPICE nicht möglich, da immer nur mit den Bauteil-Eigenschaften (hier: Induktivitätswerte und magnetische Kopplung) simuliert wird. Also müssen wir das anders lösen und gehen auf die Induktivitäten los.
Dabei gilt folgendes:

$$\frac{L_{Primär}}{L_{Sekundär}} = (\ddot{u})^2 = \left(\frac{U_{Primär}}{U_{Sekundär}}\right)^2$$

und diese Formel stellen wir (wenn wir L1 vorgegeben haben und das Übersetzungsverhältnis kennen) einfach nach $L_{Sekundär}$ um. Ist also sehr einfach!

Die Frage ist nun: wie geht man in der Praxis vor, wenn die Eigenheiten eines echten gewickelten Übertragers berücksichtigt werden müssen? Das ist z. B. bei Schaltnetzteilen unerhört wichtig, denn dort wird sonst der Unterschied zwischen Simulation und wirklicher Schaltung so groß, dass die Simulation keinen Sinn mehr macht.

Das folgt im Kapitel
(Da soll gezeigt werden, wie man sich hilft, wenn man einen **echten, gewickelten Übertrager** vor sich hat und das ist z. B. bei Schaltnetzteilen unerhört wichtig. Dort wird nämlich sonst der Unterschied zwischen Simulation und wirklicher Schaltung so groß, dass die Simulation keinen Sinn mehr macht. Die Ausgangsbasis war das SPICE-Modell für den Ausgangsübertrager eines röhrenbestückten HiFi-Verstärkers. Es wurde im Internet entdeckt und entsprechend umfunktioniert).

8.2.2. Einpuls-Gleichrichter mit idealem Trafo

Aufgabe:

Untersuchen Sie ein Netzteil mit idealem Netztransformator, Gleichrichterdiode, Lastwiderstand RL = 1kΩ und Ladekondensator C = 100µF. Der Trafo-Eingang liegt an der Netzspannung mit 230V / 50 Hz, das Übersetzungsverhältnis sei 20:1. Simulieren Sie die Zeit von Null bis 100 Millisekunden.

Lösung:

Die meisten Bauteile (Spannungsquelle, Kondensator, Widerstand, Diode, Ground-Symbol) kennen wir ja schon vom letzten Beispiel her. Dazu kommt nun der ideale Übertrager aus dem vorigen Kapitel. Bitte gleich alles platzieren sowie die Werte der Bauteile, der Spannungsquelle und die Simulationsanweisung eintragen.

Jetzt müssen wir uns noch um den Transformator kümmern und die Sache mit dem Übersetzungsverhältnis unterbringen. Dabei gilt folgendes:

$$\frac{L_{Primär}}{L_{Sekundär}} = (ü)^2$$

Wenn wir (wie in den meisten Fällen für einen fast idealen Trafo zulässig) mal von einer **Primär-Induktivität mit 1 Henry** ausgehen, dann müssen wir eine Sekundär-Induktivität von

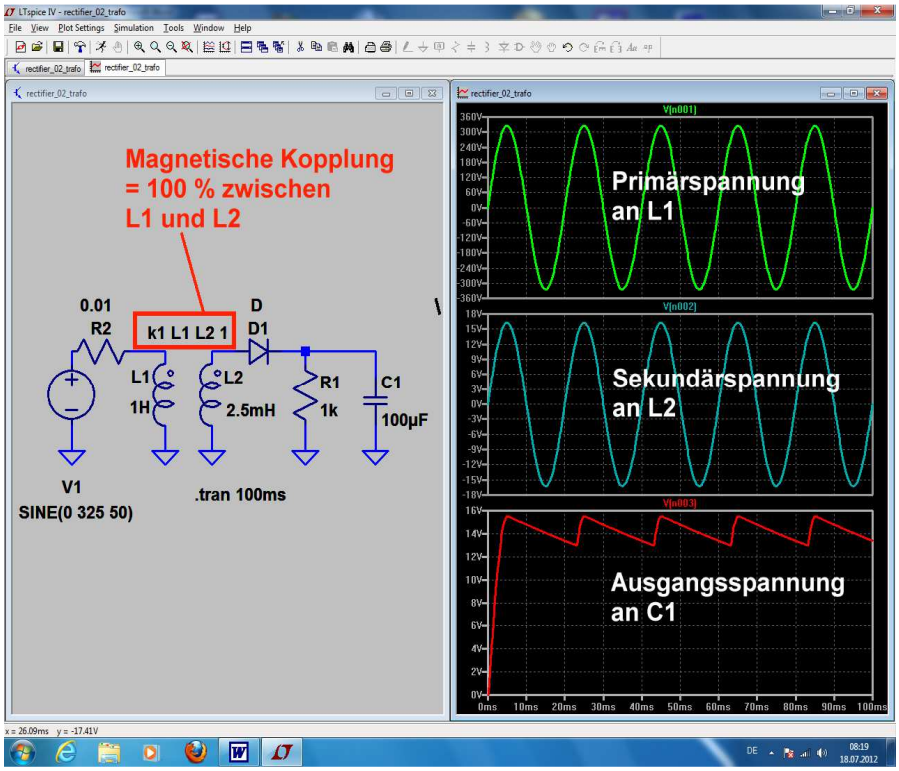
$$L_{Sekundär} = \frac{L_{Primär}}{(20)^2} = \frac{1H}{400} = 2{,}5mH$$

benützen. Für einen idealen Trafo kann man dann die Wicklungs-Widerstände, die Streu-Induktivitäten und die Wicklungs-Kapazitäten weglassen.

Aber Vorsicht: der Wicklungswiderstand darf NIEMALS = Null sein!

Einen fehlenden Wirkwiderstand in Reihe zu einer Induktivität beantwortet das originale SPICE sofort mit einem Abbruch und einer Fehlermeldung!

(Grund: legt man eine solche ideale Induktivität an eine Gleichspannung an, so muss diese Spannung durch Induktion erzeugt werden. Das bedeutet, dass der Strom sich wegen der Beziehung

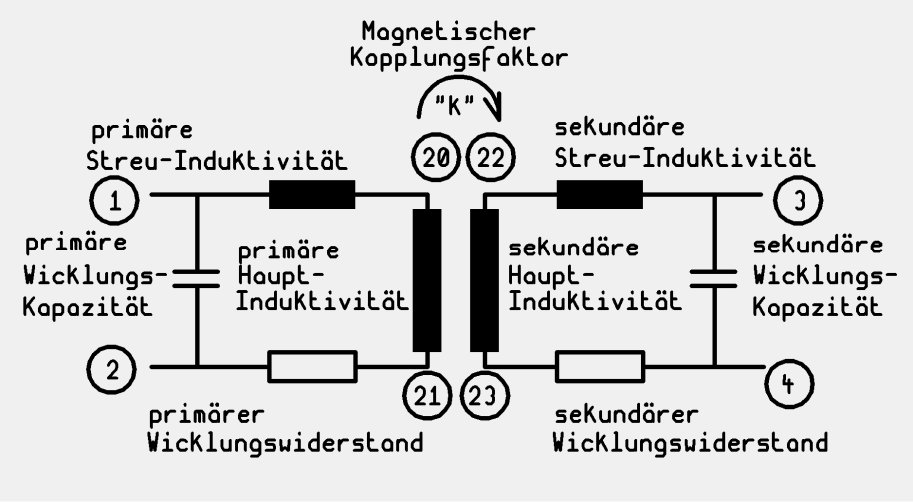


$$U_{ind} = L \bullet \frac{dI}{dt}$$

pausenlos ändern muss und theoretisch bis Unendlich ansteigt. Deshalb ist diese Vorschrift mit einem **„Strombegrenzungs-widerstand“** bereits in SPICE eingebaut).

Im nebenstehenden Schaltbild findet sich deshalb der Widerstand R2 = 0,01Ω zwischen der Spannungsquelle und dem Anschluss der Primärwicklung.

8.2.3. Erstellung des SPICE-Modells für einen realistischen Transformator mit zwei Wicklungen



Grundlage ist das übliche Standard-Ersatzschaltbild, in dem sowohl die Streuung wie auch die „Kopplung“(= Anteil des primären Magnetfeldes im Sekundärkreis) berücksichtigt werden. Ebenso werden die Wicklungswiderstände und die Wicklungskapazitäten einbezogen.

Die Knoten 1 und 2 stellen die Anschlüsse der Primärwicklung, die Knoten 3 und 4 dagegen die der Sekundärwicklung dar. Für die „inneren Knoten“ dieses „Subcircuits“ wählt man höhere Knotennummern, um bei Erweiterungen irgendwelche Kollisionen zu vermeiden.

Nun zum untenstehenden SPICE-Modell mit dem **Subcircuit „xformer_01“ samt den Anschlüssen 1 / 2 / 3 / 4,** das wir mit einem Texteditor erstellen.

Lleak bilden die Streu-Induktivitäten. Sie repräsentieren diejenigen Magnetfeldanteile, die nicht im Kern und damit von der Primär- zur Sekundärwicklung verlaufen. Der Rest der Bauteilbezeichnungen ist (bei einem Blick auf das obige Ersatzschaltbild) selbsterklärend.

Ein **Sternchen** („Asterisk“) kennzeichnet eine Kommentarzeile. Und so würde z. B. das Ergebnis aussehen:

```
*
*   [1]    ---   ||
*             )  ||    .---   [3]
*             )  ||   (
*             )  ||   (
*   [2]    ---   ||    .---   [4]
*

.SUBCKT xformer_01  1  2  3  4
*
** Primary
Lleak1        1       20      1mH
Lpri1         20      21      1H
Rpri1         21      2       1
Cpri1         1       2       20pF
*
**Secondary
Lleak3        3       22      1mH
Lsec1         22      23      1H
Rsec1         23      4       1
Csec1         3       4       20pF

K   Lpri1 Lsec1   0.999
.ENDS
```

Jede Bauteilzeile beginnt mit der Bezeichnung des Teils. Dann folgen die Schaltungsknoten, zwischen denen es hängt. Und den Schluss bildet der Bauteilwert.
Die Kopplung erreicht in der Praxis nie ganz den Wert „1“ und so sollte man sie auch eingeben (hier: k = 0,999). Manche SPICE-Versionen protestieren sogar, wenn das nicht so gemacht wird.

Bei der Primär- und Sekundärwicklung wurden gleiche Werte eingetragen, deshalb handelt es sich hier um einen 1:1 – Übertrager.

Die eigentliche Schaltungsbeschreibung beginnt mit .SUBCKT …… und endet mit .ENDS

==========

Sehr wichtig:

a) Dieses fertige Modell wird nun mit der Bezeichnung **xformer_01.sub** im Ordner

Programme / LTC / LTSpiceIV / lib / sub

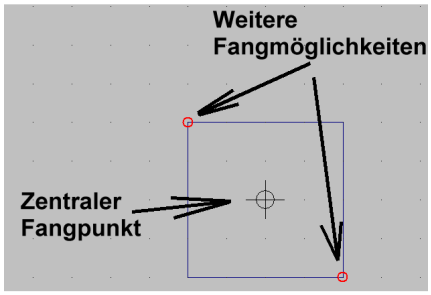
bei den schon vorhandenen übrigen SPICE-sub-Modellen gespeichert. Bitte sorgfältig auf den Pfad UND die korrekte Endung achten…**und natürlich geht das nur, wenn wir vorher den Texteditor mit Administrator-Berechtigung starten….**

b) Es gibt keine **Möglichkeit, das Übersetzungsverhältnis direkt einzugeben**. Das läuft ausschließlich über eine Veränderung der Sekundären Hauptinduktivität und das werden wir im nächsten Projekt sehen.

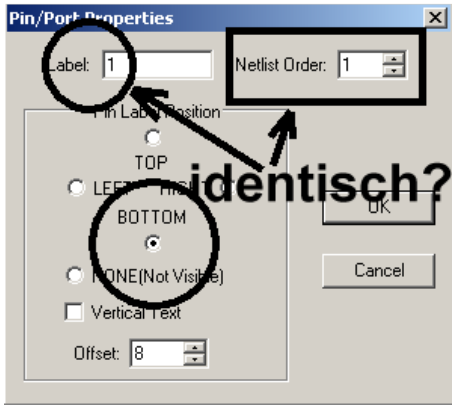
==========

8.2.4. Erzeugung eines passenden Symbols für diesen Transformator

1. Schritt:
Im Menü „File“ findet sich die Option „New Symbol“ -- sie wird angeklickt. Dadurch erscheint ein neuer Bildschirm mit einem „Fadenkreuz“ im Zentrum (es bildet später den „Fangpunkt“ des Symbols beim Anklicken).



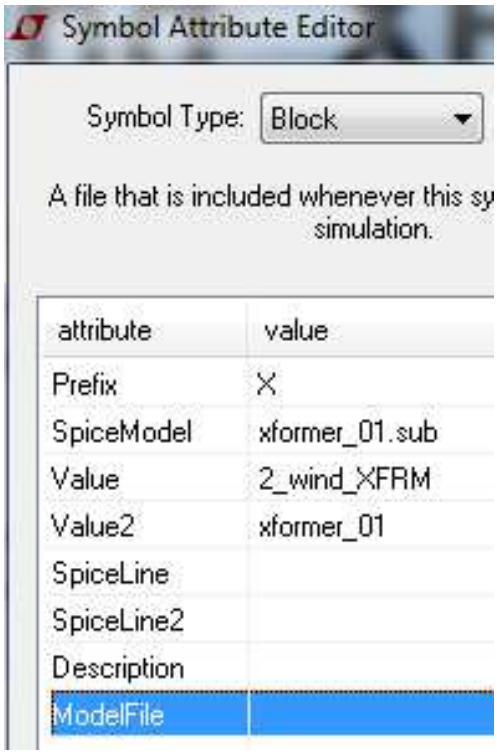
2. Schritt:
Unter „**DRAW**“ wählen wir das Rechteck („Rect“) und zeichnen es in der Größe **20mm x 20mm** um den Fangpunkt herum (…über dem Bildschirm liegt ein feines 5mm – Punktraster). Die roten Kreise an den Ecken des Rechtecks müssen sein, denn sie bilden später weitere Fangmöglichkeiten.



3. Schritt:
Unter „**Edit**“ findet sich „**Add Pin / Port**“. Damit können wir die erforderlichen vier Anschlüsse erzeugen. Im zugehörigen Menü müssen wir jedoch jedes Mal **die Pin-Nummer UND die Position des Anschlusses gegenüber der Pinbezeichnung („Label“) eintragen**.
Die nebenstehenden Einträge gelten für Pin 1.

Bitte darauf achten, dass auch der Pin-Label und die Netlist-Order übereinstimmen!

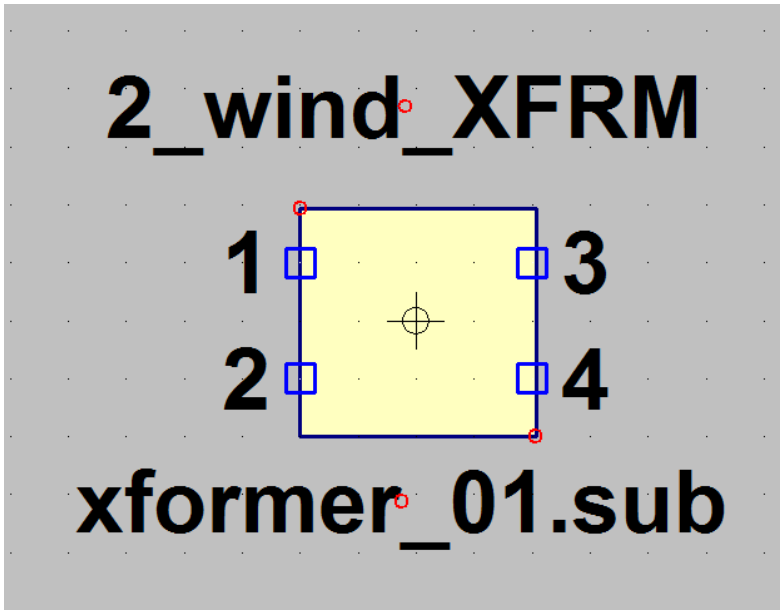
Nach dem Klick auf OK hängt der Pin am Cursor und kann dann auf der Kante des Symbols abgesetzt werden. Dann folgen die drei weiteren Pins (Siehe Bild auf der nächsten Seite).



4.Schritt:
Jetzt wird es ernst, denn nun soll aus dem hübschen Bildchen unser echter Transformator werden. Also heißt es, über „**EDIT / Attributes / Edit Attributes**“ an das nebenstehende Menü heranzukommen.

Dann übernehmen wir nacheinander diese Vorgaben, wobei wir auf die betroffene Zeile in der Liste klicken und im so markierten Feld eingeben **(Hier bedeutet z. B. Prefix „X“ einen Subcircuit).**

Warnung:
In der Zeile für Value2 MUSS IMMER ein Eintrag stehen (…am besten dieselbe Bezeichnung wie das Spice-Modell). Wenn der fehlt, kann es eine völlig unerklärliche Fehlermeldung geben….



6. Schritt:
Jetzt gilt es noch, wichtige Informationen beim Aufruf des Symbols für den Anwender sichtbar zu machen. Dazu geht es nochmals in „**Edit**“ und „**Attributes**“, jetzt aber zu „**Attribut Window**“. Wenn wir nun auf „**value**“ klicken, dann hängt die Bezeichnung „**2_wind_XFRM**“ am Cursor und kann oberhalb des Symbols platziert werden. Dann wiederholt man das Spiel und wählt dieses Mal „**SpiceModel**“. Ganz korrekt erscheint das Modell-File „**xformer_01.sub**“ am Cursor und kann unterhalb des Symbols angeordnet werden. Fertig!

7. Schritt:
Den Abschluss bildet das korrekte Abspeichern des Symbols. Wir wählen **„File**“ und „**Save as**“, müssen uns aber dann über **„LTC / LTSpiceIV“ und „lib“ bis zum Ordner „sym“ (= Symbols)** durchhangeln.

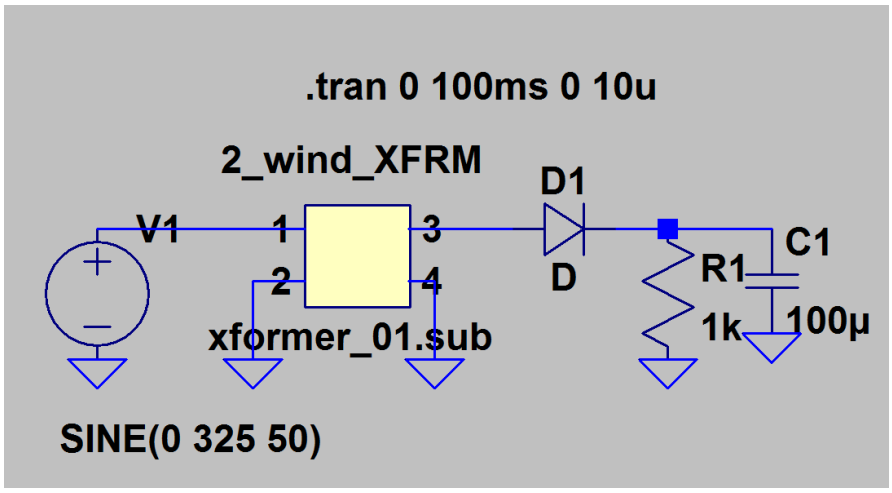
Darin sollte man sich einen neuen Ordner „Xformers“ anlegen und das fertige Bauteil als „xformer_01.asy“ endgültig speichern….**und das klappt natürlich nur mit Administrator-Berechtigung, die man bereits beim Programmaufruf einschalten muss** (= über einen rechten Mausklick auf den Icon beim Programmstart zugänglich).

Tatsächlich: nun ist wirklich alles geschafft!

8.2.5. Einpuls-Gleichrichter mit realistischem Trafo

Nun wollen wir die Ansprüche etwas höher schrauben und uns eine Schaltung mit einem realen Netztransformator samt Einpuls-Gleichrichter und Ladekondensator vornehmen.

Aufgabe:
Untersuchen Sie ein Netzteil mit realem Netztrafo, Gleichrichterdiode, Lastwiderstand RL = 1kΩ und Ladekondensator C = 100µF. Der Trafo-Eingang liegt an der Netzspannung mit 230V / 50 Hz, das Übersetzungsverhältnis sei 20:1. Simulieren Sie die Zeit von Null bis 200 Millisekunden.



Lösung:
Die meisten Bauteile (Spannungsquelle, Kondensator, Widerstand, Diode, Ground-Symbol) kennen wir ja schon. Dazu kommt nun der realistische Übertrager aus dem vorigen Kapitel. Bitte gleich alles platzieren sowie die Werte der Bauteile, der Spannungsquelle und die Simulationsanweisung eintragen.

Jetzt müssen wir uns noch um den Transformator kümmern und die Sache mit dem Übersetzungsverhältnis unterbringen. Dabei gilt folgendes:

$$\frac{L_{Primär}}{L_{Sekundär}} = (ü)^2$$

Wenn wir (wie in den meisten Fällen für einen fast idealen Trafo zulässig) mal von einer Primär-Induktivität von 1Henry ausgehen, dann müssen wir in unserem SPICE-File „xformer_01.lib“ eine Sekundär-Induktivität von

$$L_{Sekundär} = \frac{L_{Primär}}{(20)^2} = \frac{1H}{400} = 2{,}5mH$$

benützen.

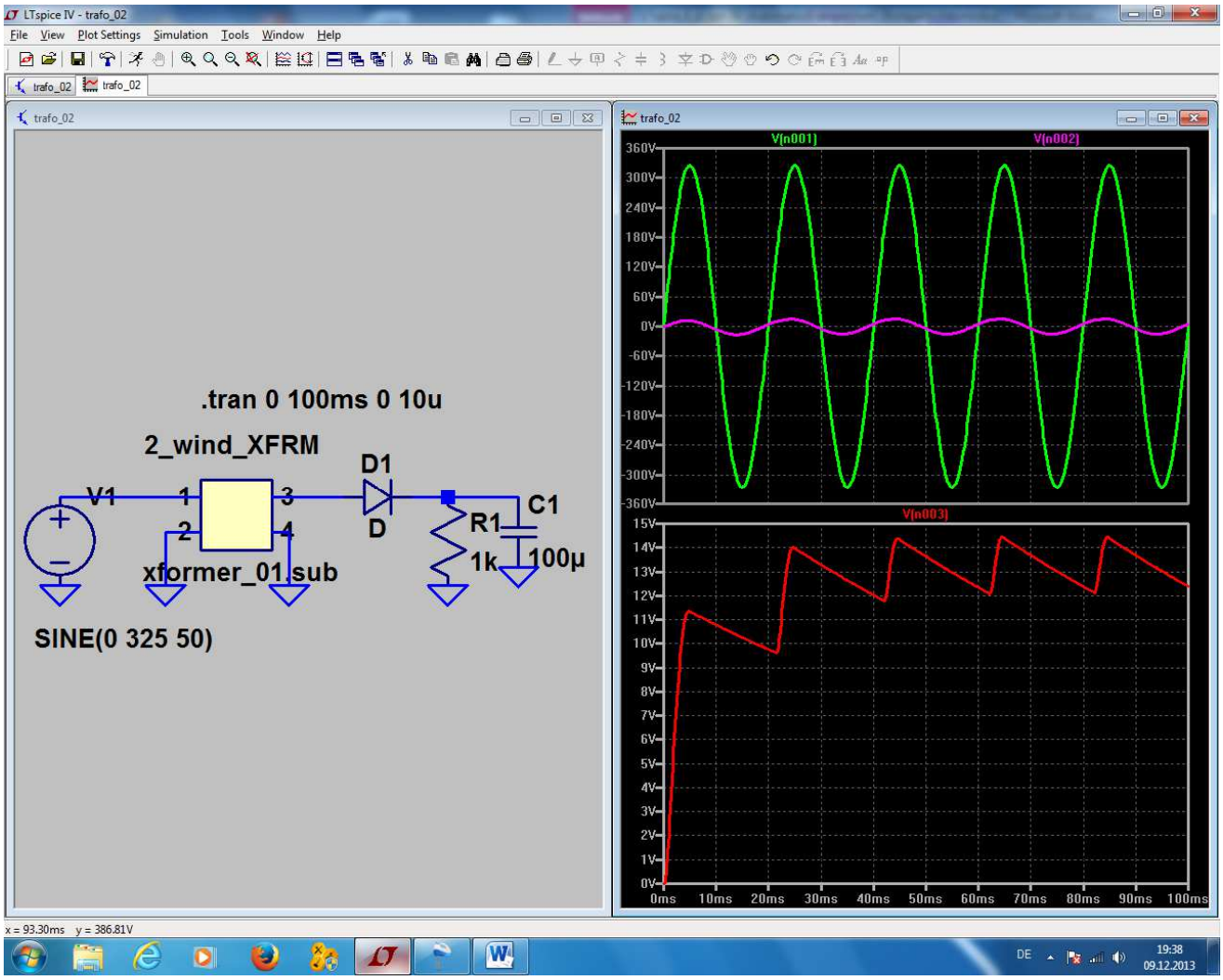
```
*
*    [1]    ---.    ||
*               )   ||    .---    [3]
*               )   ||   (
*               )   ||   (
*    [2]    ---.    ||    .---    [4]
*
*turn ratio = 20:1
*
**
.SUBCKT xformer_01  1  2  3  4
*
** Primary
Lleak1          1       20      1mH
Lpri1           20      21      1H
Rpri1           21      2       100
Cpri1           1       2       500pF
*
**Secondary
Lleak3          3       22      50µH
Lsec1           22      23      2.5mH
Rsec1           23      4       5
Csec1           3       4       2.5pF

K   Lpri1       Lsec1    0.999
.ENDS
```

Für einen fast idealen Trafo kann man dann sowohl die Wicklungs-Widerstände, die Streu-Induktivitäten und die Wicklungs-Kapazitäten verkleinern. Aber man darf sie nie komplett auf null setzen!

Speziell einen Widerstand von Null Ohm in Reihe zu einer Induktivität beantwortet das originale SPICE sofort mit einem Abbruch und einer Fehlermeldung!

So sieht dann ein **realistisches Trafo-Modell für ein Übersetzungsverhältnis von 20:1** aus. Also öffnen wir das File **„xformer_01.sub“** **mit Administrator-Rechten**, ändern es und speichern es wieder am korrekten Ort (= **Ordner „sub“ im „lib“-Verzeichnis von LTSpiceIV**).

Der Rest, also die eigentliche Simulation, ist eine Kleinigkeit und bekannt. Deshalb folgt gleich das Ergebnis für die alle Spannungen (am Eingang sowie vor und nach der Diode):



8.3. Verwendung einer anderen Diode in der Gleichrichterschaltung

Da gibt es zwei Möglichkeiten:

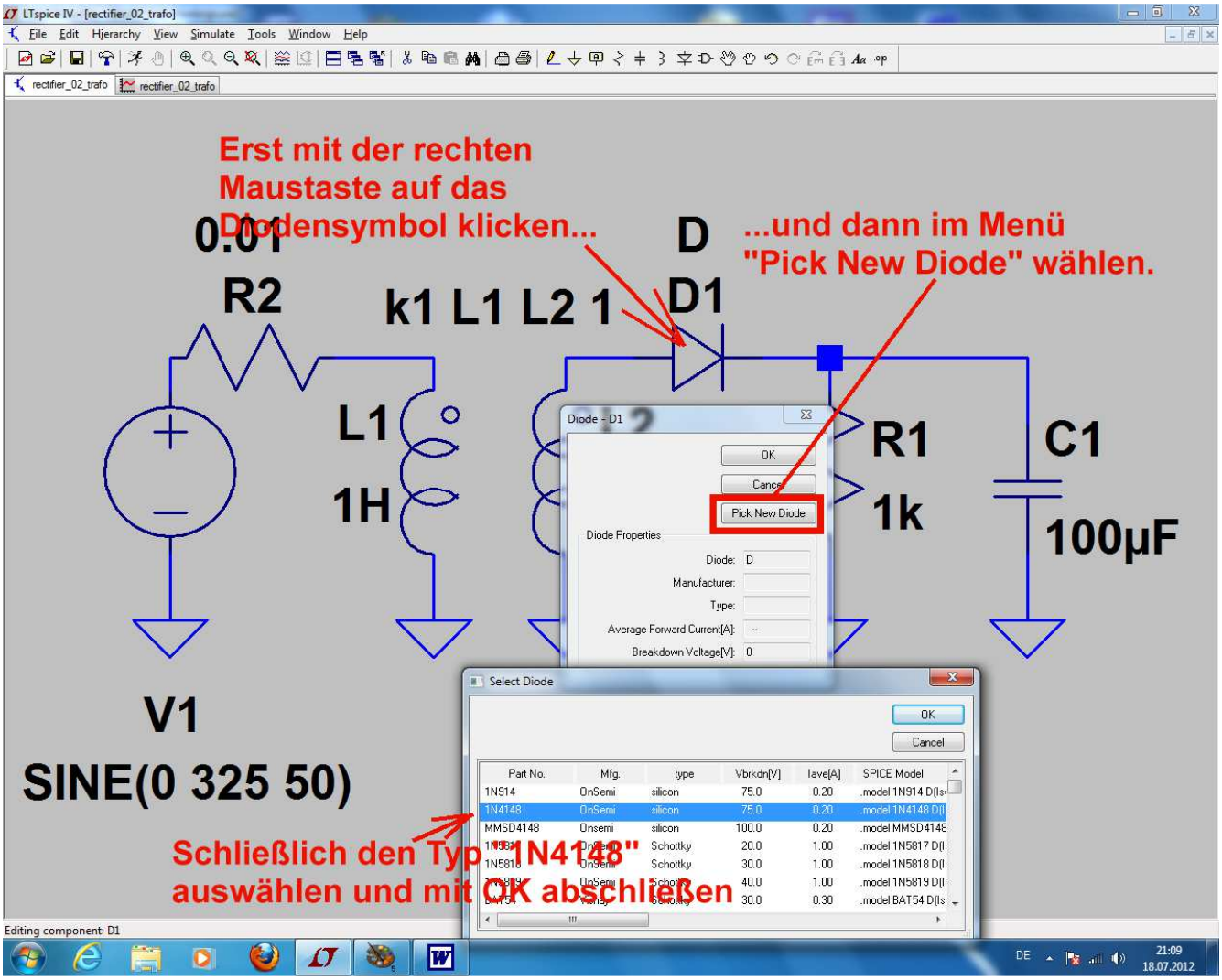
a) Wir benützen **eine der mitgelieferten Diodenmodelle** anstelle der Universaldiode „D“ oder

b) Wir **holen das** SPICE-Modell der gewünschten Diode aus dem Internet.

Wir wollen an dieser Stelle nur die Umstellung auf eine Diode aus der Vorschlagsliste ansehen. Die Prozedur zur Umstellung auf ein Diodenmodell aus dem Internet ist aufwendiger und erhält deshalb ein eigenes Kapitel.

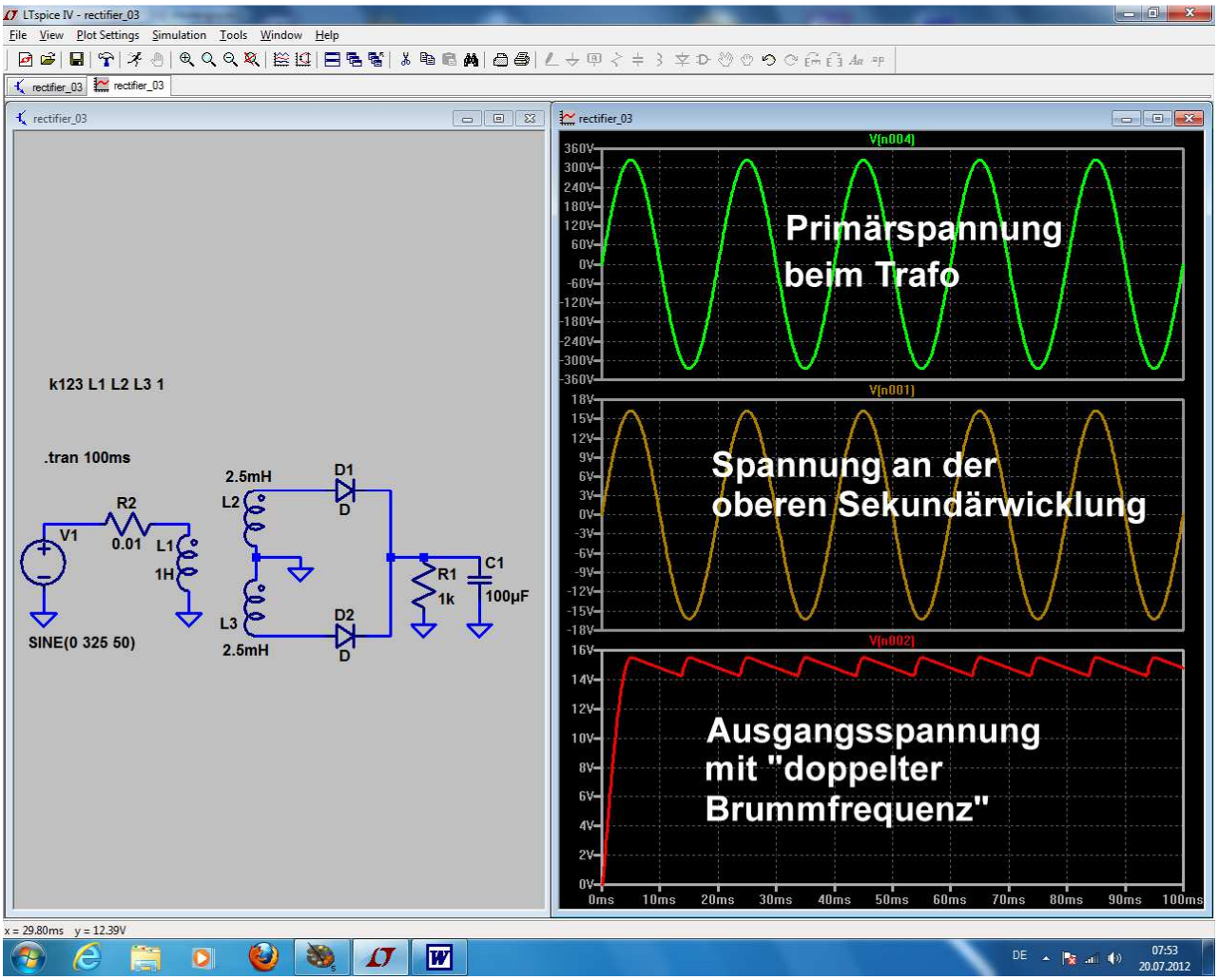
8.3.1. Einsatz einer Diode aus dem mitgelieferten Vorrat von LTspice

Sehen wir uns diese **erste Methode** an, wenn wir uns für die „**1N4148“** entscheiden:



8.4. Zweipuls-Gleichrichter mit idealem Trafo

Dazu ist eine weitere Sekundärwicklung erforderlich, die zur ersten in Reihe geschaltet wird. Aber dabei bitte das Bauteile „**ind2**“ verwenden und auf die richtige Polung achten….



Nun werden beide Halbwellen der Sekundärspannung für die Gleichrichtung genutzt und deshalb beobachten wir bei der Ausgangs-Spannung die doppelte Eingangs-Frequenz

(unter Fachleuten = „doppelte Brumm-Frequenz“)

8.5. Zweipuls-Gleichrichter mit realistischem Trafo

Hier ist ein neues Bauteil erforderlich: ein Transformator mit zwei getrennten Sekundärwicklungen. Und wir wollen wieder den steinigen Weg mit dem selbst erstellten Trafomodell wählen.

1. Schritt:

```
*
*    [1]      ---.    ||
*                 )   ||    .---    [3]
*                 )   ||   (
*                 )   ||   (
*    [2]      ---.    ||    .---    [4]
*                     ||    .---    [5]
*                     ||   (
*                     ||   (
*                     ||    .---    [6]
*
*
*
*turn ratio = 20:1
*
*
*
.SUBCKT xformer_02  1  2  3  4  5  6
*
** Primary
Lleak1                1       20      1mH
Lpri1                 20      21      1H
Rpri1                 21      2       100
Cpri1                 1       2       500pF
*
**Secondary_01
Lleak2                3       22      50µH
Lsec1                 22      23      2.5mH
Rsec1                 23      4       5
Csec1                 3       4       2.5pF

**Secondary_02
Lleak3                5       24      50µH
Lsec2                 24      25      2.5mH
Rsec2                 25      6       5
Csec2                 5       6       2.5pF

K   Lpri1 Lsec1 Lsec2    0.999
.ENDS
```

Also gehen wir in den Texteditor, rufen unser File „xformer_01.sub“ auf und ändern es ab.
Wir sehen gleich wieder ein Übersetzungsverhältnis von Ü = 20:1 vor und tragen die zu einer Primär-Induktivität von 1 H gehörenden **Sekundär-Induktivitäten von je 1H / 400 = 2,5mH** in die entsprechenden Zeilen ein.

Da wir mit einem nur fast idealen Übertrager arbeiten wollen, machen wir die Streu-Induktivitäten, die Wicklungswiderstände und die Wicklungskapazitäten wieder sehr klein, aber nicht Null.

Achtung:
Bitte nicht vergessen, die interne und externe Knoten-Nummerierung korrekt fortzusetzen!
Extern gelten nun die knoten 1….6, intern dagegen die Knoten 20…25

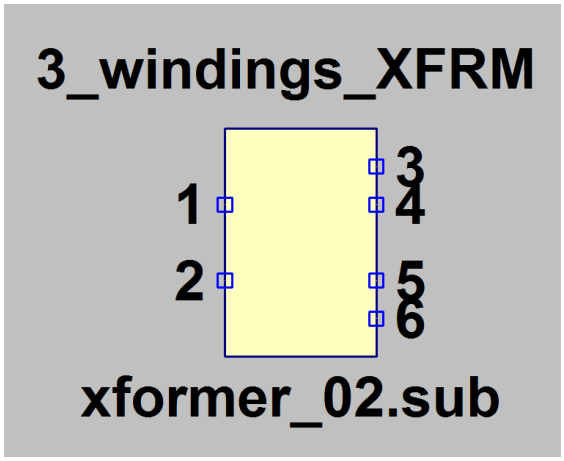
Bitte ebenfalls nicht vergessen:
Wir haben es nun mit **drei Induktivitäten zu tun, die miteinander verkoppelt sind** und müssen das in einem entsprechend **geänderten magnetischen Kopplungsfaktor „k“ als Spice-Direktive angeben:**

k Lpri1 Lsec2 Lsec2 0.999

Stimmt alles, dann speichern wir dieses neue File als „**xformer_02.sub**“ wieder im „sub“-Ordner der Library (Bitte an die Administrator-Rechte denken…).

2. Schritt:

Auch ein neues Symbol ist nötig. **Die Erstellungs-Prozedur läuft exakt so ab wie im Kapitel 7.2.4.** und am Ende muss der Übertrager so aussehen.



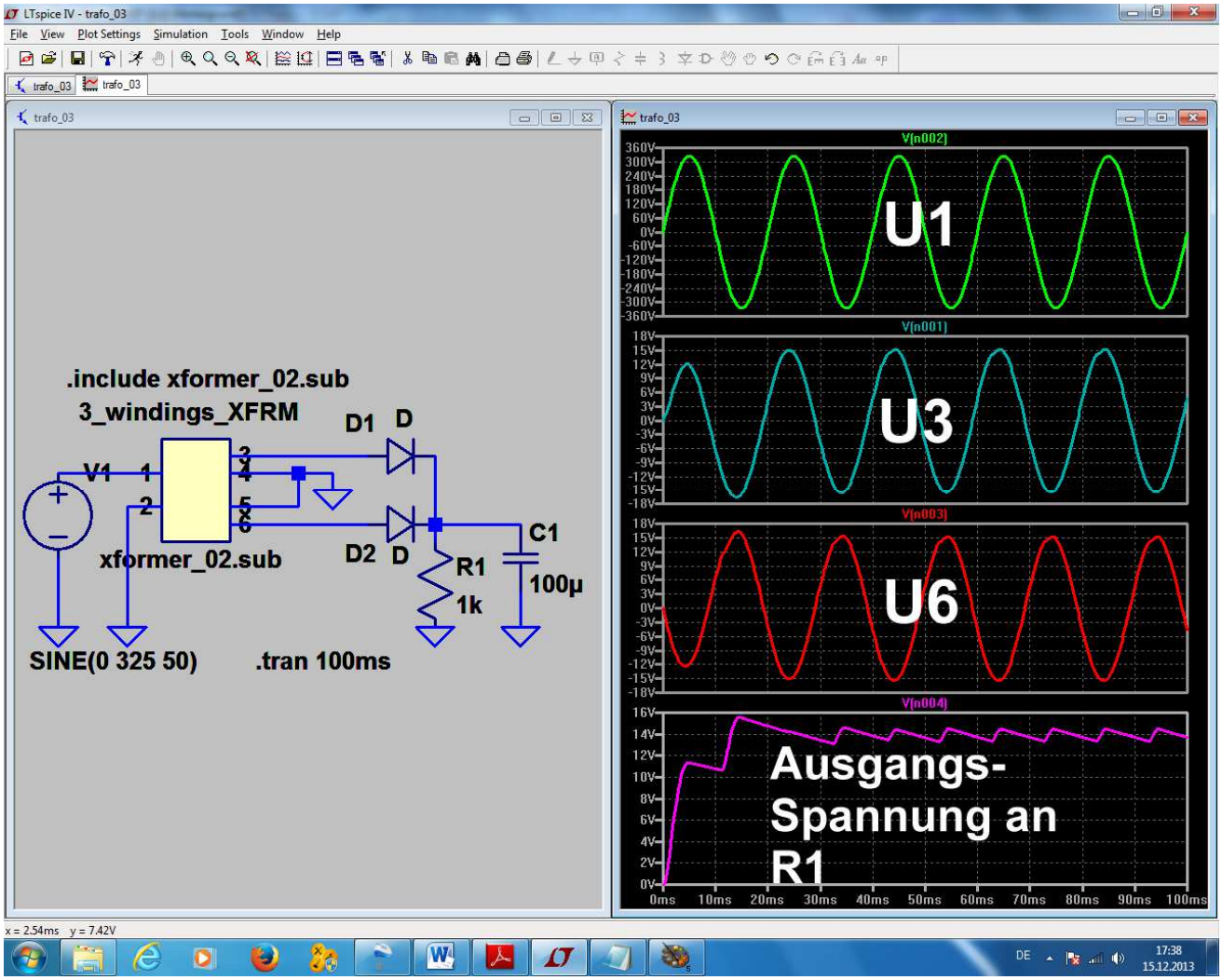
Bitte aber bei der Erstellung der **Attributliste** wieder dran denken:

In der Zeile für Value2 MUSS IMMER ein Eintrag stehen (…am besten dieselbe Bezeichnung wie das Spice-Modell). Wenn der fehlt, kann es manchmal eine völlig unerklärliche Fehlermeldung geben….

3. Schritt:

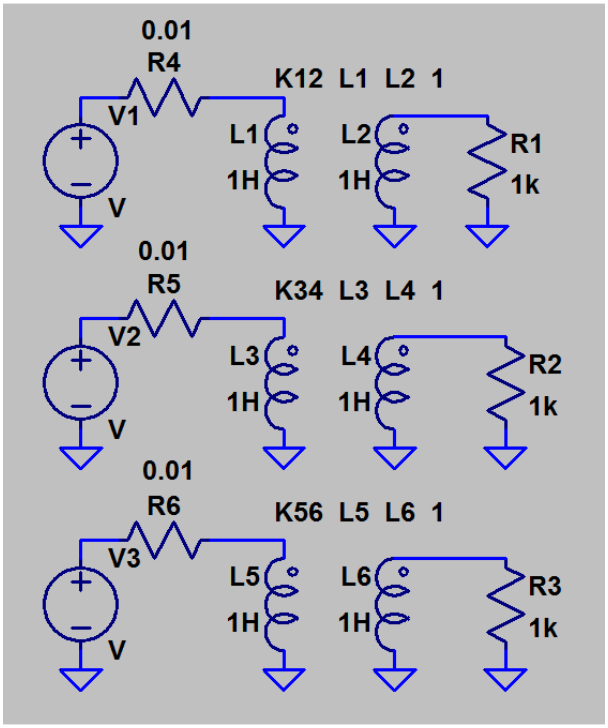
Nun wird die „Zweipuls-“ (= Mittelpunkts-) Schaltung unter Verwendung dieses Übertragers sowie 2 Dioden vom Typ 1N4007 gezeichnet und alles simuliert.

Schaltung und Ergebnis für die beiden Sekundärspannungen des Trafos sowie für die Ausgangsspannung: Siehe nächstes Blatt.



9. Drittes Projekt: Drehstrom

9.1. Programmierung eines Drehstromsystems



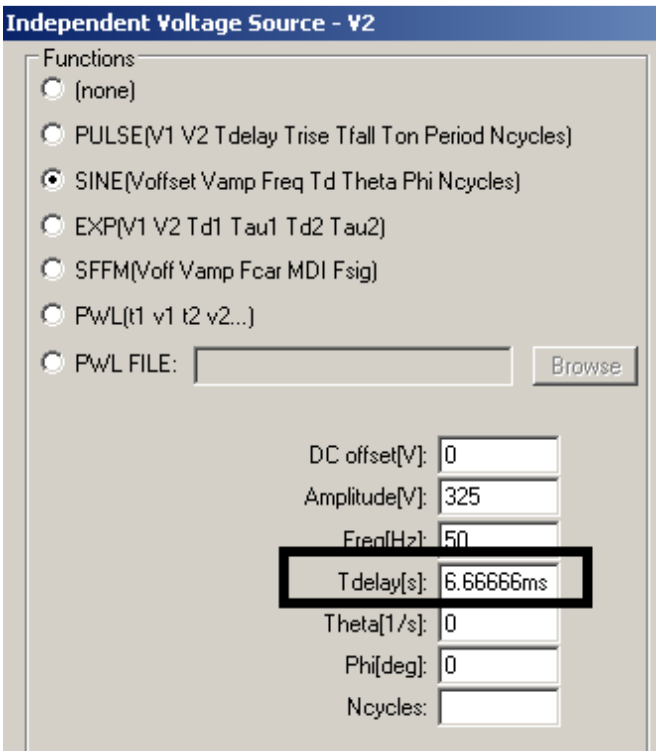
Wir wollen uns mit drei Spannungsquellen, drei idealen Trenntrafos und drei Lastwiderständen die Sache mit dem Drehstrom etwas näher ansehen. **Aber bei den Übertragern machen wir beide Hauptinduktivitäten gleich (…jeweils 1H) und erzielen so ein Übersetzungsverhältnis von ü = 1:1.**

Diese Bauteile werden geholt und in der nebenstehenden Form verdrahtet.

Hinweis:
Man kann sich die Zeichenarbeit sehr erleichtern, wenn man zuerst den obersten Schaltungsteil komplett erstellt und anschließend die Kopierfunktion

(Taste F6)

nutzt. Anschließend braucht man nur noch die Induktivitätsbezeichnungen sowie die Kopplungsangaben anzupassen!



Anschließend fahren wir bei jeder Spannungsquelle mit dem Cursor auf das Symbol und klicken rechts. Bei allen drei Quellen sind folgende Einträge identisch:

DC offset = 0 Volt

Amplitude = 325 V (= Spitzenwert für einen Effektivwert von 230 V)

Frequenz = 50 Hz

Phi = 0 degrees (= Phasenlage beim Kurvenstart)

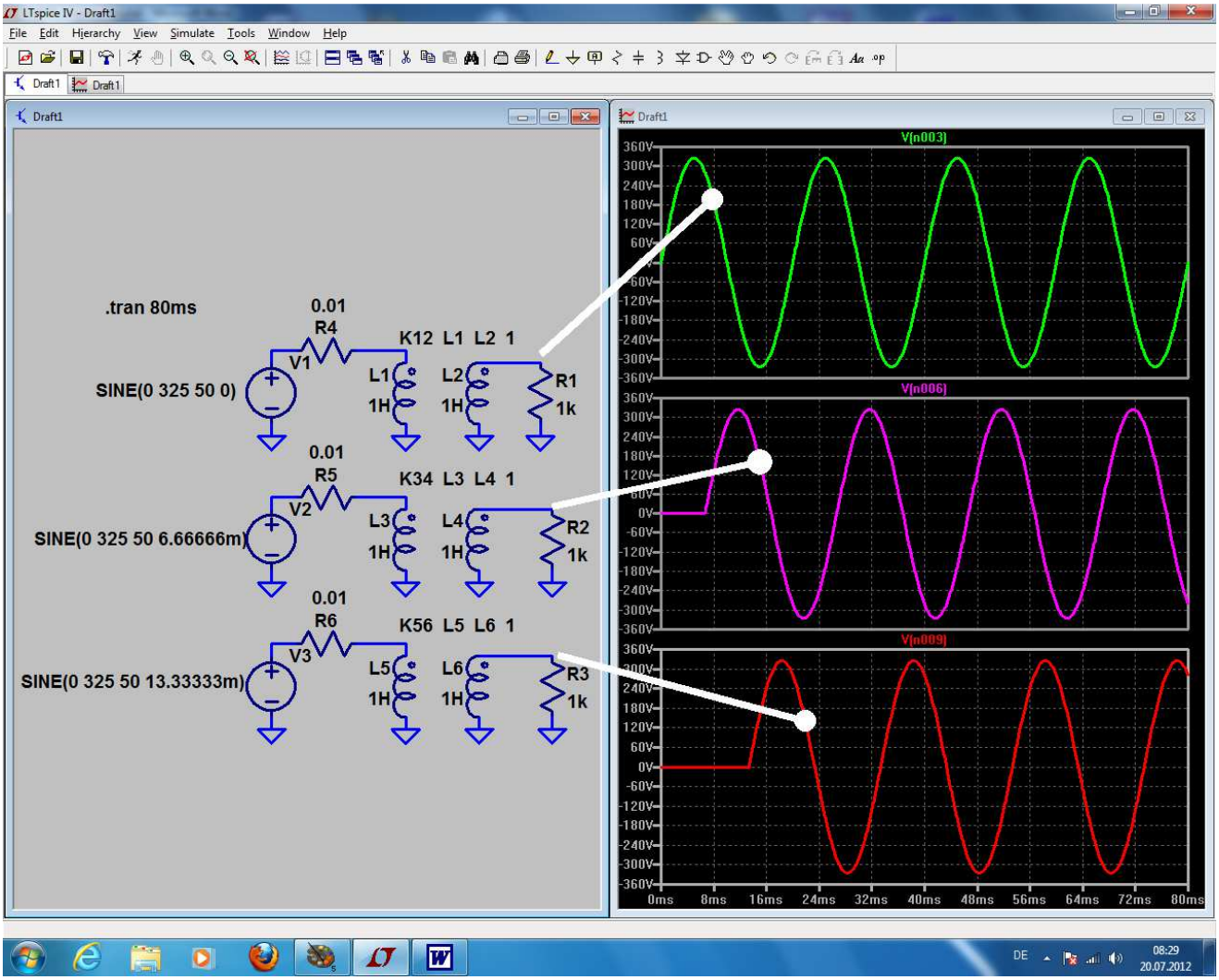
Wichtig:
Die beim Drehstrom erforderliche Phasenverschiebung von 120 Grad zwischen den einzelnen Kurven erreichen wir durch drei unterschiedliche Startverzögerungen (= Tdelay)!

Bei V1 beträgt sie Null Millisekunden

Bei V2 beträgt sie 6,66666 ms

Bei V3 beträgt sie 13,33333ms

Programmieren wir nun noch eine Simulationsdauer von 80 ms, dann sieht das Simulationsergebnis so aus:



9.2. Prinzip der Drehstrom-Lichtmaschine

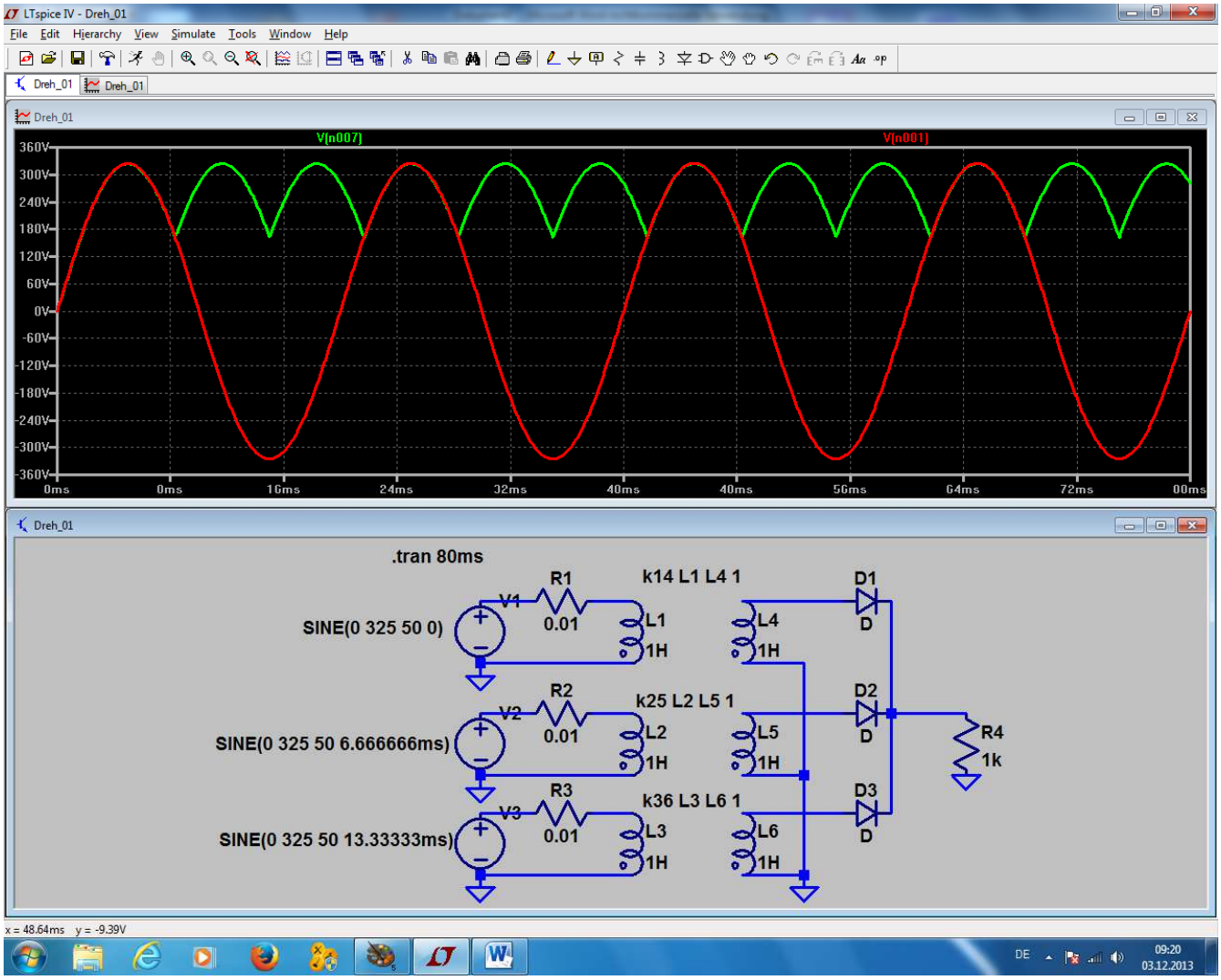
Dazu speichern wir die vorige Schaltung als neues Projekt und geben ihr einen anderen Namen. Dann wollen wir die beiden wichtigsten Schaltungen untersuchen, die hier zum Einsatz kommen.

9.2.1. Die M3-Schaltung

Man erkennt sie daran, dass hier

a) nur **drei Dioden** nötig sind und

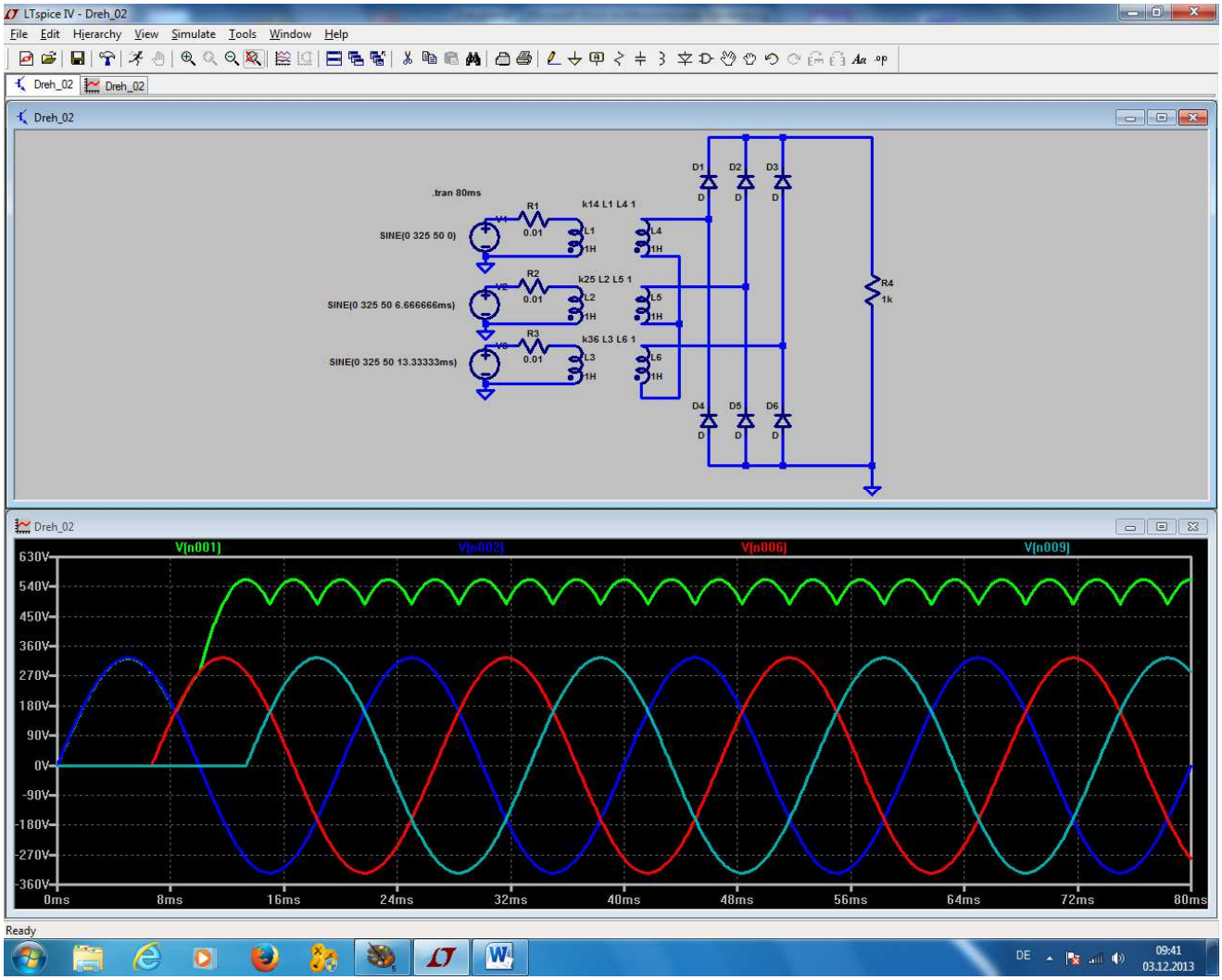
b) die **drei Sekundärwicklungen „im Stern“ zusammengeschaltet** sind (= untere Wicklungsenden miteinander verbunden und an Masse gelegt). Ebenso ist das untere Ende des Lastwiderstandes geerdet.



Sehr schön ist die Gleichrichtung der drei beteiligten und gegeneinander phasenverschobenen Spannungen zu sehen. Die „**Brummfrequenz“ hat den dreifachen Wert der Speisefrequenz.**

9.2.2. Die B6-Schaltung

Bei ihr werden **6 Dioden** eingesetzt und der „**Sternpunkt“ darf nicht geerdet werden**. Damit erhält man ein Verhalten wie beim Brückengleichrichter:



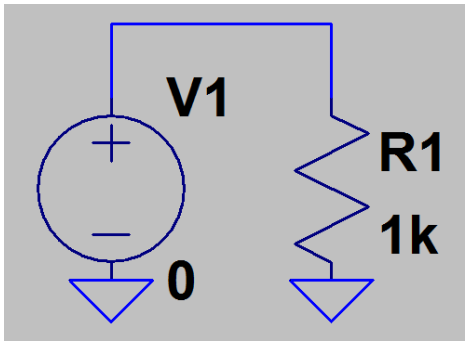
Bitte genau hinsehen:

die Brummfrequenz am Ausgang ist nun 6x höher als die Eingangsfrequenz UND die Ausgangsspannung ist größer!

10. Viertes Projekt: Darstellung von Bauteil-Kennlinien

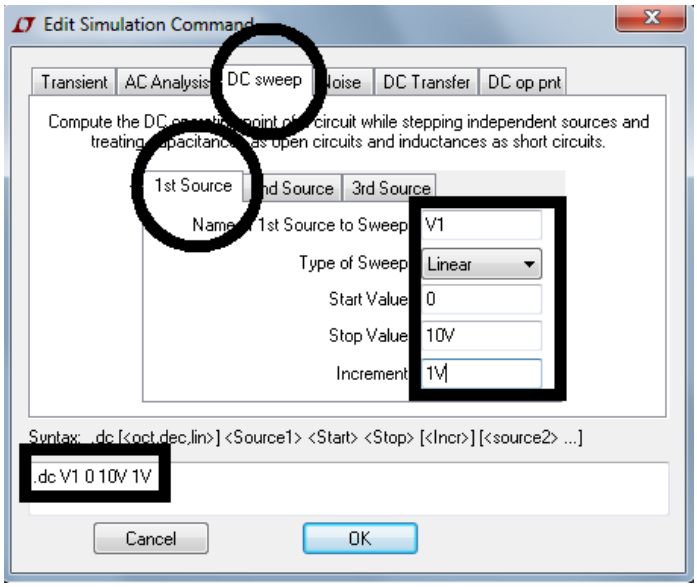
10.1. Ohm'scher Widerstand

Dieses Bauteil eignet sich besonders gut zur Demonstration des Vorgehens, wenn U-I-Kennlinien sichtbar gemacht werden sollen.



Dazu öffnen wir ein neues Projekt („New Schematic") und zeichnen uns die nebenstehende Schaltung.
Beim Widerstand „R1" wählen wir einen Wert von 1kΩ.
Dann müssen wir das Property Menu der Spannungsquelle öffnen (= rechter Mausklick auf das „Plus-Zeichen" im Symbol....) und dort im rechten oberen Eck beim DC-Value „Null Volt" hineinkriegen. Wir tippen also in das zugehörige Fenster „0" ein.

Die erfolgreiche Übernahme prüfen wir an unserem Schaltbild.



Im Menü „Simulate" wählen wir **„Edit Simulation Command"** und haben dann den nebenstehenden Bildschirm vor uns. Da können wir sogar drei Quellen sweepen…

Auf der Karteikarte „**DC Sweep**" programmieren wir nun die **Spannungsquelle V1** mit folgenden Vorgaben:

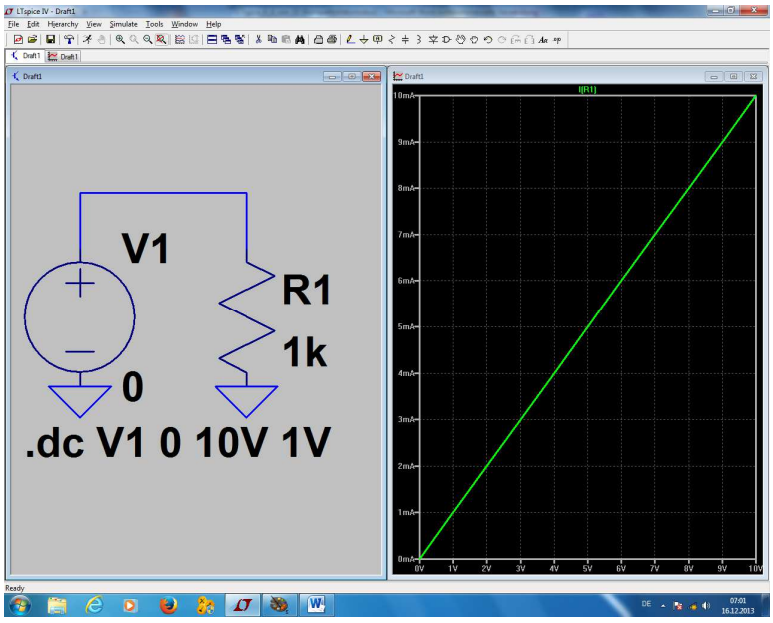
Startwert = Null Volt

Endwert = 10 Volt

Schrittweite = 1V

Linearer Sweep

Das damit erzeugte Simulation Command **„.dc V1 0 10V 1V "** findet sich im unteren Feld und sollte nochmals kontrolliert werden.



Dann bitte mit OK schließen und die Simulation starten. Fährt man dann mit dem Cursor im Schaltbild auf den oberen Anschluss des Widerstandes, bis sich der Cursor in eine **„Stromzange"** verwandelt, dann reicht ein kurzer Klick auf diesen Punkt.

Da haben wir alles...

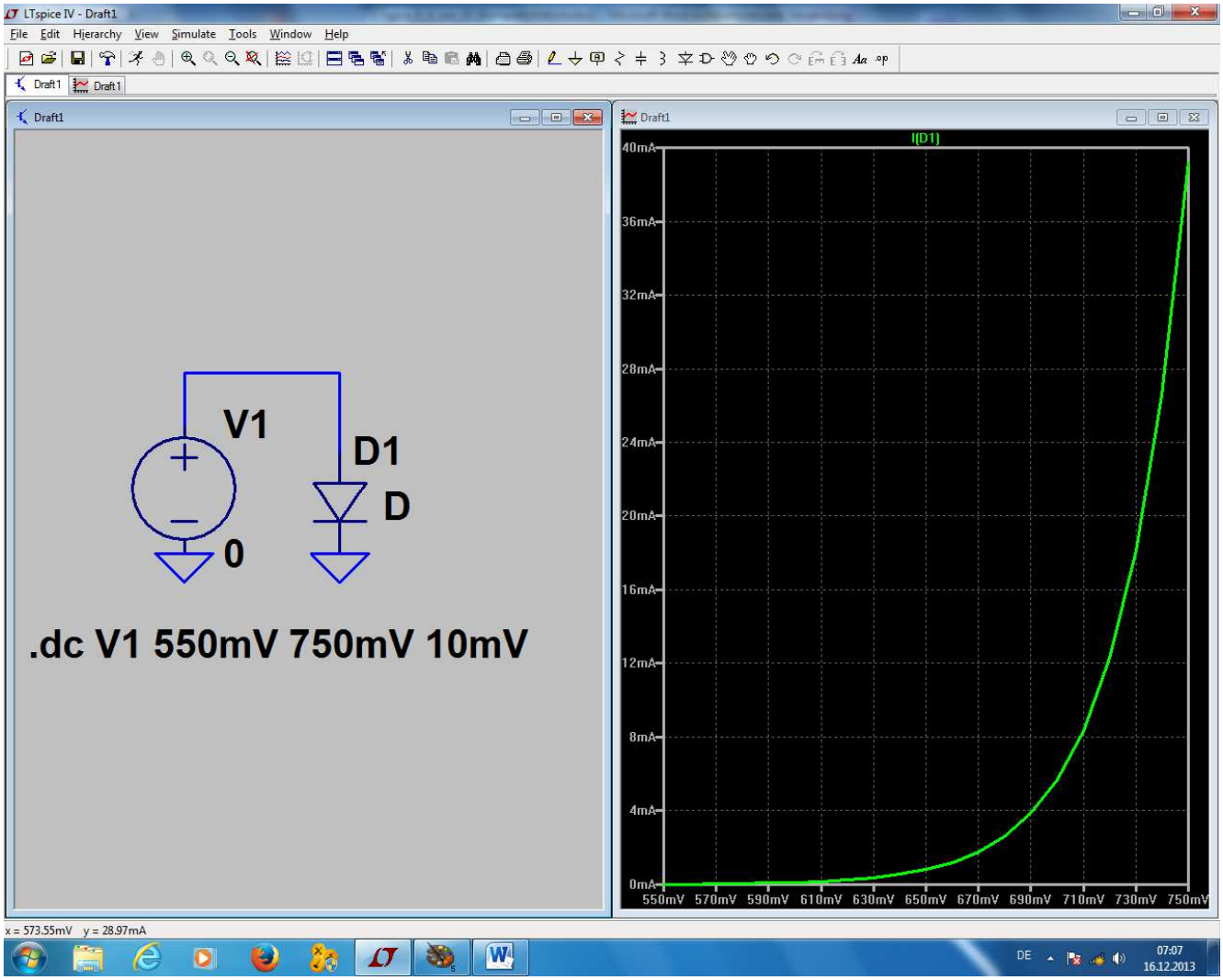
10.2. Diode

Das ist jetzt eine sehr einfache Übung:

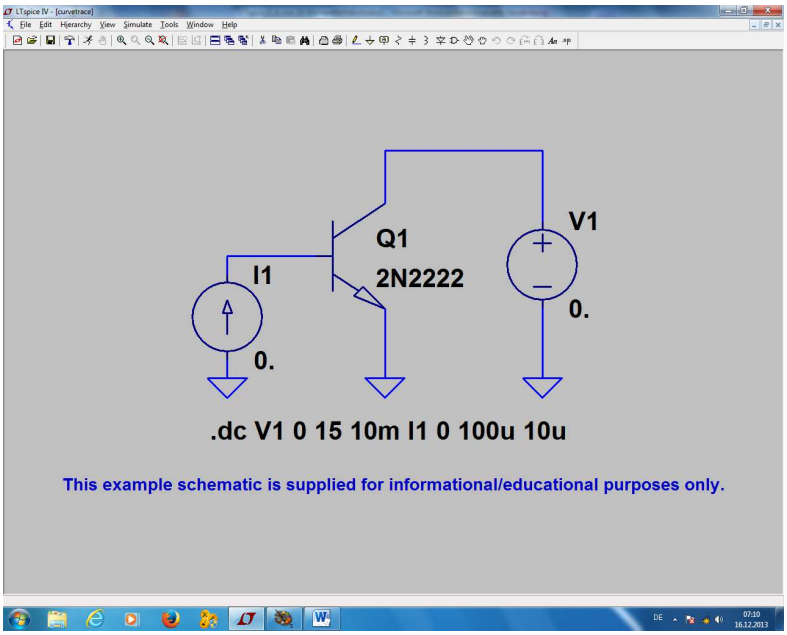
Erst das Projekt unter einem neuen Namen abspeichern. Dann den Widerstand (nach Druck auf F5) herauslöschen und dafür die Diode (...sie findet sich als Button in der oberen Menüleiste oder als „diode“ in der Bauteile-Liste...) einfügen. Anschließend bitte noch das Simulation Command umändern in

.dc V1 550mV 750mV 10mV

Damit wird nur der Spannungsbereich von 550mV bis 750mV dargestellt und in Schritten von 10mV simuliert. Das ist für eine Siliziumdiode genau richtig.



10.3. NPN-Transistor



Da gibt es in der „example“-Bibliothek des LTSpiceIV-Programmes ein sehr hübsches Beispiel mit dem Namen „**curvetrace**“, das wir uns einfach hernehmen:

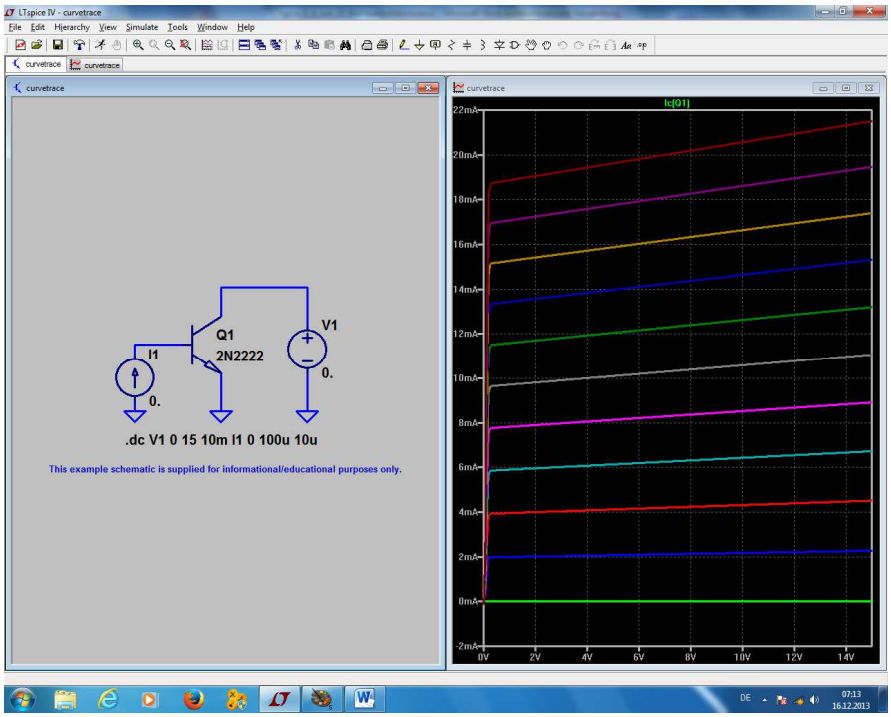
Man erkennt:

a) Am Kollektor des Transistors wird die Gleichspannungsquelle V1 angelegt. Sie liefert zunächst Null Volt.

b) Eine Konstantstromquelle I1 speist die Basis des Transistors. Auch ihr Startwert ist „Null“

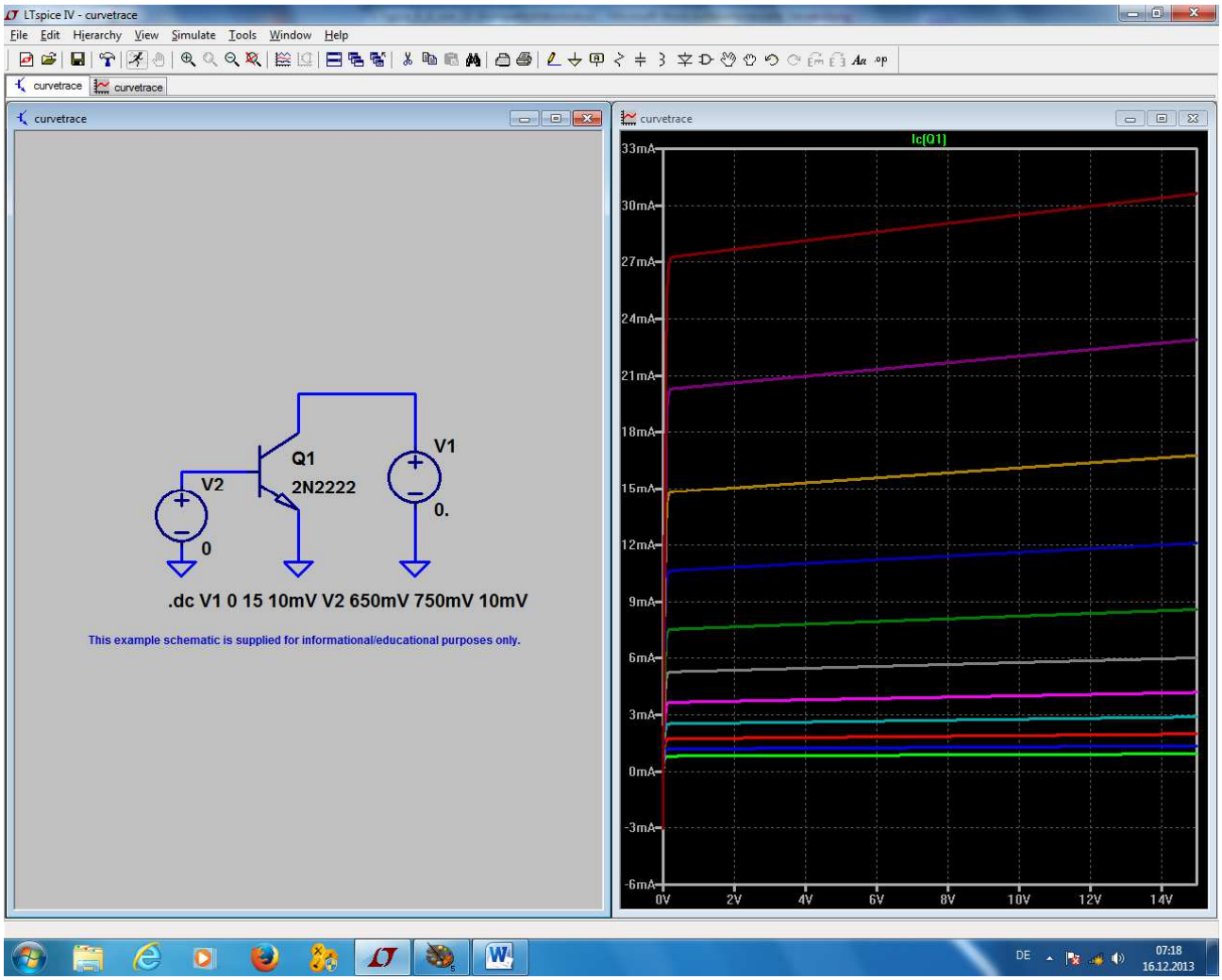
Das Simulation Command besagt: **.dc V1 0 15 10mV I1 0 100u 10u**

Ändere die Spannung V1 von Null Volt bis +15V in Schritten von 10mV. Benütze diese Spannung für die waagrechte Achse des Ergebnisdiagramms (= Abszisse).
Ändere nun den Strom I1 von Null bis 100 Mikroampere in Schritten von 10 Mikroampere und schreibe alle Kurven in das Ergebnisdiagramm.



Fährt man nun im Schaltplan mit der Maus auf den Kollektor des Symbols, dann wird sich plötzlich der Cursor in eine „Stromzange“ verwandeln. Nun reicht ein linker Mausklick, um das gewünschte Ergebnis zu erhalten.

Wer sich dagegen für den Zusammenhang zwischen Basisspannung und Kollektorstrom interessiert, braucht diese Schaltung:



a) Die Stromquelle an der Basis wird durch eine Spannungsquelle ersetzt. Ihr Startwert ist „Null“

b) Das Simulation Command wird neu geschrieben:

.dc V1 0 15V 10mV V2 650mV 750mV 10mV

Die Angaben für die Kollektorspannung V1 bleiben darin unverändert. Die Basis-Spannungsquelle V2 wird in Schritten von 10mV von 650mV auf 750mV geändert.

Ergebnis: Siehe oben...

10.4. N-Kanal-Sperrschicht-FET

Gegenüber der Transistor-Simulation des letzten Kapitels ändert sich nur eine Sache: wir müssen jetzt einen solchen FET (Bauteil „njf“ aus der Bibliothek...) verwenden. Da sollte man sich aber die Mühe machen und einen bekannten Typ wählen. Wir einigen uns auf den

2N3819

und den holen wir uns aus dem vorgeschlagenen Vorrat. (Wissen Sie noch, wie das ging? Erst den FET als „**njf**“ holen und absetzen. Dann ein rechter Mausklick auf das Schaltzeichen, unter „**Pick new JFET**“ nachsehen und „**2N3819“** anklicken). Dann folgen die beiden Spannungsquellen.

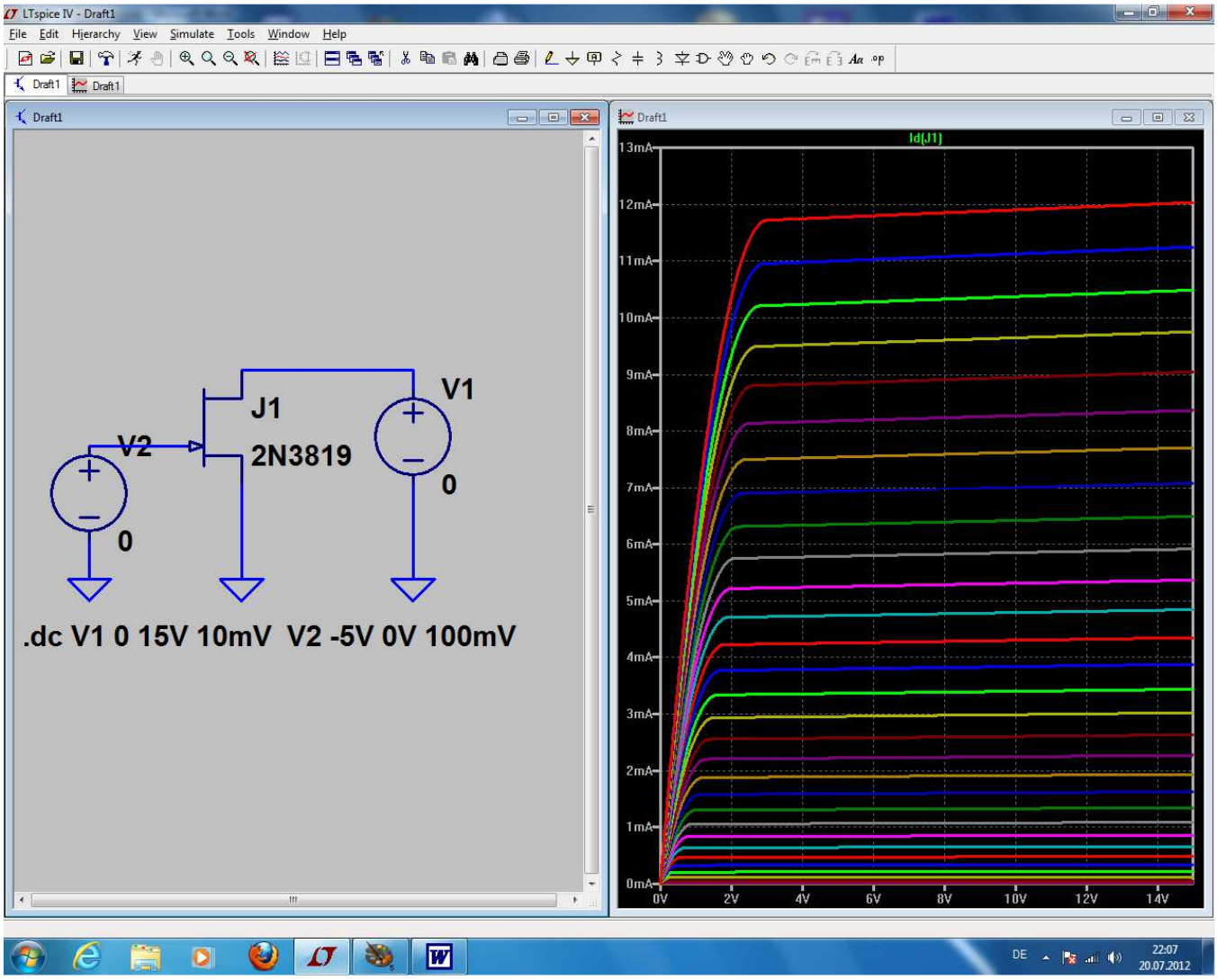
Wenn man nun noch beim Simulation Command den Spannungsbereich von V2 -- wie es sich für einen Sperrschicht-FET bei der Gate-Spannung gehört! -- auf

-5V bis 0V in Schritten von 100mV

einstellt, dann sollte alles klappen. Die Drainspannung umfasst wieder den Bereich von 0….+15V und wird in Schritten von 10mV gesteppt. Dazu gehört folgendes SPICE-Kommando:

.dc V1 0 15V 10mV V2 -5V 0V 100mV

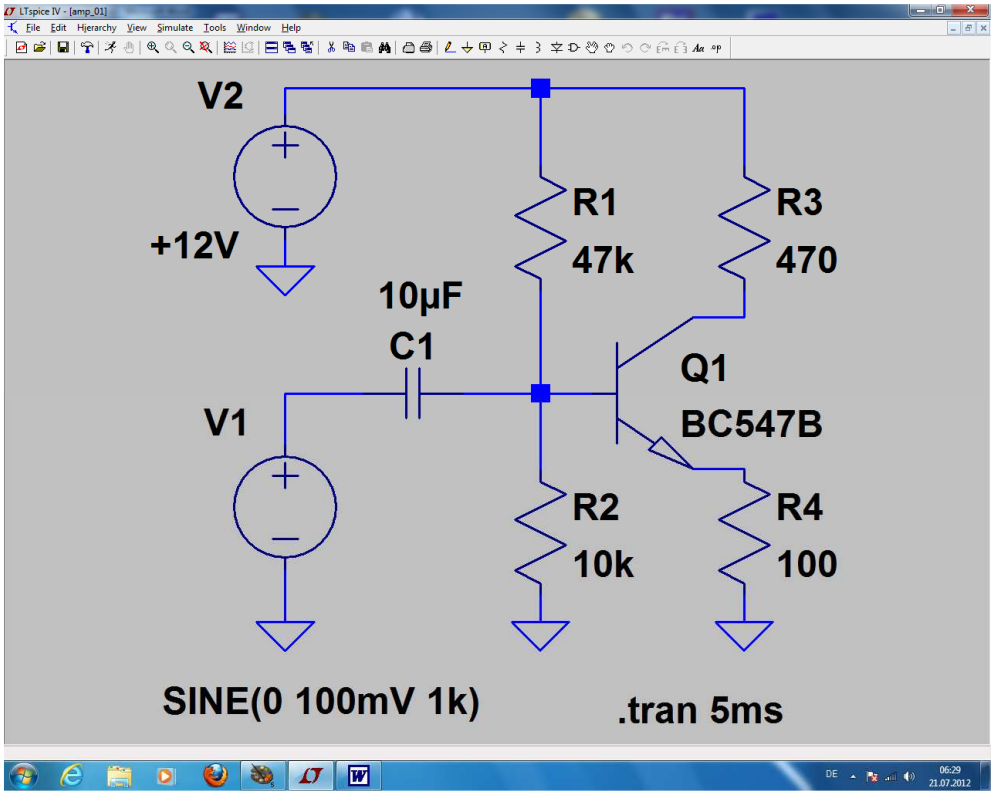
Ergebnis:



11. Füntes Projekt: Schaltungen mit Transistoren

11.1. Einstufiger Verstärker

11.1.1. Ansteuerung mit einem Sinus-Signal



Dazu brauchen wir eine Speisespannung, eine Signalquelle, einen Kondensator, vier Widerstände und einen npn-Transistor.

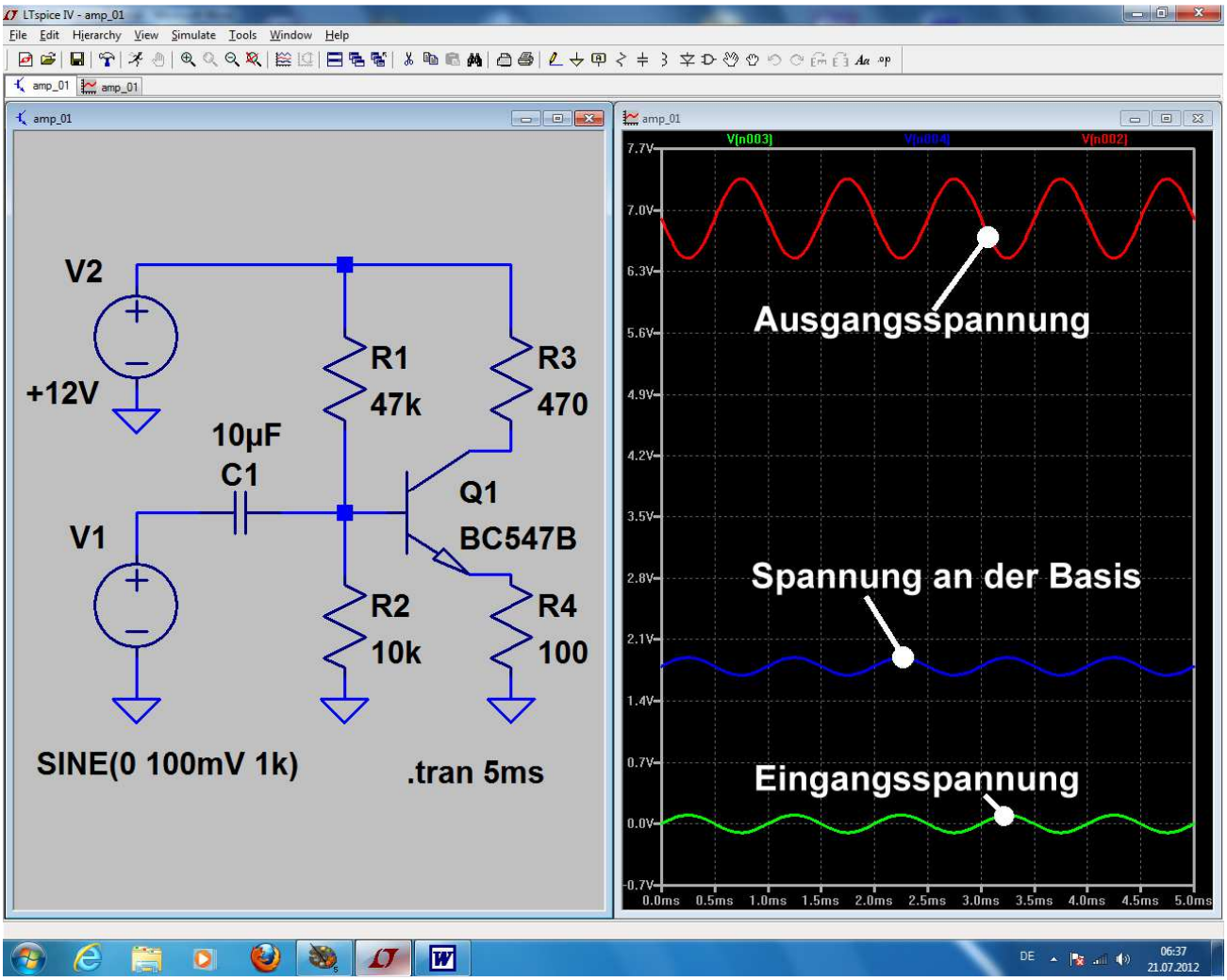
Anleitung:

1. Schritt:
4 Widerstände, 1 Kondensator, 2 Spannungsquellen („voltage“) und 1 npn-Transistor („npn“) werden platziert und verdrahtet.

2. Schritt:
Den Widerständen und dem Kondensator werden die korrekten Werte zugewiesen. Dann klicken wir mit der rechten Maustaste auf das Transistorsymbol, gehen in das Menü „Pick New Transistor“ und holen uns den **BC547B.**
Die obere Spannungsquelle V2 ist vom Typ „DC“ und liefert +12V.
Die untere Spannungsquelle V1 erzeugt einen Sinus mit 100mV Spitzenwert und der Frequenz f = 1kHz.

3. Schritt:
Es wird ein Simulationszeitraum von 0….5 Millisekunden vorgesehen und anschließend die Simulation gestartet.

Beim Simulationsergebnis schauen wir uns die Eingangsspannung, die Basisspannung des Transistors und seine Kollektorspannung an.

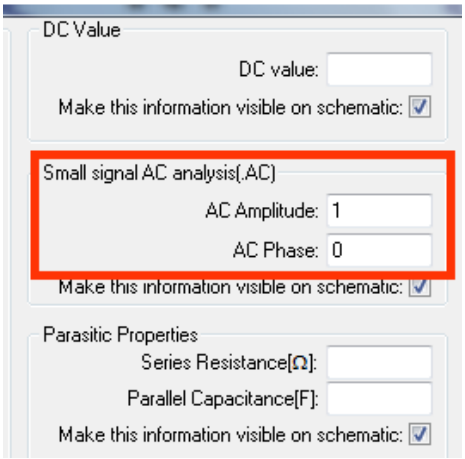


Besonders die Phasenumkehr im Transistor können wir nun sehr schön sehen.
Außerdem lässt sich nun die Behauptung der Theorie, das die Verstärkung der Stufe etwa den Wert

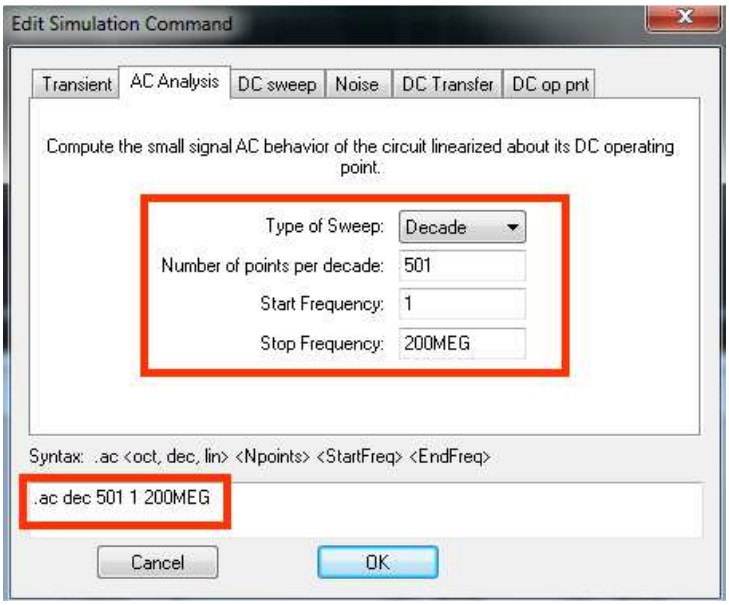
$$V_U = RC / RE = 470 / 100 = 4{,}7\text{fach}$$

habe, leicht überprüfen. (Dazu braucht man nur die Ausgangsspannung alleine darstellen und ihren Spitzenwert bestimmen. Angesteuert wird die Stufe ja mit einem Spitzenwert von 100mV)

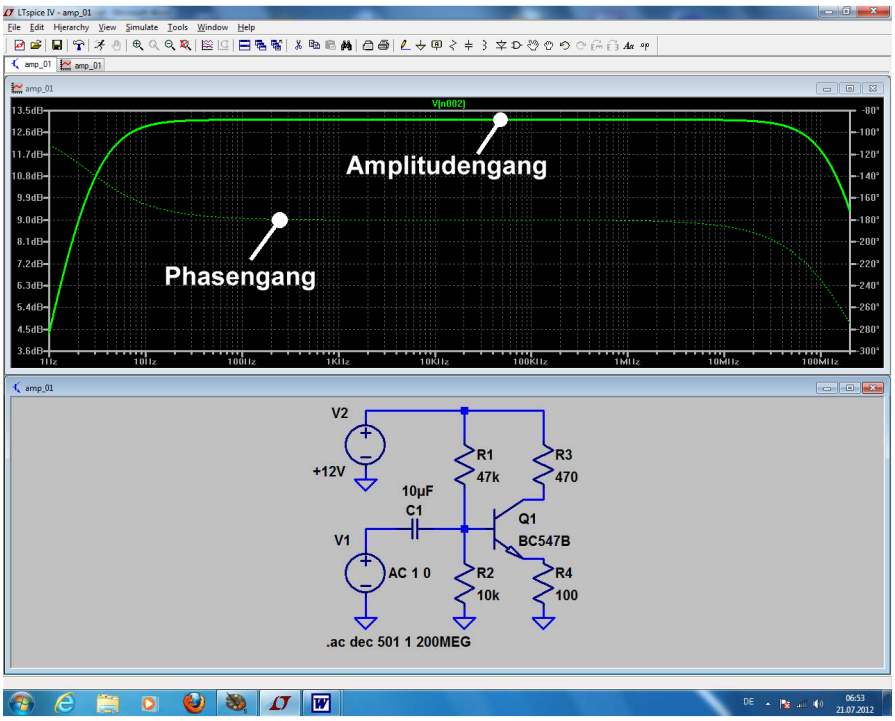
11.1.2. Simulation des Frequenzgangs (= “AC-Sweep”)



1. Schritt:
Wir stellen die Spannungsquelle V1 auf “**Sweep-Betrieb**” um.



2. Schritt:
Über „**Simulate**“ und „**Edit Simulation Command**“ kommen wir an das nebenstehende Menü heran. Auf der Karteikarte „**AC Analysis**“ programmieren wir einen dekadischen Sweep von 1Hz bis 200MHz und wählen 501 Punkte pro Dekade.



So sollte das Simulationsergebnis schließlich aussehen.

11.2. Zweistufiger gegengekoppelter Breitbandverstärker

11.2.1. Pflichtenheft

Es soll ein „Gainblock“ für ein 75 Ohm – Kommunikationssystem entwickelt werden. Das bedeutet:

a) Eingangs- und Innenwiderstand sollen je 75 Ohm betragen, damit beim Zusammenschalten mehrerer Bausteine immer Leistungsanpassung herrscht und dazwischen 75 Ohm – Koaxialkabel (Kabeltyp: RG59) verwendet werden können.

b) Wird der Ausgang mit 75 Ohm belastet, dann soll die sich dort einstellende Ausgangsspannung um den Faktor 2 (das entspricht 6 dB) größer sein als die Eingangsspannung.

c) Als Versorgungsspannung ist +12V vorzusehen.

d) Die Verstärkung soll im Frequenzbereich von 1 kHz bis 10 MHz konstant sein.

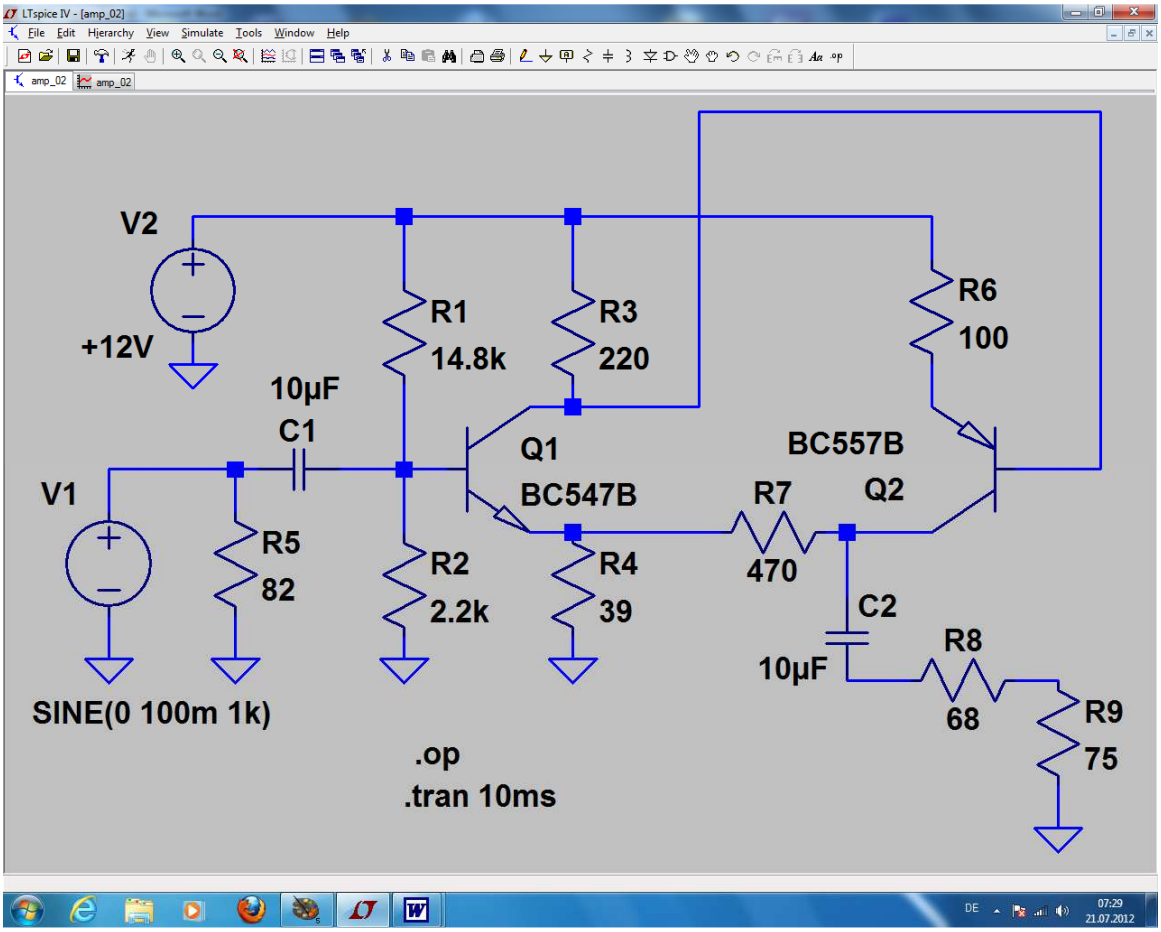
e) Es wird eine zweistufige Verstärkerschaltung mit npn- und pnp-Transistor (= BC547B und BC557B) sowie Gegenkopplung eingesetzt.

f) Zum Ausgleich von Exemplarstreuungen und zur Einstellung des richtigen Arbeitspunktes soll ein Einstellpotentiometer im Basiskreis des ersten Transistors vorhanden sein.

11.2.2. Simulations-Schaltung und Simulations-Vorgaben

Diese Schaltung wird zuerst gezeichnet, wobei eine **Sache ganz wichtig ist:**

Bei den Widerständen R1 = 14,8k und R2 = 2,2k darf man **nur ein Punkt, aber kein KOMMA IN DER WERTANGABE** benützen, da sonst mit falschen Werten simuliert wird -- und das ohne jede Warnung!



Den **pnp-Transistor BC557B** finden wir wieder als „pnp“ im Teilevorrat. Nach einem Rechtsklick auf sein Schaltzeichen kann der Typ „**BC557B**“ zugewiesen werden**. Bitte aber nicht vergessen, dieses Schaltzeichen nach dem Absetzen um 180 Grad zu drehen!**

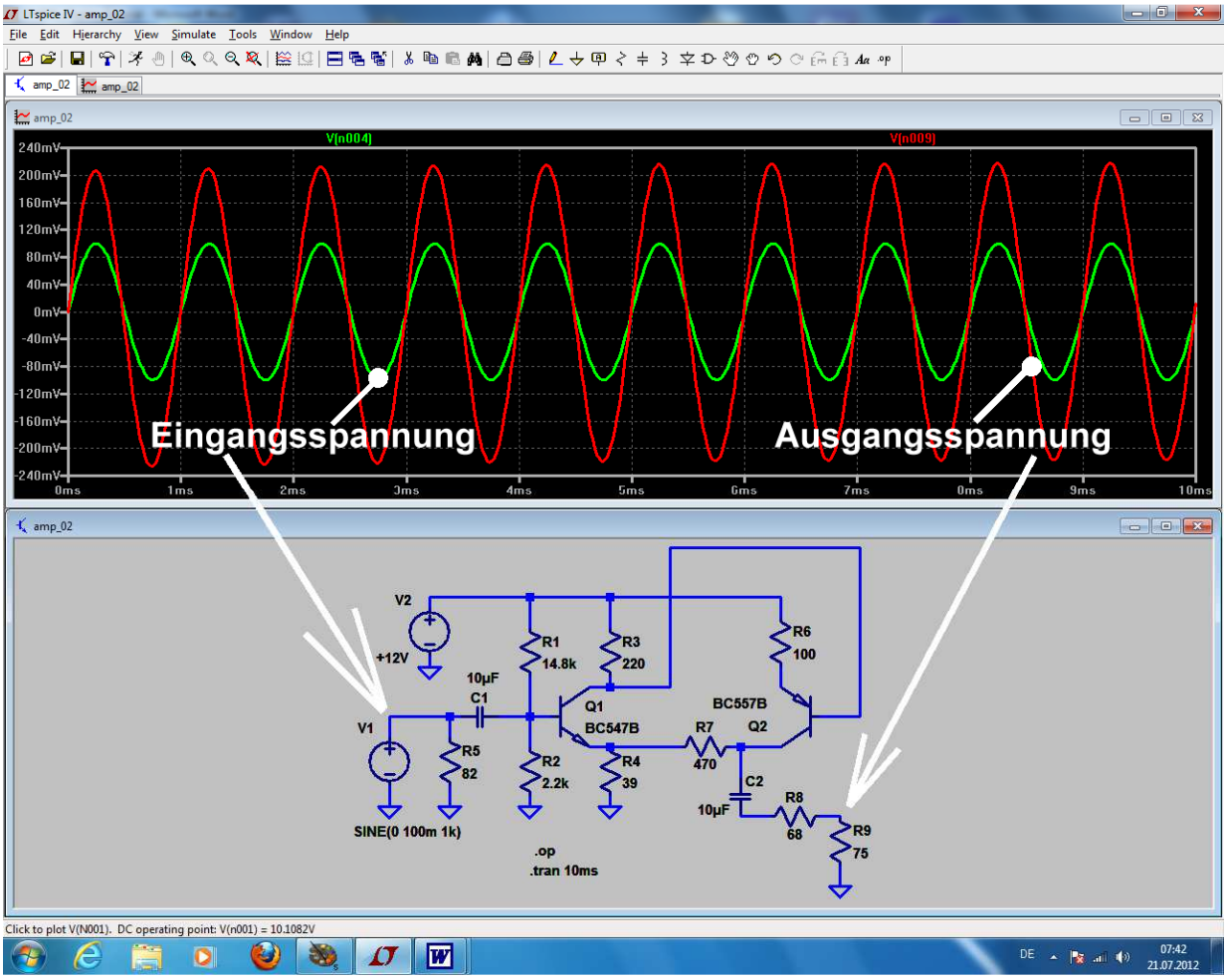
Angesteuert wird die Schaltung im ersten Durchgang mit einer 1kHz-Sinusspannung, die einen Spitzenwert von 100mV aufweist. Simuliert wird von 0…10 Millisekunden.

Neu ist die Simulationsanweisung **„.op“**

Sie sorgt dafür, dass der DC-Arbeitspunkt („Operation point“) zugänglich gemacht wird -- und zwar in Form einer Liste, die alle vorkommenden Gleichspannungen und Gleichströme des Ruhezustandes enthält.

Gemessen wird anschließend die Eingangsspannung der Quelle V1 und die Ausgangsspannung am Abschlusswiderstand R9 (75Ω)

11.2.3. Simulation in der Time Domain (= im Zeitbereich)



Man sieht, dass tatsächlich die Ausgangsspannung etwa doppelt so groß ist wie die Eingangsspannung (was einer Spannungsverstärkung von 6dB entspricht). Und zwei Emitterstufen bringen doppelte Phasenumkehr -- also muss das Ausgangssignal wieder gleichphasig mit dem Eingangssignal sein…

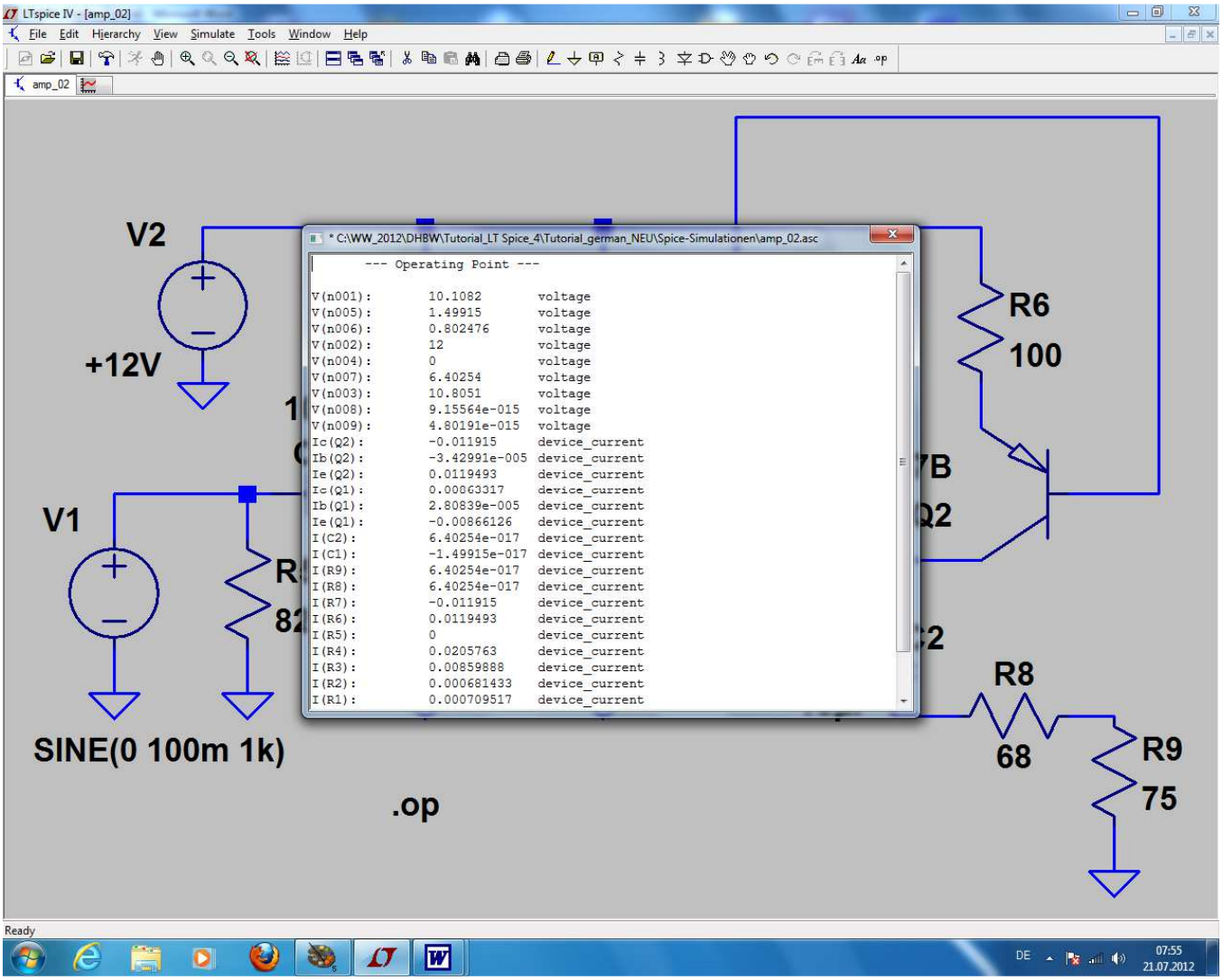
11.2.4. DC-Bias (= Gleichstrom-Analyse)

Wenn man sich für die Ruheströme und –Spannungen im gewählten Arbeitspunkt interessiert, dann geht man so vor:

Das Menü „**Simulation**“ wird geöffnet und darin „**Edit Simulation Command**“ angeklickt. Dann wechseln wir auf die letzte Karteikarte **„DC op point**“. Im Eingabefenster muss dort **„.op“** stehen.

Das bestätigen wir mit OK und prüfen, ob das Kommando „.op“ nun korrekt am Cursor hängt und im Simulationsschaltbild abgesetzt werden kann. (**Wichtige Kontrolle: vor dem vorhin verwendeten Kommando „.tran 10ms“ muss plötzlich ein Strichpunkt stehen, um es unwirksam zu machen**).

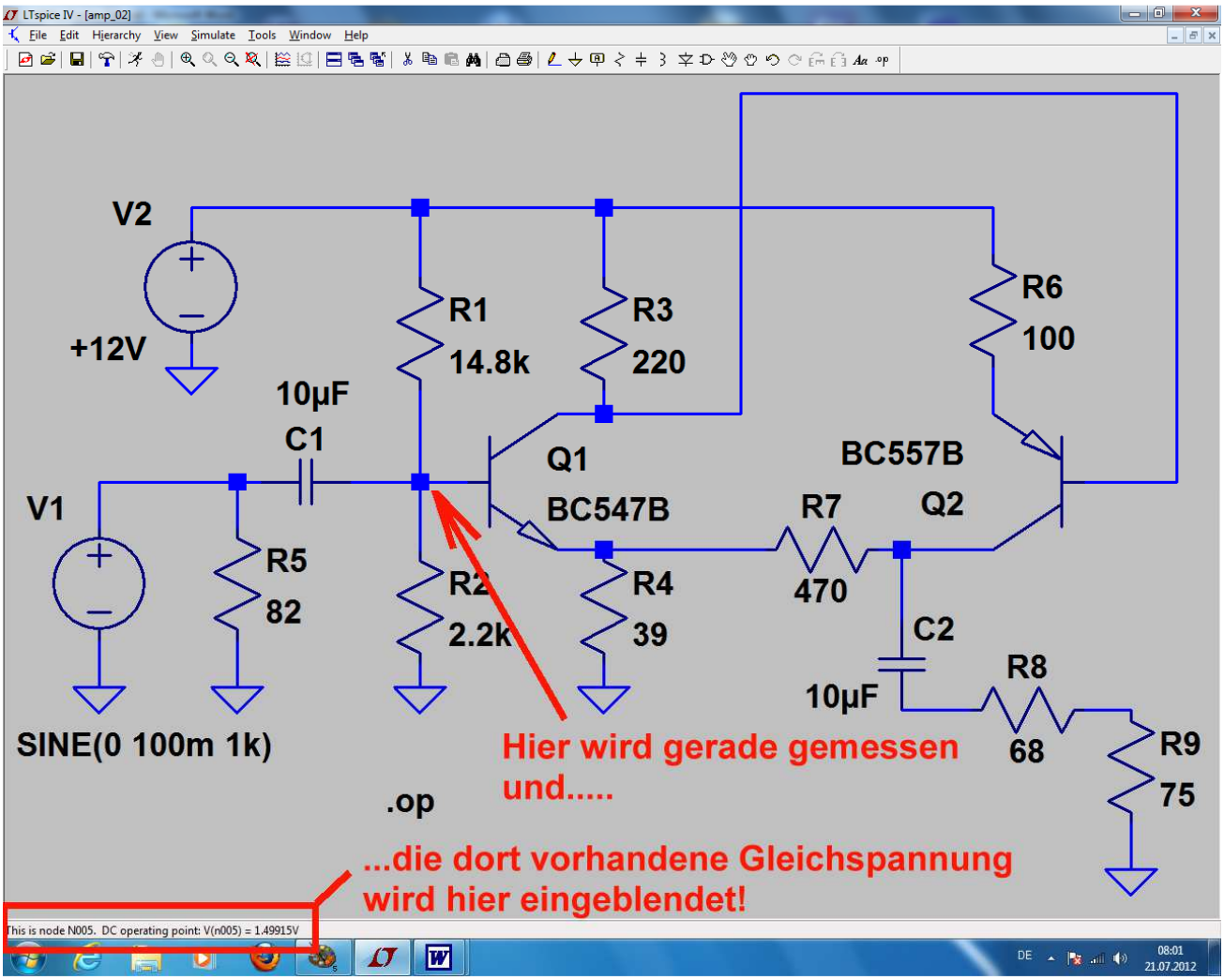
Jetzt dürfen wir wieder mal auf das „rennende Männchen“ (= Simulationsbutton) drücken und auf das Ergebnis warten.



Die Tabelle wird nach Betrachtung wieder geschlossen und dadurch das komplette Schaltbild erneut sichtbar gemacht.

Und da gibt es noch eine weitere hübsche Möglichkeit:

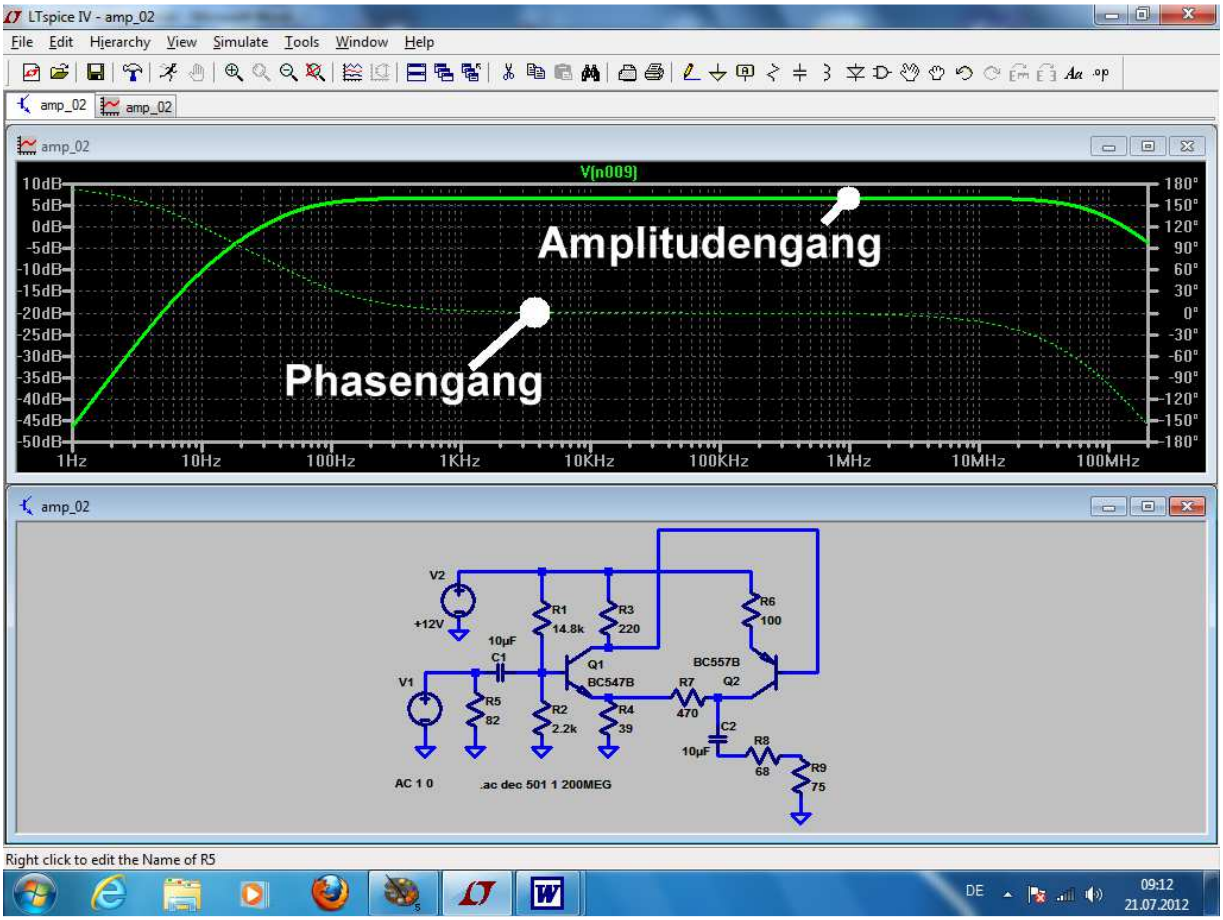
Man kann sich nämlich nun jede einzelne Spannung (oder jeden einzelnen Strom…) direkt anzeigen lassen!



Wichtig:
Möchte man wieder zur Transientensimulation (= Ansteuerung mit einer Sinusspannung) zurück, dann löscht man das Kommando „op“ und sorgt für eine neue gültige „.tran…“ – Anweisung

11.2.5. AC-Sweep (= Frequenzgang von 1 Hz bis 200 MHz)

Das sollte nun wirklich schnell gehen:
Bei der Spannungsquelle unter „Small Signal AC Analysis“ die beiden Einträge („1“ für die AC-Amplitude, „0“ für die AC-Phase) vornehmen und ein neues Simulation Command für den AC-Sweep von 1Hz bis 200MHz mit 501 Punkten pro Dekade erstellen.



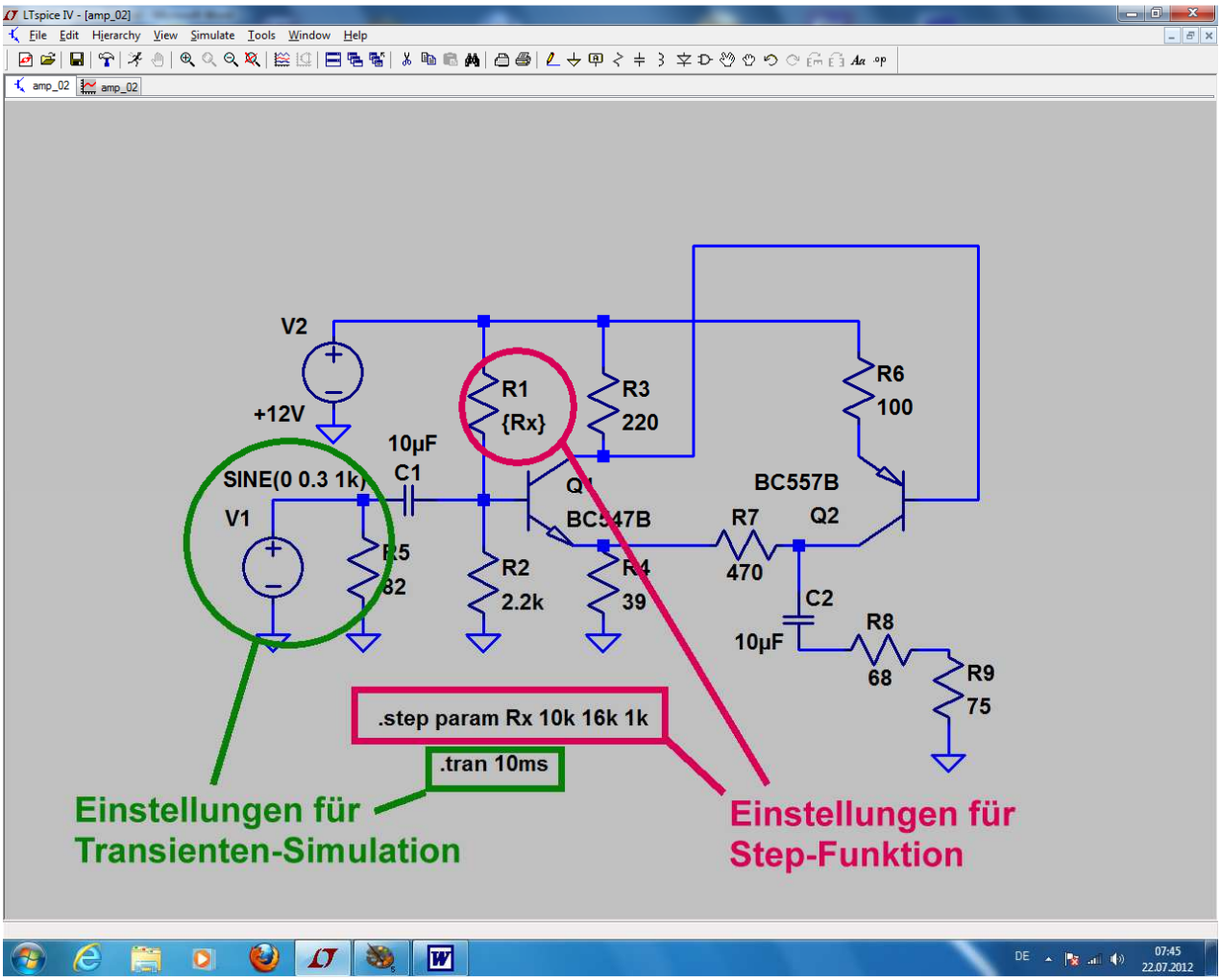
11.3. Der Parameter-Sweep

Wer in der Online-Hilfe mal unter **„LT SPICE / Dot Commands**“ nachschaut, findet eine sehr lange Liste an zusätzlichen Möglichkeiten und Befehlen.

Wir wollen uns daraus das Kommando **.step** etwas näher ansehen.

Das ist eine ganz wichtige Sache, denn damit können wir das „**Ausprobieren von verschiedenen Werten“ auf eine wesentlich exaktere und bequemere Grundlage stellen** -- egal ob das die Eingangsspannung, die Speisespannung oder einen Bauteil- oder Modellwert innerhalb der Schaltung betrifft.

Wir nehmen als Beispiel nochmals **Kapitel 9.2.3**. her **und lassen uns den oberen Basis-Spannungsteiler-Widerstand R1 in 5 Schritten von 10k bis 16k variieren.** Die Ergebnisse können dann direkt verglichen werden und so hat man sehr schnell die korrekte Potentiometer-Einstellung.



Bitte genau hinschauen:

a) Die Amplitude der 1kHz-Eingangsspannung wurde **auf 0,3V erhöht**

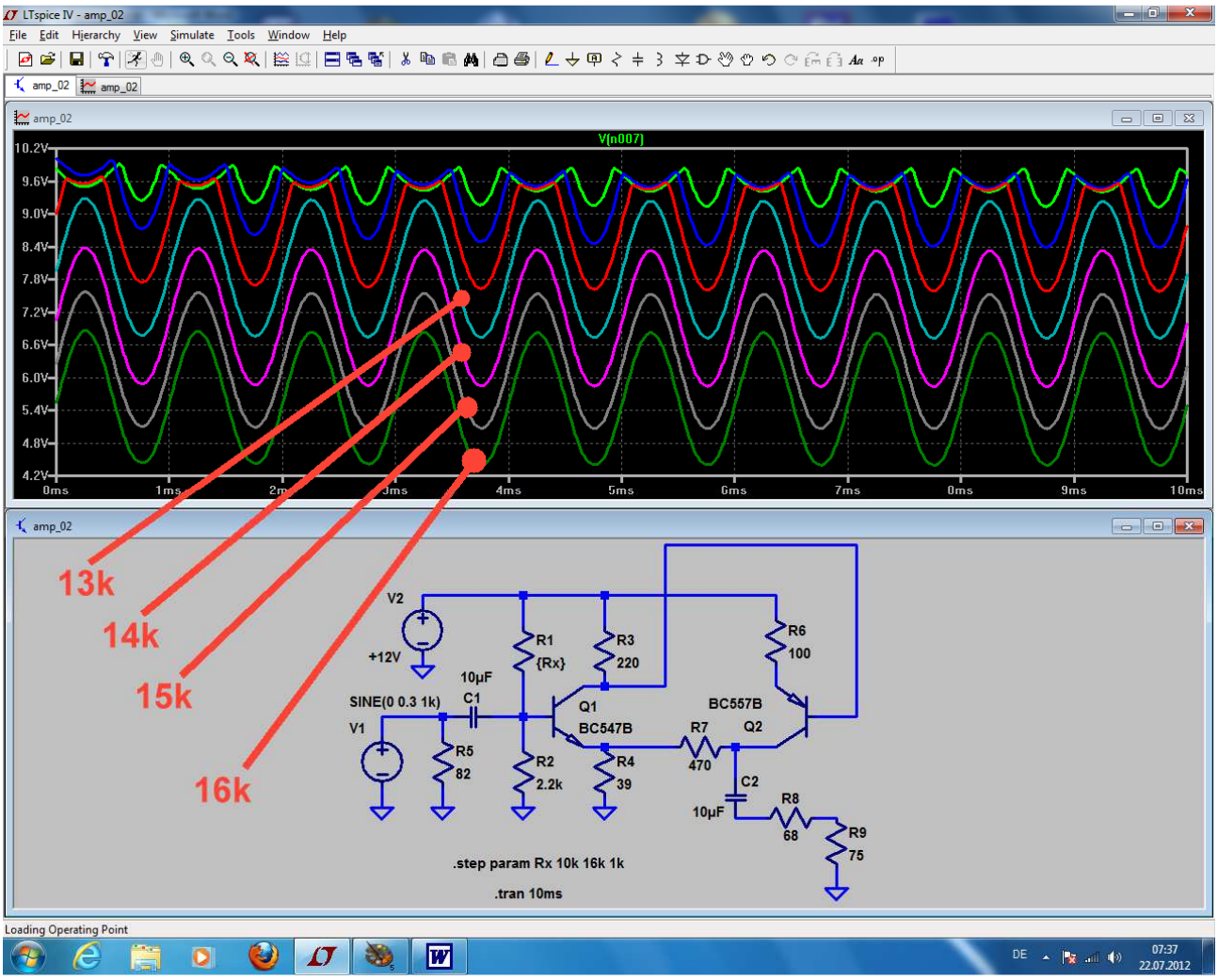
b) Beim **Widerstand R1** wurde die Wertangabe durch den Parameter **{Rx}** ersetzt

c) Schließlich ist noch die **Spice-Direktive** **.step param Rx 10k 16k 1k**

erforderlich. Dadurch wird R1 in Schritten von 1 Kilo-Ohm von 10k bis 16k verändert.

Drückt man nun die „Taste mit dem Männchen“, dann wird alles simuliert, aber noch nichts im Ergebnisdiagramm ausgegeben. Dazu müssen wir z. B. erst den Kollektor von Transistor Q2 anklicken, um die dortige Spannung beobachten zu können.

So sieht das Ergebnis dann aus und man kann nun leicht abschätzen, dass ein Wert von etwa 15k bei R1 (= zweite Kurve von untern) erforderlich ist, um eine mittlere Kollektorgleichspannung von ca. 6,5V zu erhalten.



(Hier noch ein Tipp:
Wenn man nicht weiß, welche Kurvenfarbe zu welchem Widerstandswert gehört, dann klicke man mal auf „den Button mit dem Hammer“, gefolgt von „Waveforms“ und „Color Scheme“. Da findet man die Reihenfolge der Farben bei den nacheinander geschriebenen Kurven und kann sie sogar nach Wunsch verändern.)

12. Sechstes Projekt: OPV-Schaltungen

12.1. Wichtige Vorbemerkungen

Will man nur wissen, ob eine eigene Idee grundsätzlich funktioniert, dann reicht die Simulation mit einem **idealen OPV-Modell** ohne Versorgungsspannungen. Wir finden es in der Bibliothek im Ordner „**Opamps**“ als „**opamp**“.

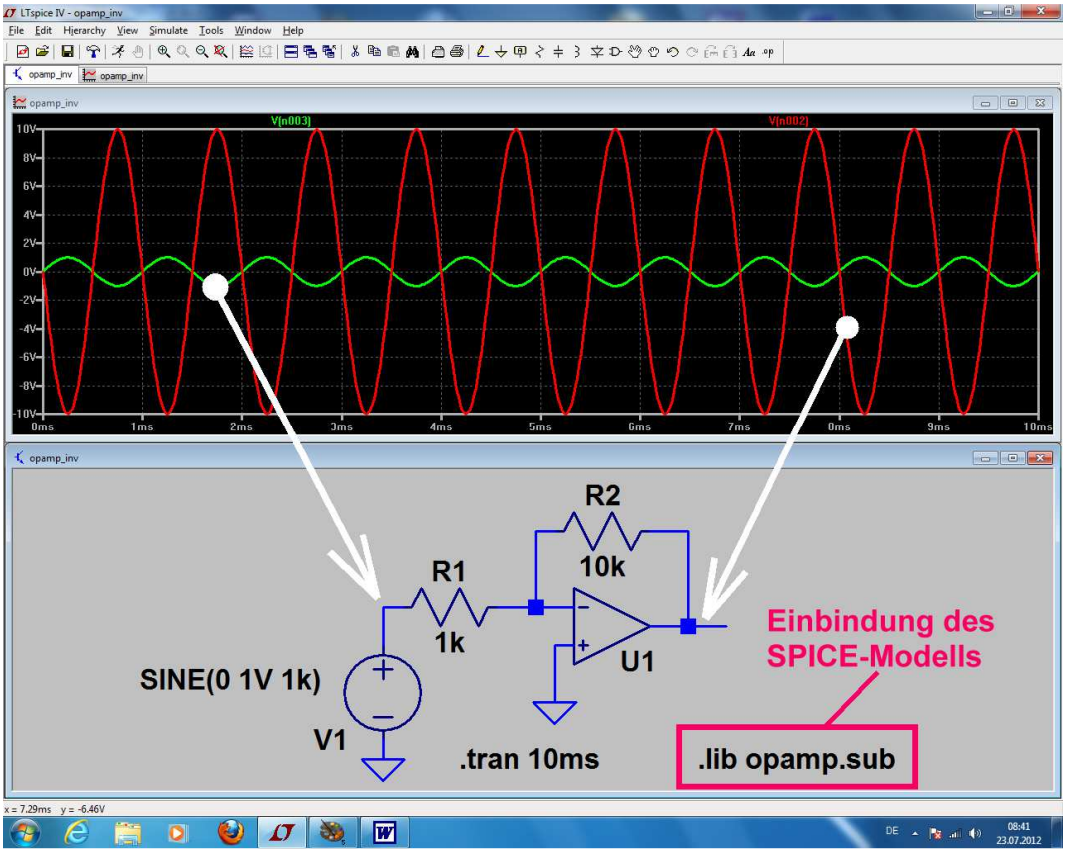
Sobald die Ansprüche an die Schaltung steigen und noch keine Entscheidung für einen bestimmten OPV-Typ gefallen ist, kann man den „**UniversalOpamp2**“ einsetzen. Er bietet 4 verschiedene Qualitätsstufen für die Praxis und arbeitet mit 2 Betriebsspannungen. Die einstellbaren Qualitätsstufen sind im mitgelieferten Beispiel „**UniversalOpamp2.asc**“ genau beschrieben.

Wird jedoch gefordert, unbedingt einen **bestimmten OPV-Typ** einzusetzen, dann sieht man zunächst im der **Bauteil-Bibliothek** nach. Dort gibt es nämlich einen großen Vorrat an Modellen für das gesamte Linear Technology – Programm.
Reicht das nicht, dann bleiben nur die Suche im Internet nach einem passenden SPICE-File und die Erstellung eines eigenen Bauteils samt Symbol.

12.2. Simulationen mit dem idealen OPV-Modell

12.2.1. Umkehrender, idealer Verstärker mit 10facher Verstärkung

Wir zeichnen die Schaltung, wobei der „opamp“ im Unterverzeichnis „Opamps“ innerhalb der Bauteilbibliothek zu finden ist. Gespeist wird der Eingang mit einer Sinusspannung (Spitzenwert = 1V / Frequenz = 1kHz). Allerdings darf man die **Spice-Direktive zur Einbindung der Modelldatei** nicht vergessen…



Bitte genau hinsehen und merken:

a) Direkt am „Minus-Eingang“ des OPVs herrscht nahezu eine Spannung von Null Volt (…genauer: eine winzige Eingangsspannung, die man über „Ausgangsspannung geteilt durch Leerlauf-Verstärkung“ berechnen kann). Also sieht die ansteuernde Quelle V1 nur den Widerstand R1, dessen anderes Bein folglich auf Massepotential liegt.

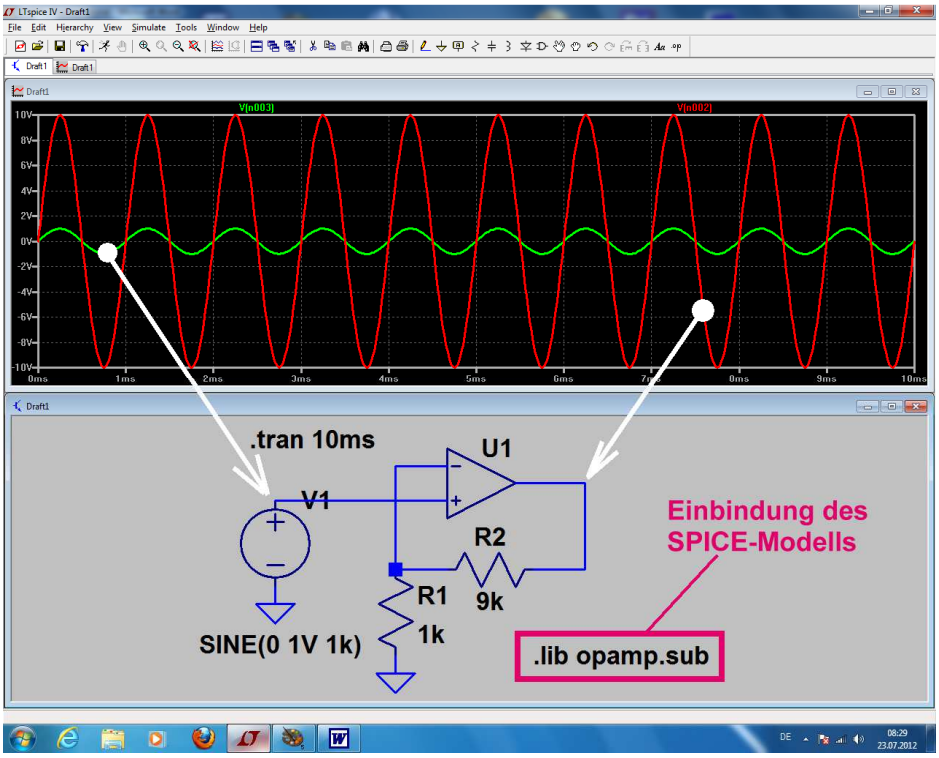
Damit stellt R1 = 1kΩ direkt den Eingangswiderstand der Schaltung dar!

b) Der Strom durch R1 kann aber nicht in den hochohmigen Eingang des OPVs hineinfließen, sondern nimmt seinen Weg über R2 und erzeugt dort nach dem Ohm‘schen Gesetz eine Spannung, die gleichzeitigt die Ausgangsspannung darstellt.

Also erhält man die Verstärkung der Stufe zu $V = -R2 / R1$

Das Minuszeichen weist darauf hin, dass die Ausgangsspannung gegenphasig ist.

12.2.2. Nicht umkehrender, idealer Verstärker



Gefordert wird wieder 10fache Verstärkung, aber der Einsatz einer nicht umkehrenden Schaltung.

Dadurch erreicht man einen sehr hohen Eingangs-Widerstand, denn der OPV-Eingang benötigt praktisch keinen Eingangsstrom.

Auch die Verstärkung ist einfach zu bestimmen, denn man muss nur davon ausgehen, dass im Dauerbetrieb an beiden OPV-Eingängen dieselbe Spannung (…mit der Amplitude und dem Verlauf der Eingangsspannung) beobachtet werden kann.
Die Spannung am invertierenden Eingang wird aber von einem Spannungsteiler geliefert, der die Ausgangsspannung herunterteilt. Damit erhält man als Verstärkung „den Kehrwert des Teilungsfaktors“:

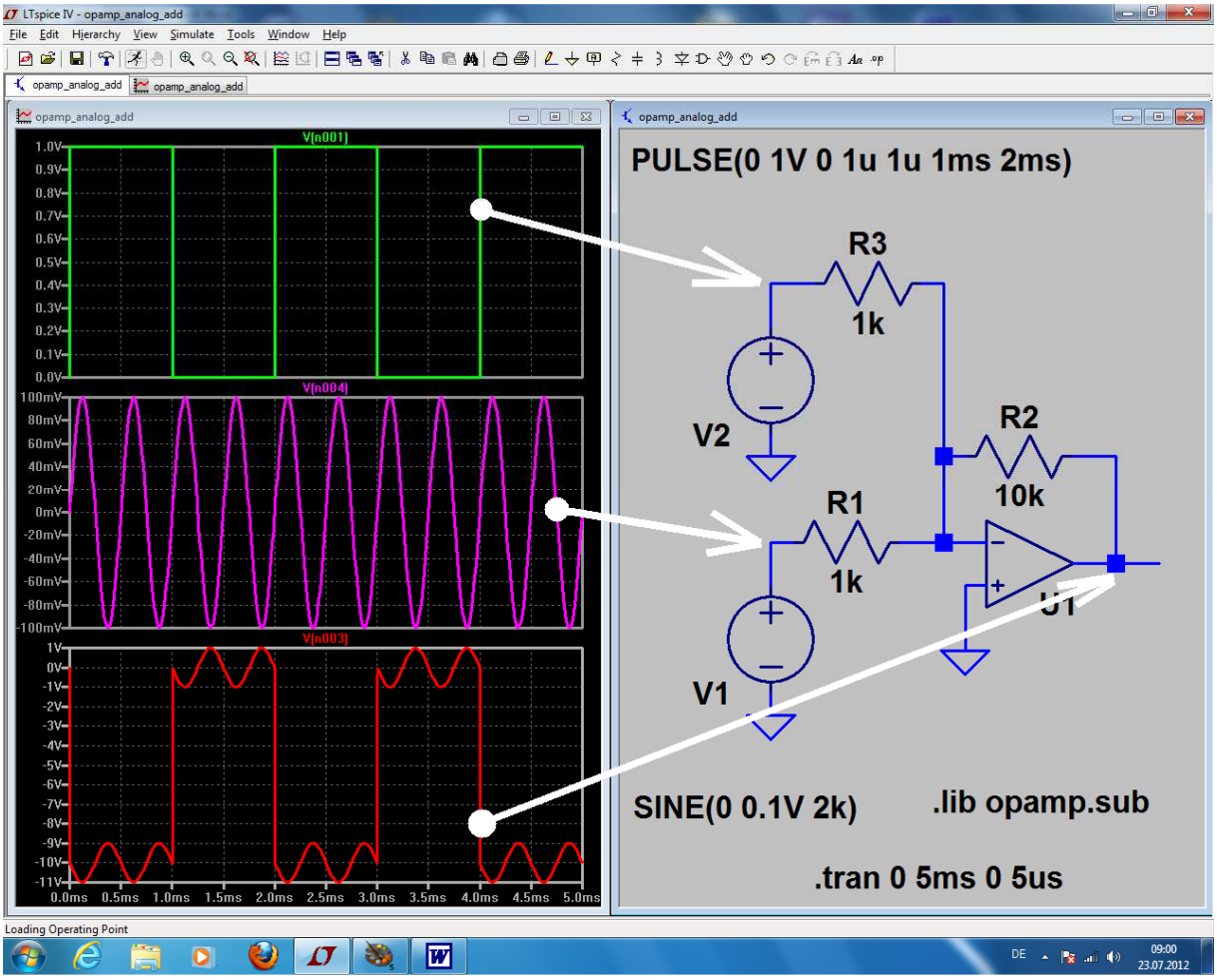
$$V = (R1 + R2) / R1 = 1 + (R2 / R1)$$

12.2.3. Idealer Analog-Addierer

Mit OPVs lassen sich bekanntlich analoge Signale sehr leicht zusammenaddieren. Wir wollen das mit einem Sinus (f = 2kHz, Spitzenwert = 0,1V) und einem Rechtecksignal (f = 500Hz, Umin = 0V, Umax = 1V) ausprobieren und fordern zusätzlich:

a) eine 10fache Verstärkung und
b) einen Eingangswiderstand von 1kΩ bei jedem Eingangskanal.

Dazu dient eine umkehrende Schaltung, bei der der Invertierende Eingang des OPVs einen „Virtuellen Nullpunkt“ bildet.



12.2.4. Aktive Filterschaltung

Für Filterzwecke (nicht nur im Audiobereich!) werden immer häufiger **die aktiven Filter** den passiven Filtern vorgezogen, da sie folgende Vorteile aufweisen:

-- Verwendung von Operationsverstärkern, die nur mit R und C gegengekoppelt sind
-- viel billiger, da keine (teuren) Spulen nötig sind und alle Abgleicharbeiten wegfallen
-- als SMD - Version oder IC sehr kleine Abmessungen
-- Verstärkung in weiten Grenzen einstellbar
-- bei starker Gegenkopplung und kleiner Verstärkung fast keine nichtlinearen Verzerrungen
-- wegen der Fortschritte in der OPV - Technik nun bis ca. 100MHz realisierbar

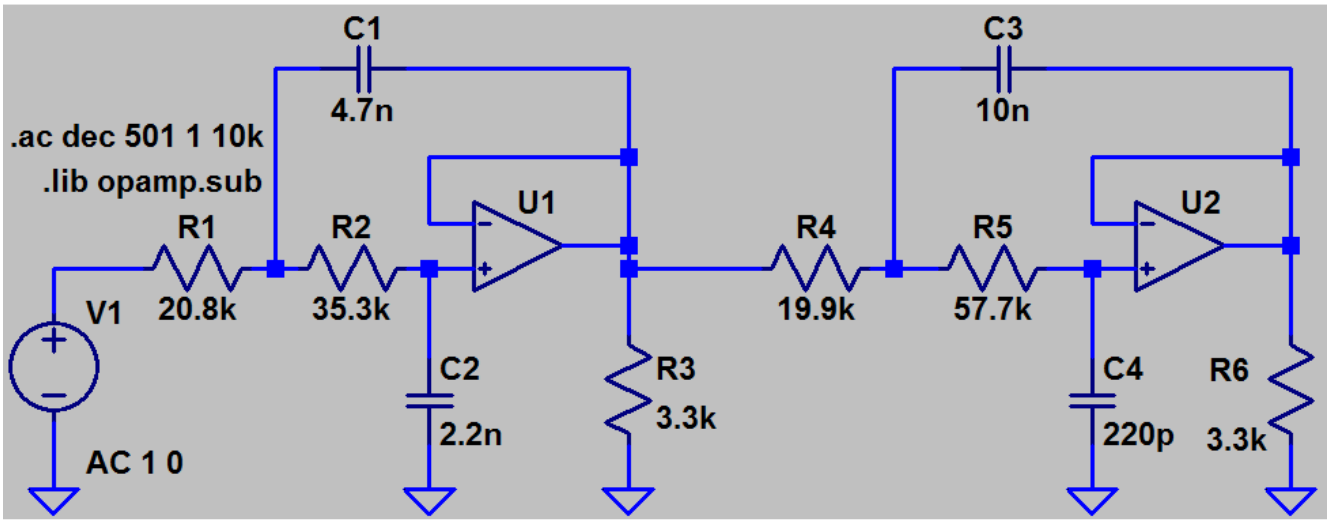
Die erforderlichen theoretischen Grundlagen und alle Berechnungshinweise finden sich z. B. im Standardwerk „Halbleiter-Schaltungstechnik“ von Tietze-Schenk (= Springer-Verlag). In der Zwischenzeit gibt es auch viele Tutorials und Shareware- oder Freeware-Entwurfsprogramme im INTERNET (z. B. in der Homepage von Burr - Brown oder von Microchip oder von Texas Instruments).

Anwendungsbeispiel: Sallen – Key - Tiefpass vierten Grades

Die erforderliche Schaltung und die nötigen Bauteilewerte liefern die erwähnten Filterprogramme nach Eingabe folgender Parameter:

Filterart:	**Lowpass**
Filtertyp:	**Tschebyschef**
Grenzfrequenz:	**3400 Hz**
passband ripple“: (= Maximale Welligkeit der Dämpfung im Durchlassbereich)	**0,5 dB**
Filtergrad: (Er bestimmt die Anzahl der erforderlichen Bauteile und legt fest, wie steil der Übergang vom Durchlass- in den Sperrbereich erfolgt).	**4**

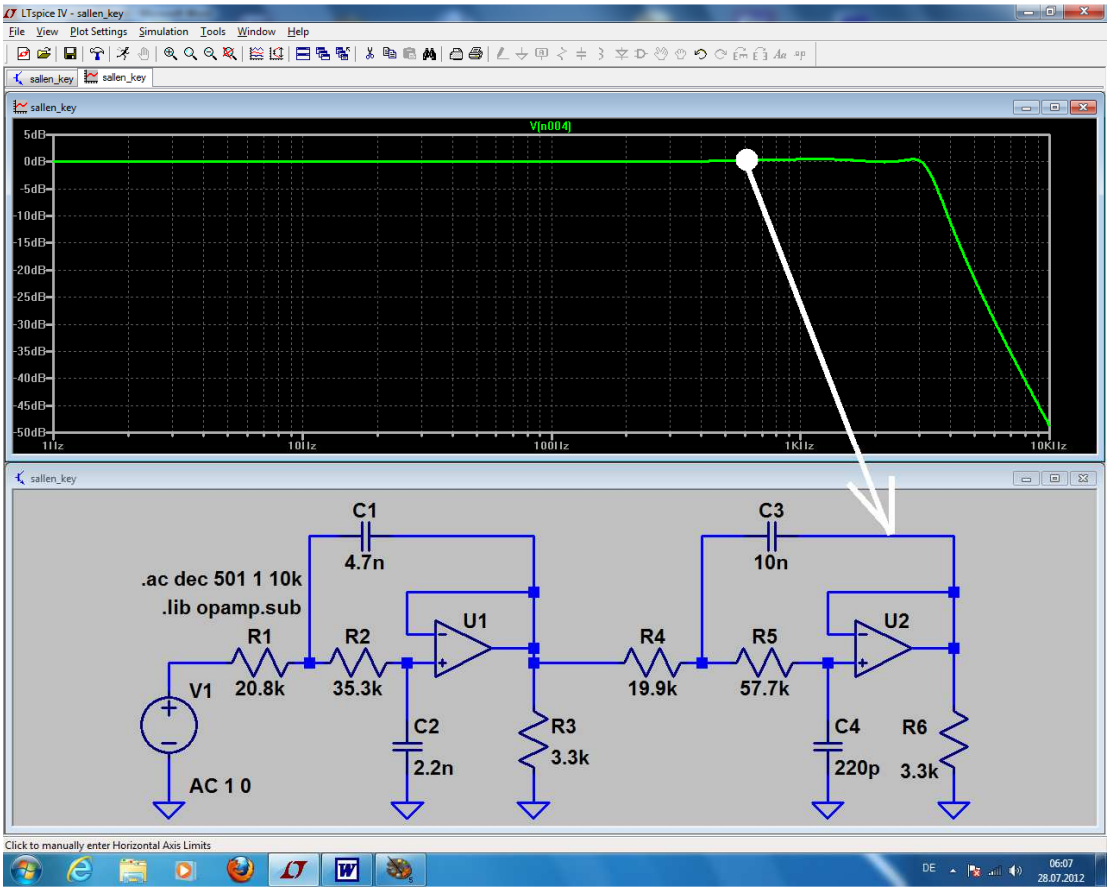
Und so sieht die Schaltung aus, wenn wir zur Simulation den idealen OPV („opamp“) verwenden:



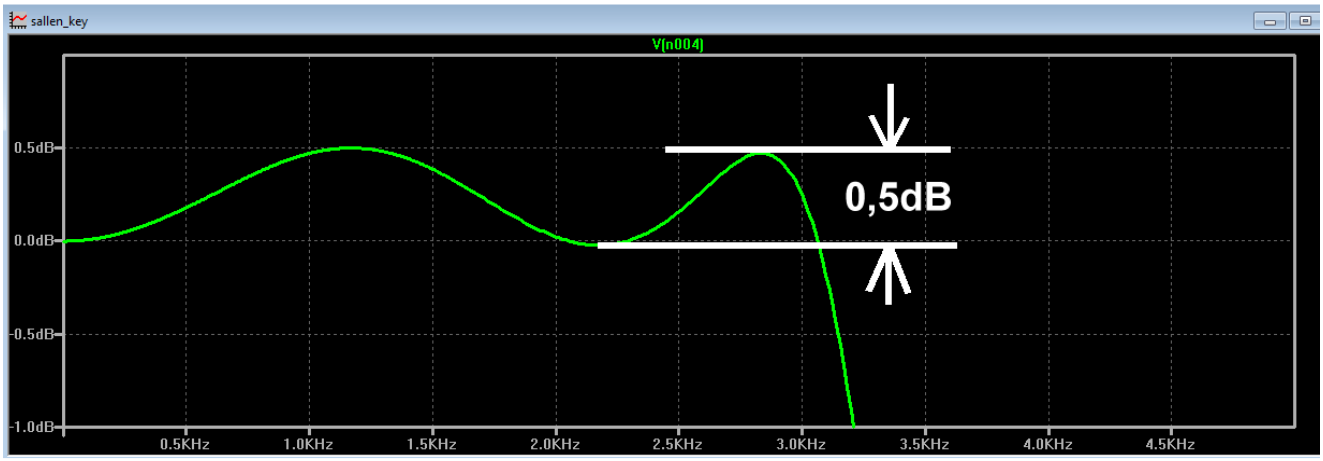
Mit der in das Schaltbild eingetragenen SPICE-Direktive

.ac dec 501 1 10k

führen wir einen dekadischen AC-Sweep von 1Hz bis 10kHz durch, wobei die Auflösung der Frequenzachse 501 Punkte pro Dekade beträgt.
Aber Achtung: **auch die Spannungsquelle am Eingang muss dazu auf „AC 1 0“ umgestellt werden** (= AC-Sweep mit einer Amplitude von 1V und einer Phasenlage von Null Grad).



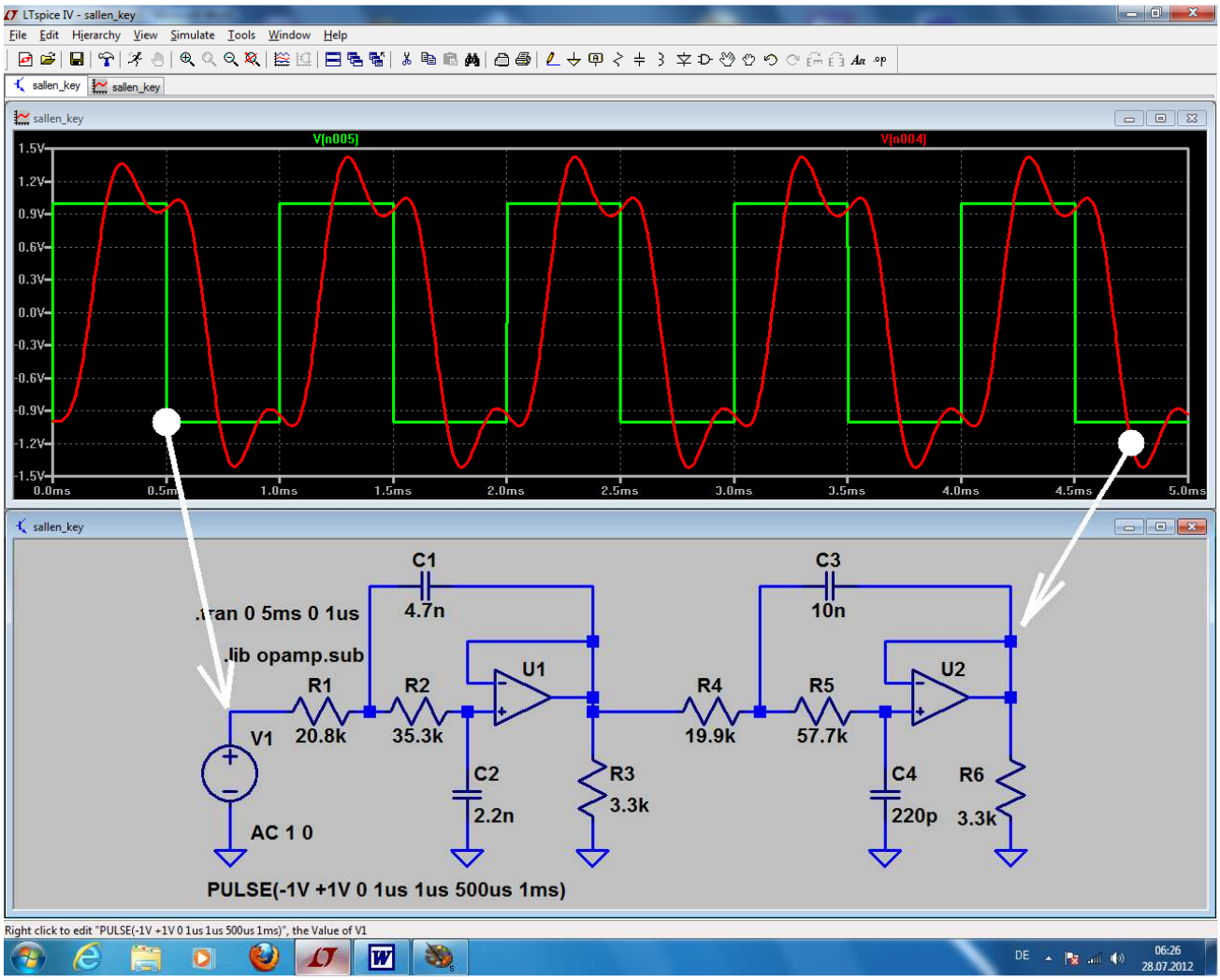
Die „Tschebyschef-Wellen“ sieht man aber erst richtig, wenn man beim Simulationsergebnis die waagrechte Achse auf „linear von 0 bis 5 kHz“ und die senkrechte Achse auf den Bereich „-1dB bis +1 dB“ umstellt:



Nun wollen wir uns an einem praktischen Fall die Filterwirkung der Schaltung ansehen und speisen den Eingang mit einem **Puls-Signal:**

f =	**1kHz**
Negativer Spitzenwert =	**-1V**
Positiver Spitzenwert =	**+1V**
Pulsbreite =	**500µs**
Periodendauer =	**1ms**
Anstiegs- und Abfallzeit je	**1µs**

Simuliert wird von 0….5ms und einem kleinsten Time Step von 1µs:



Man erkennt deutlich, wie das Ausgang der Schaltung

a) die **Oberwellen des Pulses wegen der Filterwirkung gedämpft** sind oder sogar fehlen, aber auch

b) wegen der Phasen- und Gruppenlaufzeitverzerrungen eines Tschebyschef-Filters **Signalverzerrungen und Über**schwinger auftreten.

Die Filterwirkung lässt sich auch sehr schön durch einen Vergleich der Spektren beider Signale über die FFT zeigen. Allerdings ist da zuerst eine zusätzliche SPICE-Direktive

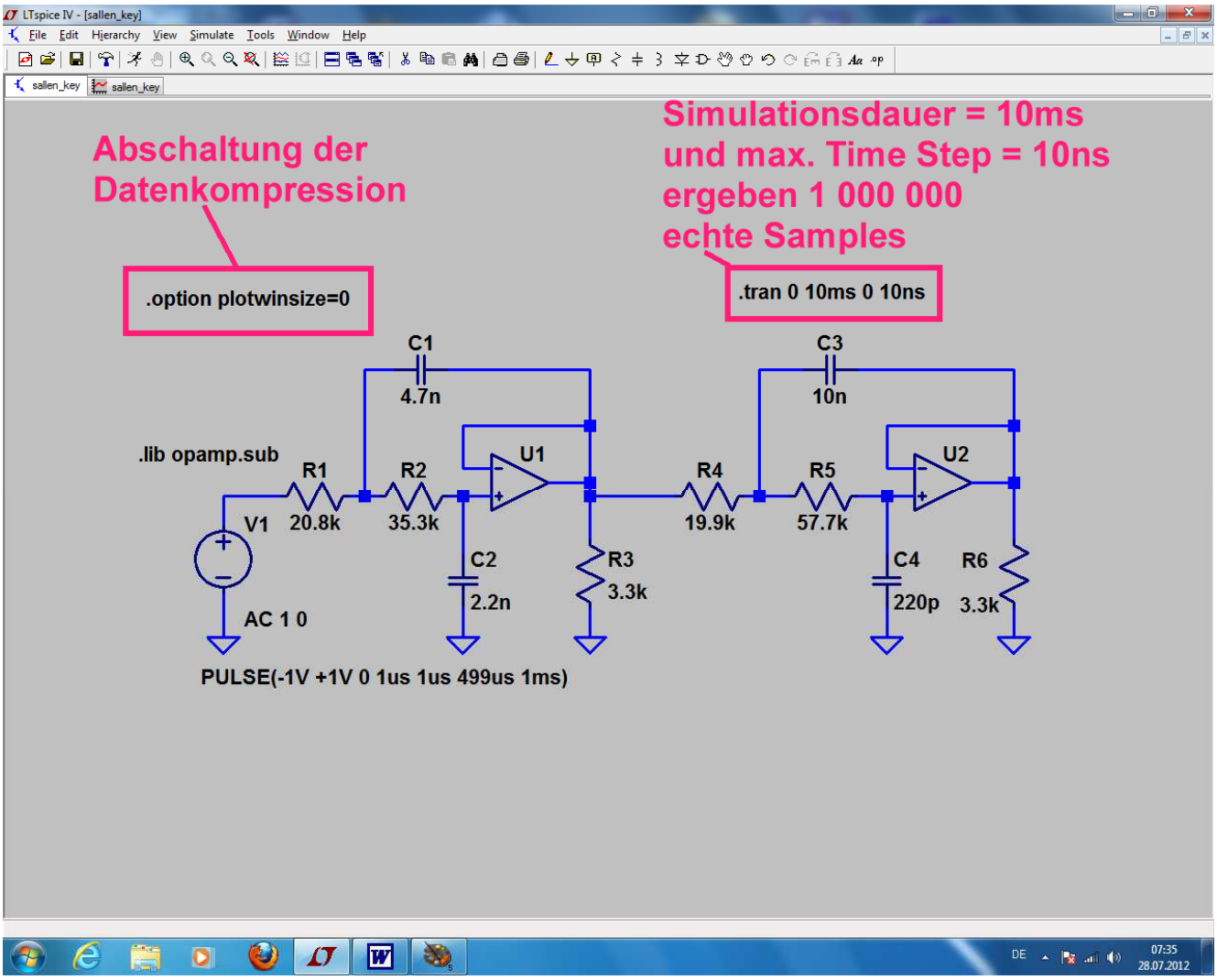
.option plotwinsize=0

zur **Abschaltung der Datenkompression** erforderlich (…sonst entstehen böse Fehler bei der Fast Fourier Transformation, da jede Menge Daten unterdrückt werden). Außerdem benötigen wir möglichst viele Simulationspunkte (Samples) und wählen deshalb folgende Simulationseinstellungen in der Time Domain:

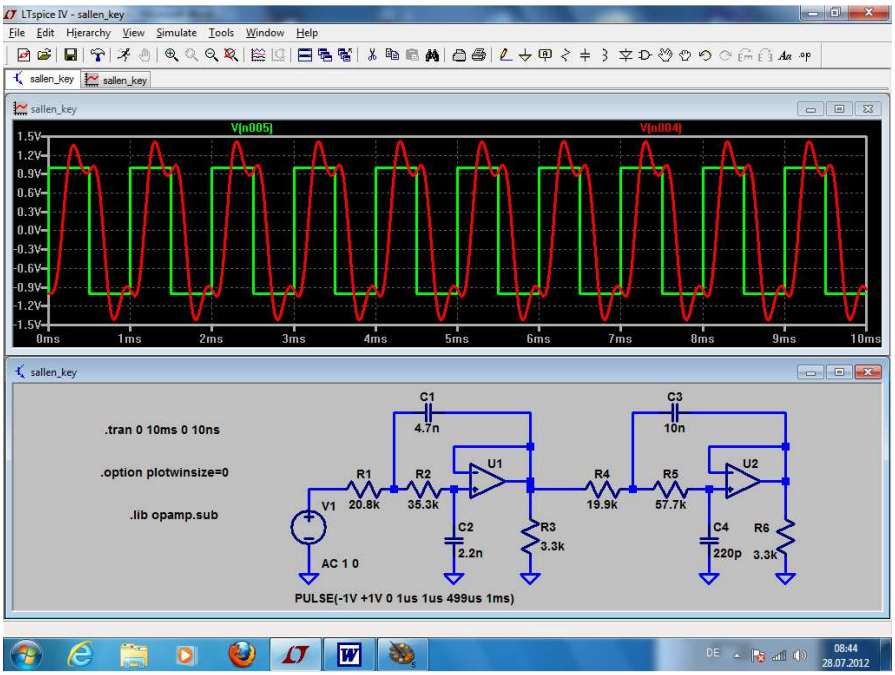
Gesamte Simulationszeit = 10ms.

Das ergibt eine **Frequenzauflösung = Startfrequenz = Linienbreite von 1/10ms = 100Hz**

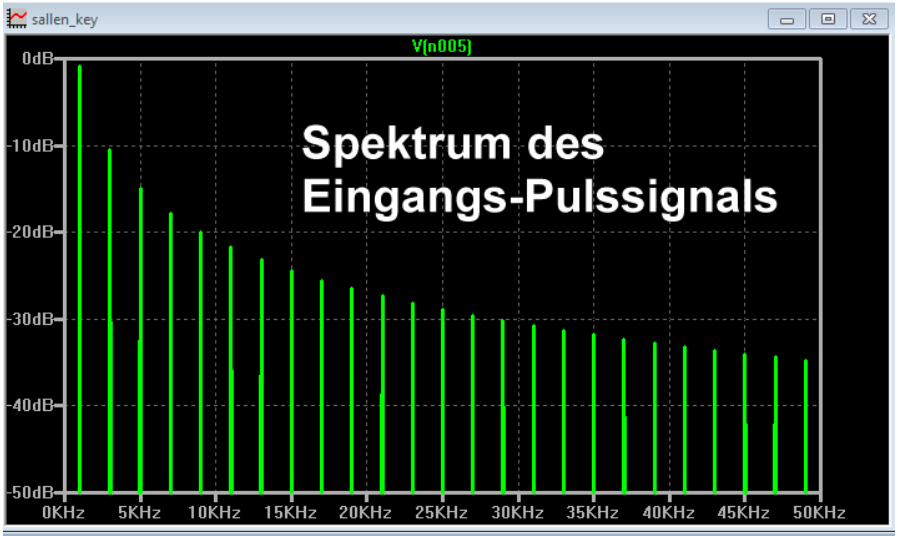
Mit einem maximalen Zeitschritt von 10ns (= maximum Time Step) ergeben sich insgesamt **10ms / 10ns = 1 000 000 Simulationspunkte** (= Samples). Außerdem werden bei einem solch kleinen Zeitschritt die Anstiegs- und Abfallflanken des Pulssignals sicher erfasst und ausreichend abgetastet. Außerdem erhalten wir einen großen dargestellten Frequenz- und Dynamikbereich.



Nach der Simulation in der Time Domain klicken wir mit der rechten Maustaste auf das Diagramm und wählen „View“, gefolgt von „FFT“. Im auftauchenden FFT-Menü schalten wir auf 1 048 576 Samples und wählen zuerst die Eingangsspannung. Anschließend wird der Vorgang für die Ausgangsspannung wiederholt.



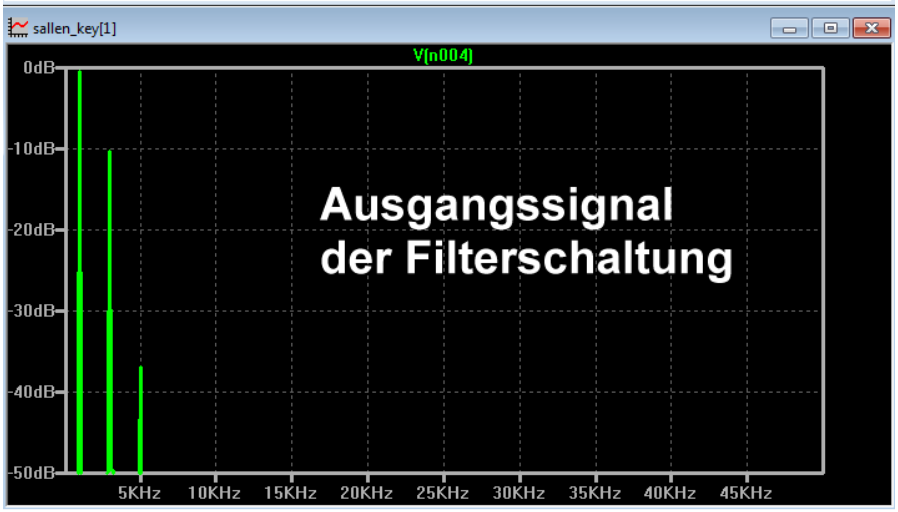
Zu dieser Simulation in der Time Domain….



….gehört dann dieses Ergebnis in der Frequency Domain.

(Einstellung:
0…50kHz mit einem Tick von 5kHz bei der Frequenzachse.

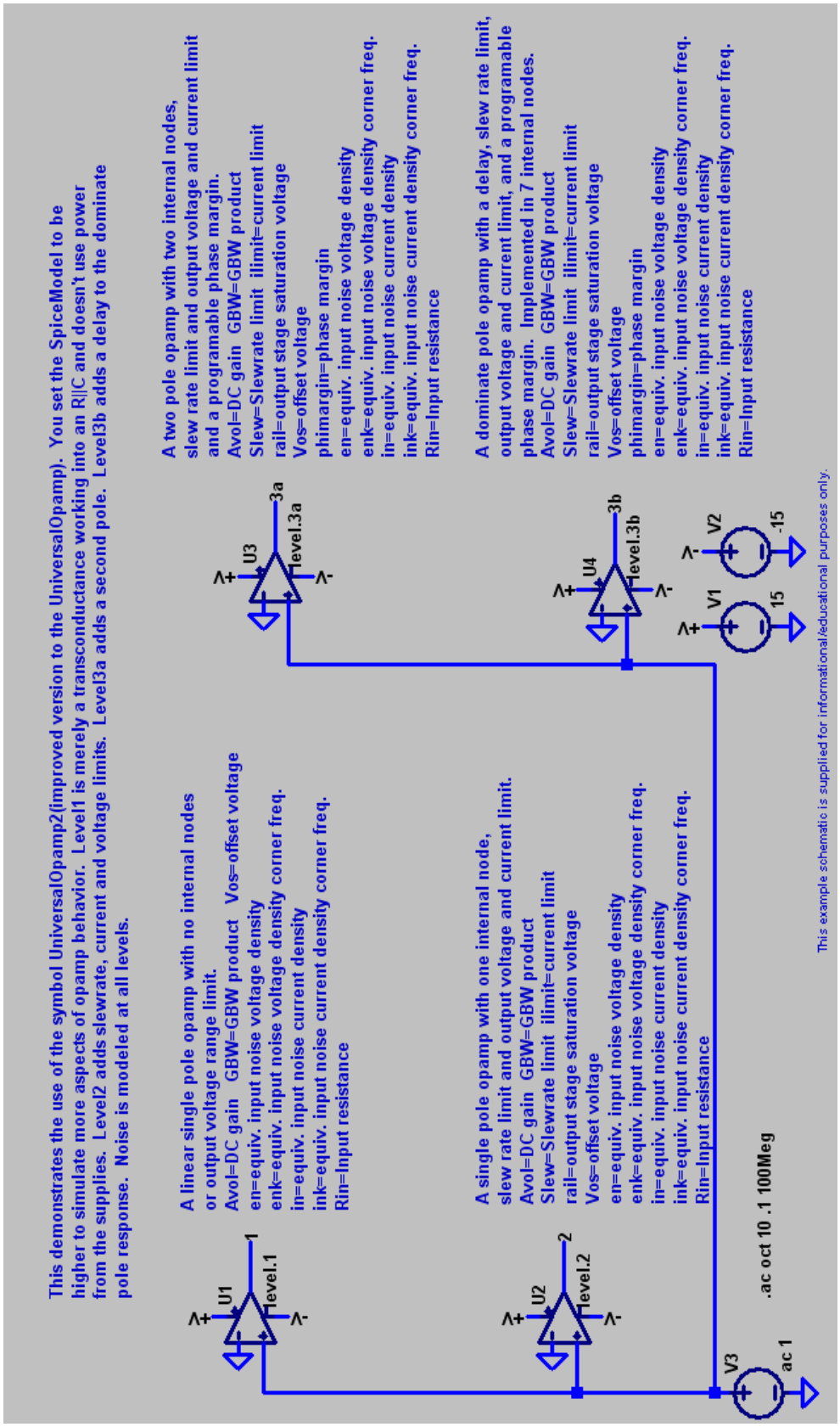
0….-50dB mit einem Tick von 10dB bei der senkrechten Amplitudenachse)

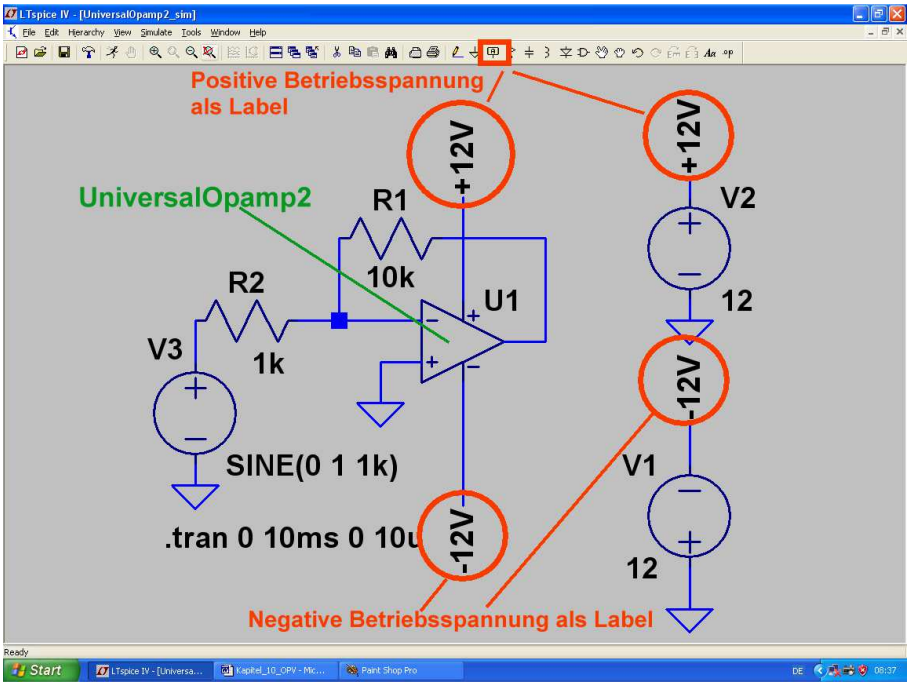


Die Filterwirkung wird nun sehr überzeugend dargestellt.

12.3. Das erweiterte OPV-Modell „UniversalOpamp2“

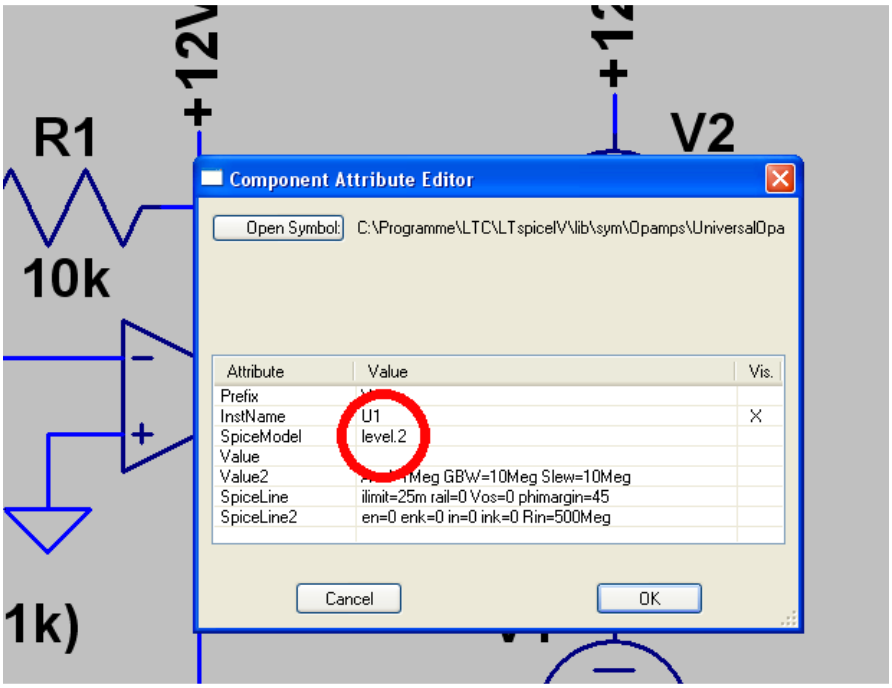
Das sind die Eigenschaften dieses neuen Modells, wobei die Ziffer 2 am Ende von „UniversalOpamp2“ den „level“ des Modells -- letztlich also den in ihm steckenden Entwicklungsaufwand! -- angibt.





Und so sieht die Schaltung aus, wenn man einen „**UniversalOpamp2**“ als invertierenden Verstärker mit 10facher Verstärkung einsetzt. Dabei wurden die beiden Speisespannungs-quellen mit „**Labels**“ versehen, um das Schaltbild übersichtlicher zu gestalten.

Hinweis:
Die Bezeichnung „**UniversalOpamp2“** sagt aus, dass das Spice-Modell für „**Level 2**“ aktiv ist. Die Level-Änderung ist sehr einfach:



Man klickt mit der rechten Maustaste auf das OPV-Schaltzeichen.

Dadurch öffnet sich das Property-Menü und man kann an der markierten Stelle den gewünschten Level (level.1 / level.2 / level.3a / level.3b) eintragen

Anmerkung:
Die Auswirkungen der verschiedenen Levels sieht man natürlich nicht in unserem Beispiel bei der Frequenz f = 1kHz.
Sobald man jedoch einen AC-Sweep zur Ermittlung des Frequenzgangs programmiert und sich dabei den Phasenverlauf mit einblenden lässt oder die Schaltung mit einem Pulssignal mit höherer Frequenz (z. B. f = 100kHz) ansteuert, zeigen sich die Unterschiede.

12.4. Probleme beim Betrieb mit nur einer Betriebsspannung

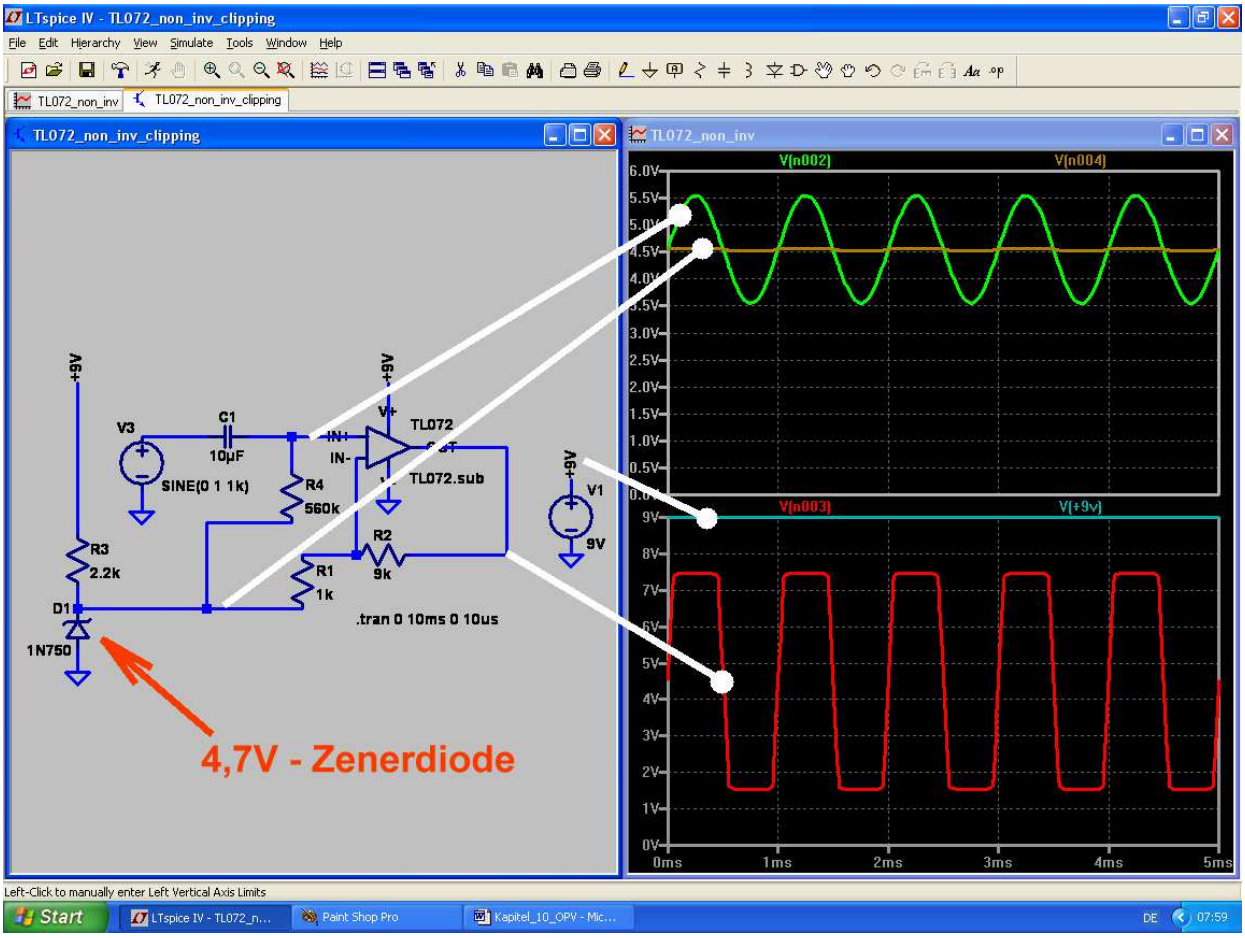
Oft steht in der Praxis nur eine einzige positive Versorgungsspannung zur Verfügung. Dann muss man Folgendes beachten:

a) Der **„V+“** - Anschluss des OPV wird an diese **Versorgungsspannung** angeschlossen.

b) Der „**V-**“ - Anschluss wird mit **Masse** verbunden.

c) Das Gleichspannungspotential an den Eingängen des OPVs muss dann auf halbe Betriebsspannung gebracht werden.

d) Bei vielen OPVs (hier als Beispiel: TL072) wird im Ausgang eine „Komplementärendstufe in Kollektorschaltung“ eingesetzt. Die funktioniert gut, hat aber den Nachteil, dass der Spitzenwert der Ausgangsspannung dadurch stets kleiner ist als die halbe Betriebsspannung. Ist eben so! Deshalb wollen wir die maximale unverwehrte Ausgangsspannung durch eine Simulation ermitteln.



Analyse der Schaltung und Auswertung der Ergebnisse:
Mit der Zenerdiode 1N750 wird eine „Vorspannung“ mit Ub / 2 = +4,5V aus der Versorgungsspannung von +9V erzeugt. Sie wird über den Widerstand R4 = 560k dem Nichtinvertierenden Eingang des OPVs zugeführt. Außerdem dient sie (wegen des kleinen dynamischen Widerstandes der Z-Diode) gleichzeitig als „Virtueller Masse-Fußpunkt“ für den Gegenkopplungs-Spannungsteiler aus R1 und R2.

Im oberen Diagramm ist die Spannung am Nichtinvertierenden Eingang (= Summe aus Vorspannung und zu verstärkender Wechselspannung) zu sehen.
Im unteren Diagramm wird die Ausgangsspannung am Pin „OUT“ des OPVs dargestellt. Deutlich ist zu sehen, dass im oberen und unteren Teil des Spannungsverlaufes nun jeweils 1,5V fehlen -- das ist der Teil, den die Endstufe zum Leben braucht. Zum genauen Vergleich ist auch die Versorgungsspannung mit +9V eingeblendet.

12.5. Verwendung eines OPV-Modells aus dem Internet

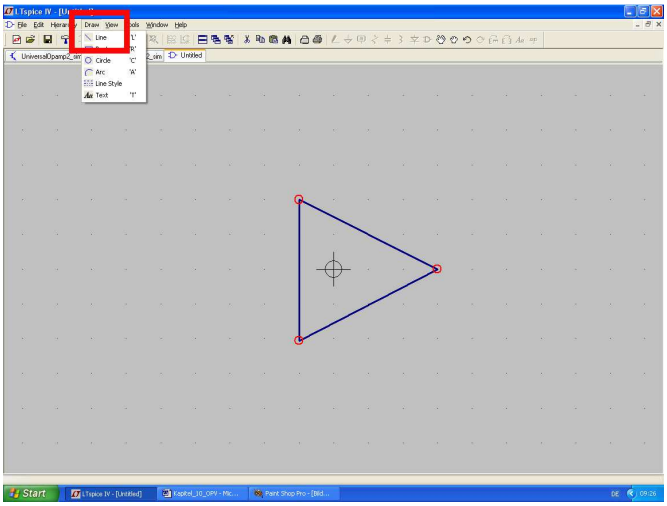
Wir wollen uns die Prozedur am Beispiel des bekannten Typs „TL072“ ansehen. Bei ihm handelt es sich um einen sogenannten „Wald- und Wiesentyp“: oft verwendet, leicht beschaffbar und nicht teuer.

1. Schritt:
Man gibt in die Internet-Suchmaschine den Begriff „**TL072 spice model**“ ein und lädt sich dann die entsprechende Datei z. B. vom Original-Entwickler (Texas Instruments) auf den Rechner.

2. Schritt:
Das Dokument wird geöffnet, der Text der Modellbeschreibung markiert und dieser Text in die Zwischenablage kopiert. Nun ruft man einen einfachen Texteditor (z. B. notepad) **mit Administrator-Rechten** auf und kopiert die Zwischenablage in ein leeres Blatt des Editors. Das entstehende File wird anschließend (mit dem Dateityp „**Alle Dateien**) unter der Bezeichnung „**TL072.sub**“ im Ordner **„LTSpice / lib / sub“** abgelegt. So sieht das am Ende aus:

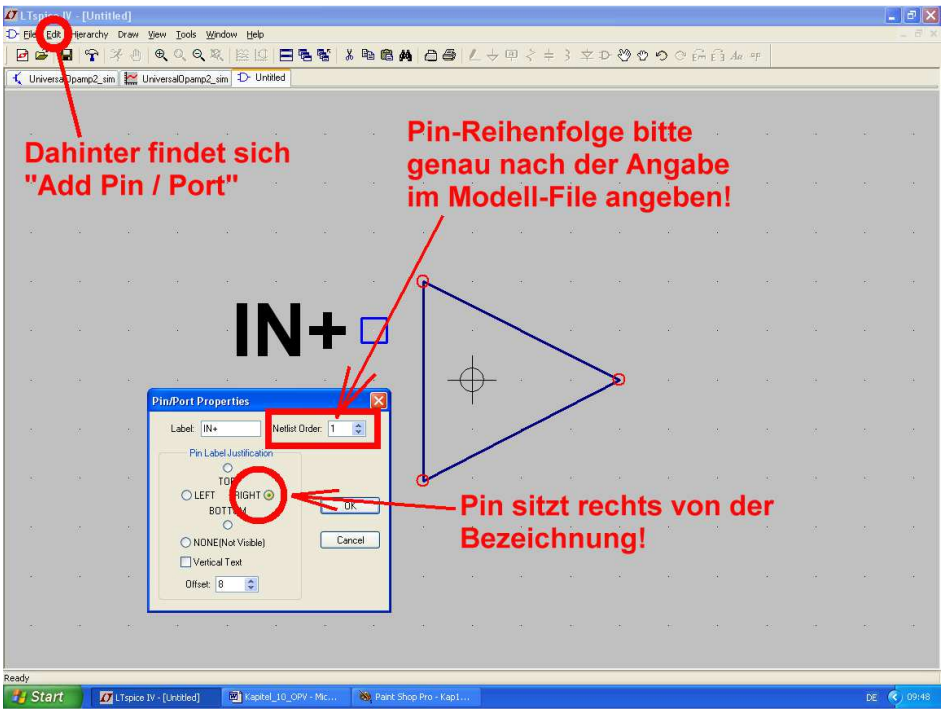
```
* TL072 OPERATIONAL AMPLIFIER "MACROMODEL" SUBCIRCUIT
* CREATED USING PARTS RELEASE 4.01 ON 06/16/89 AT 13:08
* (REV N/A)      SUPPLY VOLTAGE: +/-15V
* CONNECTIONS:   NON-INVERTING INPUT
*               | INVERTING INPUT
*               | | POSITIVE POWER SUPPLY
*               | | | NEGATIVE POWER SUPPLY
*               | | | | OUTPUT
*               | | | | |
.SUBCKT TL072    1 2 3 4 5
*
  C1   11 12 3.498E-12
  C2    6  7 15.00E-12
  DC    5 53 DX
  DE   54  5 DX
  DLP  90 91 DX
  DLN  92 90 DX
  DP    4  3 DX
  EGND 99  0 POLY(2) (3,0) (4,0) 0 .5 .5
  FB    7 99 POLY(5) VB VC VE VLP VLN 0 4.715E6 -5E6 5E6 5E6 -5E6
  GA    6  0 11 12 282.8E-6
  GCM   0  6 10 99 8.942E-9
  ISS   3 10 DC 195.0E-6
  HLIM 90  0 VLIM 1K
  J1   11  2 10 JX
  J2   12  1 10 JX
  R2    6  9 100.0E3
  RD1   4 11 3.536E3
  RD2   4 12 3.536E3
  RO1   8  5 150
  RO2   7 99 150
  RP    3  4 2.143E3
  RSS  10 99 1.026E6
  VB    9  0 DC 0
  VC    3 53 DC 2.200
  VE   54  4 DC 2.200
  VLIM  7  8 DC 0
  VLP  91  0 DC 25
  VLN   0 92 DC 25
.MODEL DX D(IS=800.0E-18)
.MODEL JX PJF(IS=15.00E-12 BETA=270.1E-6 VTO=-1)
.ENDS
```

3. Schritt:
LTSpice wird gestartet, aber unter „**File**“ die Option „**New Symbol**“ angeklickt. Dadurch öffnet sich ein neues Blatt in einem etwas helleren Grau, um es von einem Schaltbild unterscheiden zu können. Über dem Bildschirm liegt ein feines Punktraster und in der Mitte ist das Zentrum für das Symbol durch ein Fadenkreuz markiert.



Unter „**Draw**“ findet sich mit „**Line**“ die Möglichkeit zum Zeichnen von Linien. Damit erstellen wir das dreieckförmige OPV-Schaltzeichen.

Die drei Eckpunkte haben dabei einen gegenseitigen Abstand von vier „Kästchen“.



Jetzt werden die Anschlusspins platziert. Der zugehörige Aufruf findet sich hinter „**Edit**“.

Im nebenstehenden Menü muss man dann auf 2 Dinge achten:

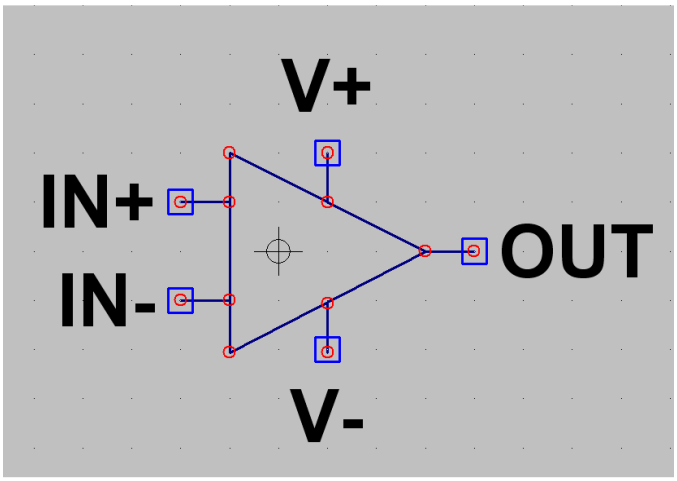
Erstens muss die richtige Pin-Nummer gemäß den Angaben im Modell-File („TL072.sub“) eingetragen werden.

Da gilt für den TL072 folgende Zuordnung:

Pin 1 = Nicht invertierender Eingang =	**IN+**
Pin2 = Invertierender Eingang =	**IN-**
Pin 3 = Positive Betriebsspannung =	**V+**
Pin 4 = Negative Betriebsspannung =	**V-**
Pin 5 = Ausgang =	**OUT**

Mit der Taste F7 und darauf folgendem Anklicken des Pinsymbols kann der Pin samt Bezeichnung verschoben werden. Wie immer, wird die Schiebe-Aktion durch „Escape“ beendet.

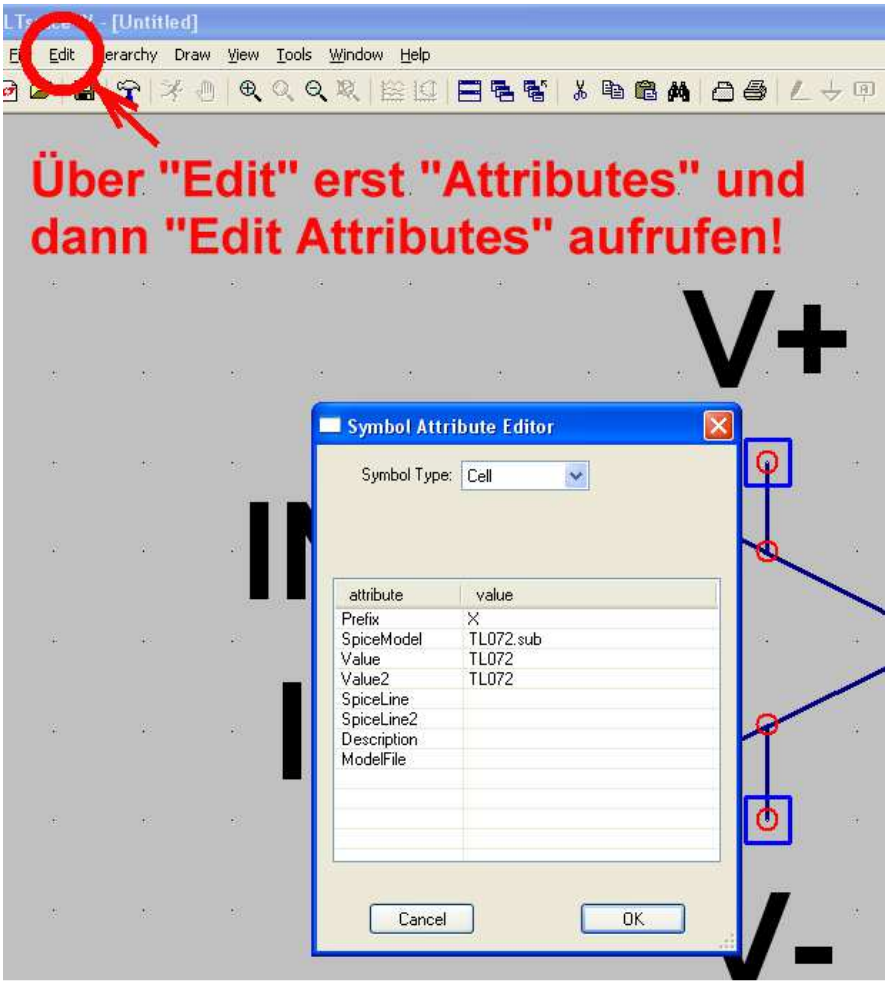
Zweitens muss man angeben, wo sich der Pin räumlich gegenüber der Bezeichnung befinden soll (hier: rechts).



Sind alle Pins platziert, dann sorgt man mit „**Draw Lines**“ für **Verbindungsleitungen** von jedem Pin zum Schaltzeichen.

Das ist das Ziel…

Dann werden dem Symbol die einzelnen Eigenschaften zugewiesen. Bitte deshalb die auftauchende Tabelle sehr gewissenhaft ausfüllen:



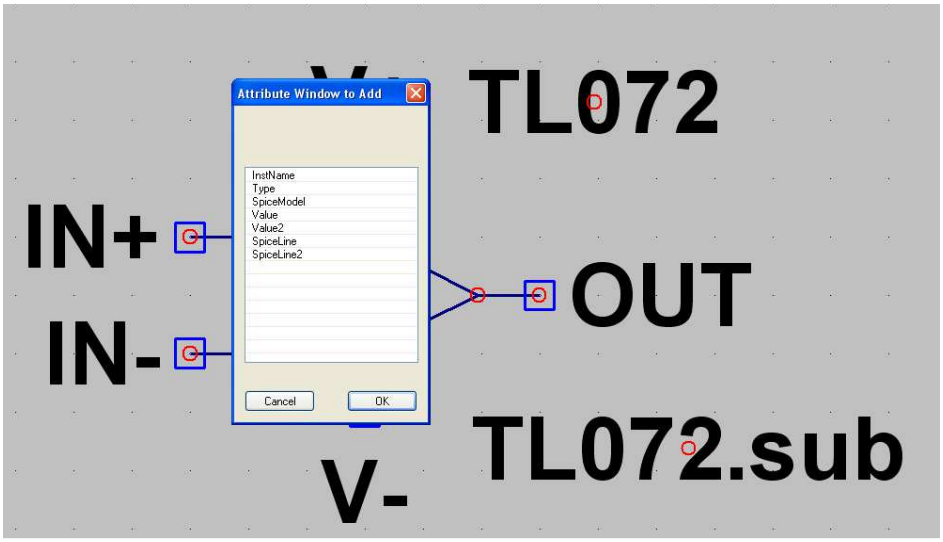
Symbol Type: Cell

Prefix: X

SpiceModel: TL072.sub

Value: TL072

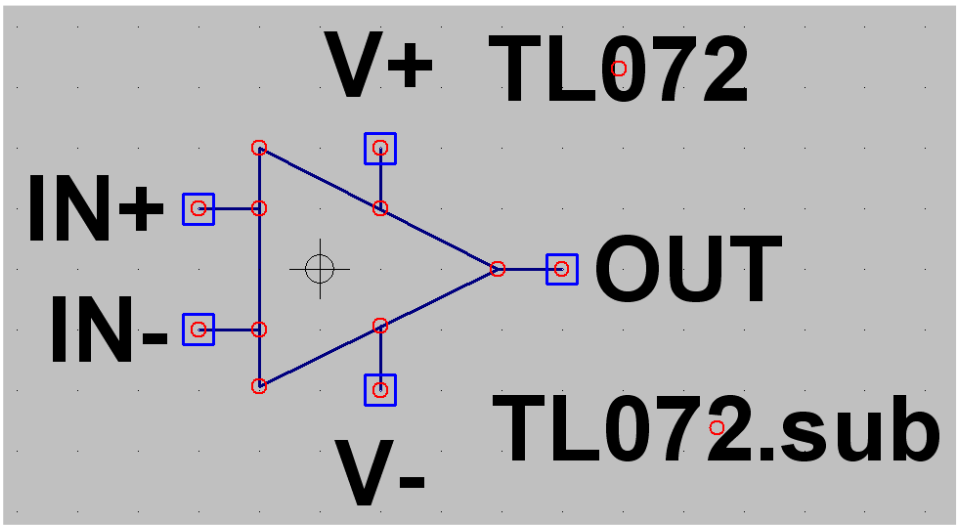
Value2: TL072



Nun wiederholt sich das Spiel: über „**Edit**“ und „**Attributes**“ kommen wir nun an „**Attribute Window“** heran.

In der auftauchenden Liste wird „**SpiceModel“** angeklickt und mit OK bestätigt. Die Modellangabe „**TL072.sub**“ kann nun neben dem Symbol abgesetzt werden.

Dann wiederholt man die Prozedur, wählt aber jetzt „**Value**“.



Wenn wir diesen Anblick vor uns haben, dann können wir das Symbol speichern.

Achtung:
Es muss im Pfad

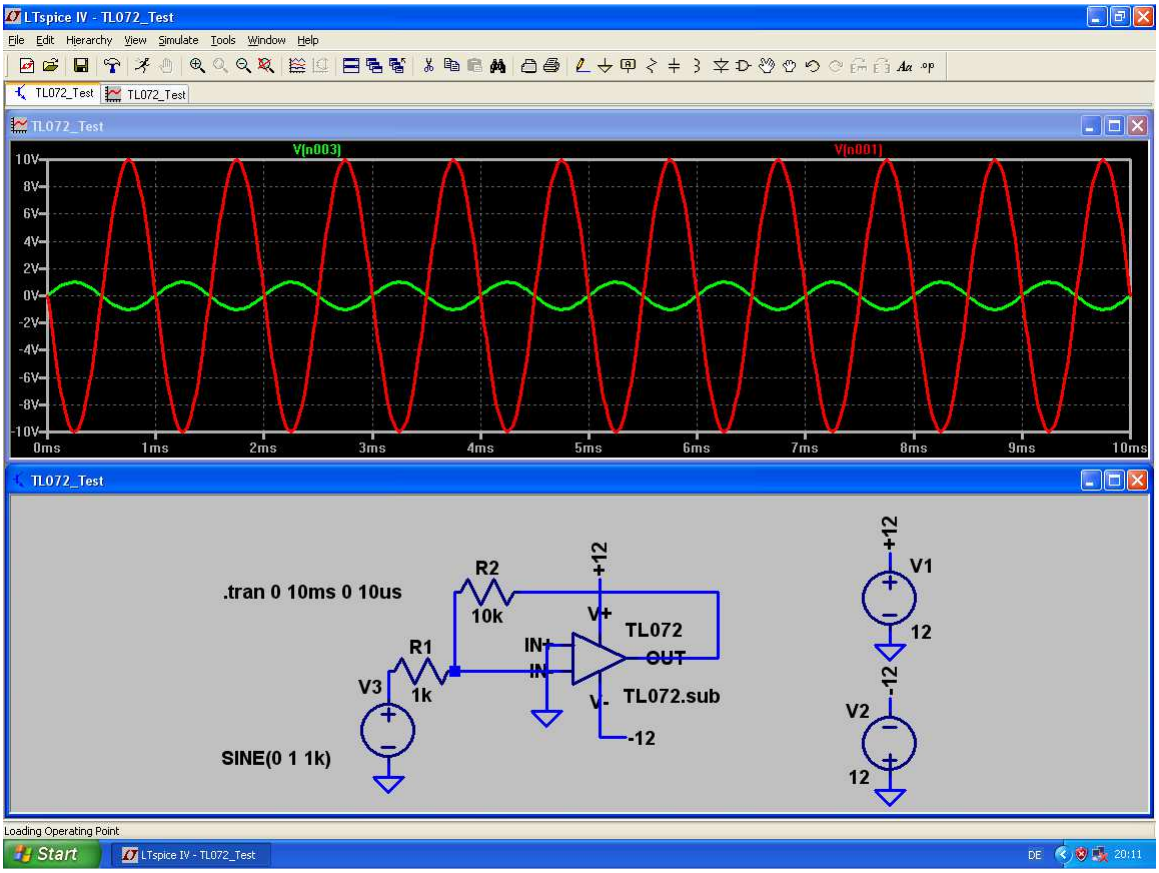
LTspice / lib / sym / Opamps“

unter dem Namen „**TL072**“ abgelegt werden

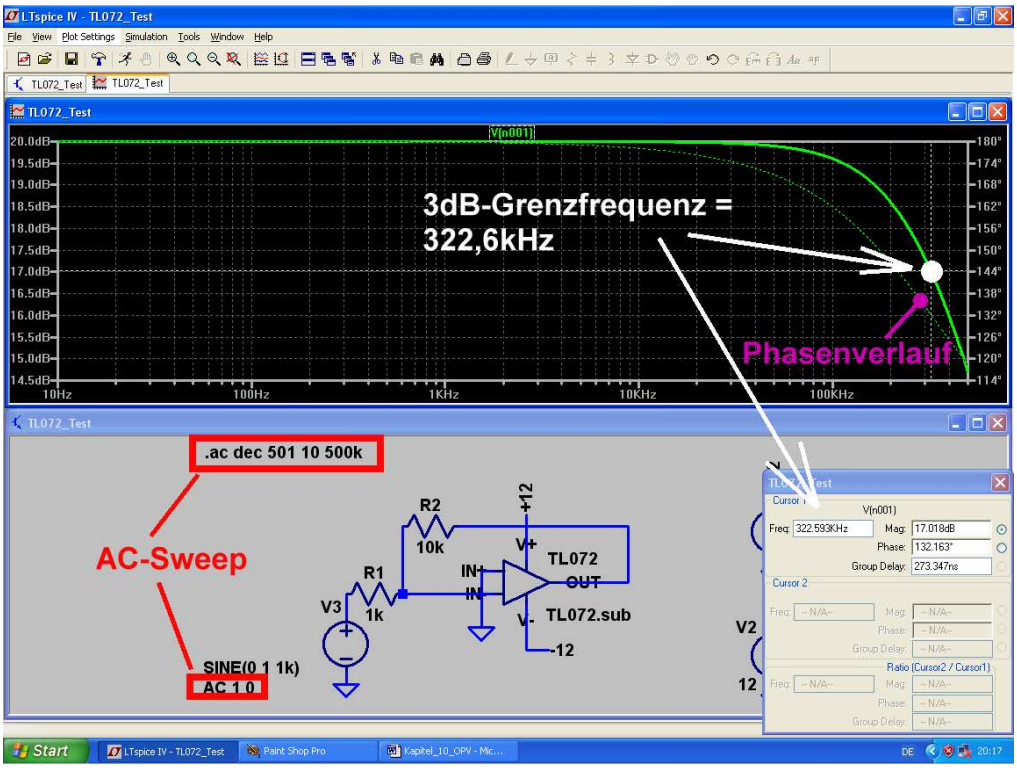
(…und das geht wieder mal nur mit Administrator-Rechten….)

Fertig!

12.5.1. Ein invertierender Verstärker mit dem TL072

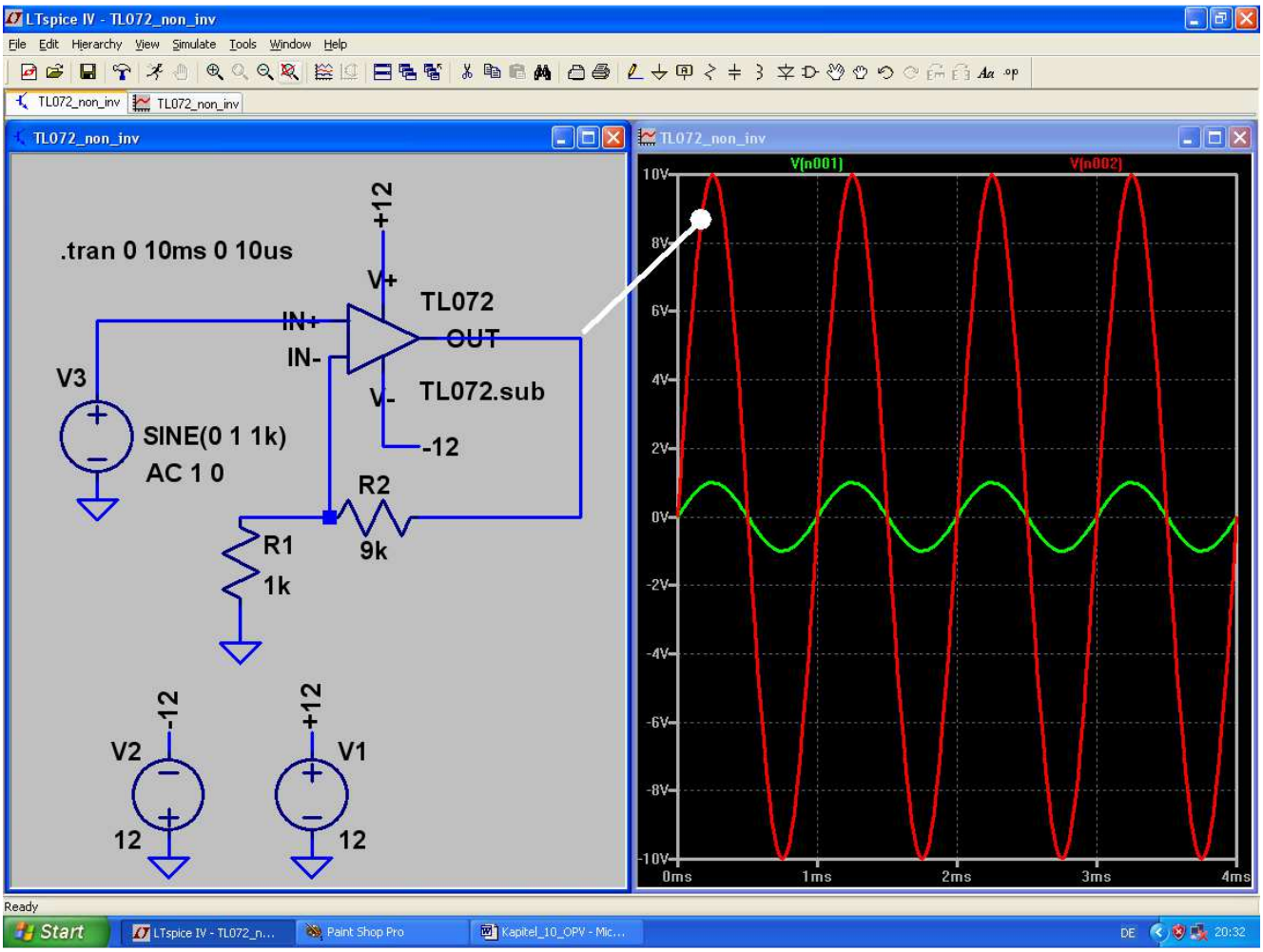


Interessant dürfte dagegen ein AC-Sweep sein, der das Verhalten bei höheren Frequenzen bzw. die obere Grenzfrequenz zeigt. Bitte mal simulieren!



12.5.2. Ein nicht umkehrender Verstärker mit dem TL072

Gefordert wird wieder 10fache Verstärkung, aber der Einsatz einer nicht umkehrenden Schaltung. Dadurch erreicht man einen sehr hohen Eingangs-widerstand:



13. Siebtes Projekt: DC-DC-Konverter

Dieses Projekt beginnt gleich mit einem Hinweis und einer **Warnung:**
Solche Schaltungen nutzen die zum Teil recht verwickelten Vorgänge in Schaltungen mit Induktivitäten aus. Das übliche einfache Modell einer Induktivität wird deshalb in der Simulationspraxis dieser Sache nicht gerecht und das Ergebnis ist entweder Nonsens oder die Simulation bricht ab. Nicht umsonst haben teure Programme hier spezielle zusätzliche Rechenmethoden eingebaut, die trotzdem ein Ergebnis ermöglichen. Aber: nur bei sehr teuren Maschinen vorhanden und dann streng geheim gehalten, wie das funktioniert.

Wir arbeiten jedoch mit dem „Berkely-Standard-SPICE“ und da hilft nur eines:

a) **Niemals mit idealen Induktivitäten arbeiten, sondern immer einen fühlbaren Reihenwiderstand vorsehen.**

b) Die Induktivitäten ergeben mit den in der Gesamtschaltung vorhandenen Kapazitäten oft **schwingfähige Gebilde** mit recht wilden Simulationsergebnissen. Da sollte man z.B. die Schaltung so stark am Ausgang belasten, dass diese Effekte stark bedämpft werden und / oder genügend lang simulieren.

13.1. Bereitstellung des Power-MOSFETs

Für die folgende Untersuchung der verschiedenen Grundschaltungen benötigen wir einen passenden elektronischen Schalter. Hier haben sich **Power-MOSFETs** durchgesetzt und die Firmen IRF (= International Recitifiers) oder Infineon (= ex Siemens) sind bekannte und führende Hersteller. Selbstverständlich stellen sie für alle Produkte auch SPICE-Modelle bereit, aber die Produktpalette ist ungeheuer groß.

Wir greifen bei unserer Simulation auf einen Typ aus der mitgelieferten Modellbibliothek zurück. Es handelt sich um den

BSC077N12NS3

der Firma Infineon und der kann Spannungen bis 120V verarbeiten (Imax = 98A / Ron = 8mΩ).

Hinweis:
Soll ein Typ eingesetzt werden, der nicht in der Vorratsliste aufgeführt ist, dann beginnt wieder das alte Spiel:

a) Modell aus dem Internet holen

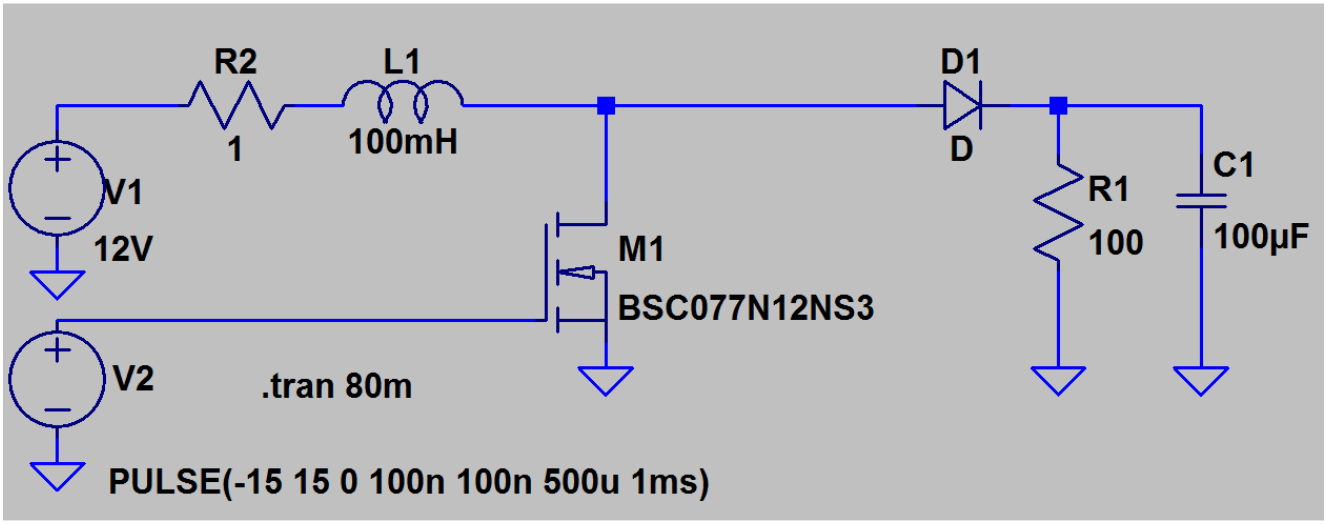
b) Modell in ein leeres Blatt des Texteditors einfügen

c) File als „*.sub“-Datei im Verzeichnis „LTspice / lib / sub“ mit Administrator-Rechten speichern.

d) Symbol erstellen und abspeichern.

usw.

13.2. Der Step-Up-Konverter (= Aufwärtswandler)



Die Schaltung besteht zunächst aus der Gleichspannungsquelle V1 mit 12 V, an die unsere Induktivität mit L = 100mH (Fachausdruck: „Speicherdrossel“) angeschlossen wird. Dazu gehört der geforderte Reihenwiderstand mit 1Ω, bevor der Power-MOSFET seine Aufgabe als gesteuerter Kurzschluss-Schalter gegen Masse übernehmen kann. Solange er eingeschaltet ist, liegt die Drossel an +12 V und der Strom in ihr steigt etwa linear an. Dadurch wird aber nach der Beziehung

$$E = \frac{1}{2} \bullet L \bullet I^2$$

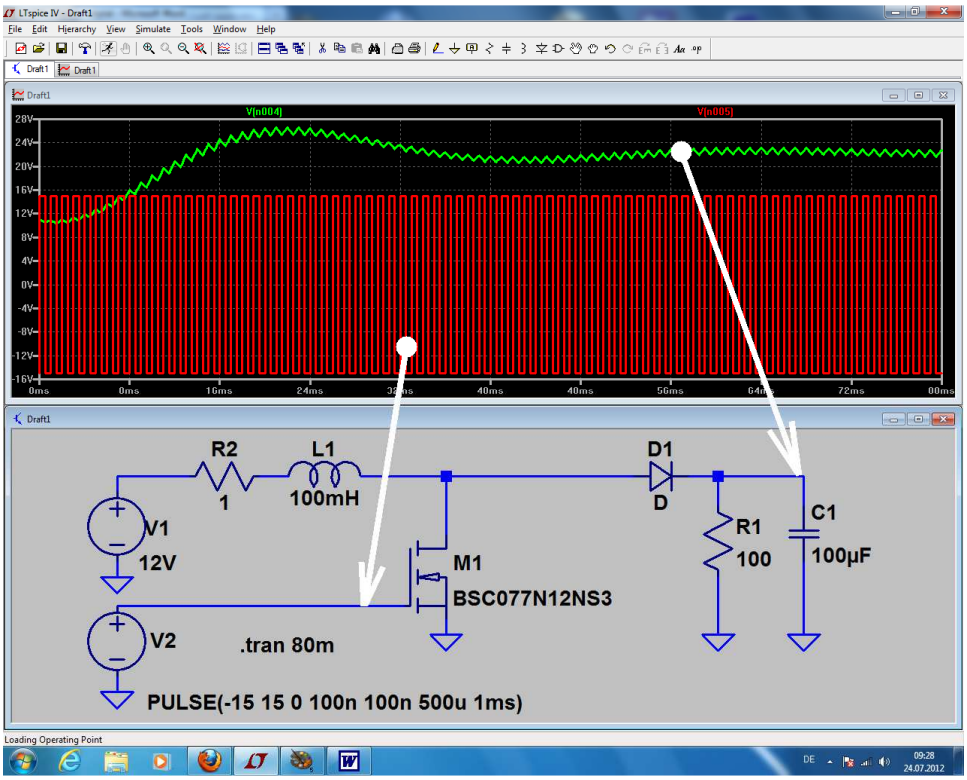
immer mehr Energie im Magnetfeld der Spule gespeichert. Sperrt man nun den FET, dann kann diese Energie nicht verschwinden, außerdem kann sich der Strom in einer Spule nicht ruckartig ändern (denn das gäbe unendlich hohe Induktionsspannungen). Also fließt der Spulenstrom nach rechts, schaltet die Schottky-Diode D ein, speist den Lastwiderstand und lädt den Kondensator nach. Dieser Vorgang wird (da am Gate des FET eine Rechteckspannung anliegt) dauernd wiederholt, bis ein Gleichgewicht zwischen aufgenommenen und abgegebenen „Energieportionen“ hergestellt ist.
Wichtig: bei dieser Schaltung addieren sich die Versorgungsgleichspannung und die Induktionsspannung der Spule.

Folge: die Ausgangs-Gleichspannung wird höher als die Betriebsspannung!

Nun wird diese Schaltung gezeichnet und gleich die Puls-Spannungsquelle programmiert.

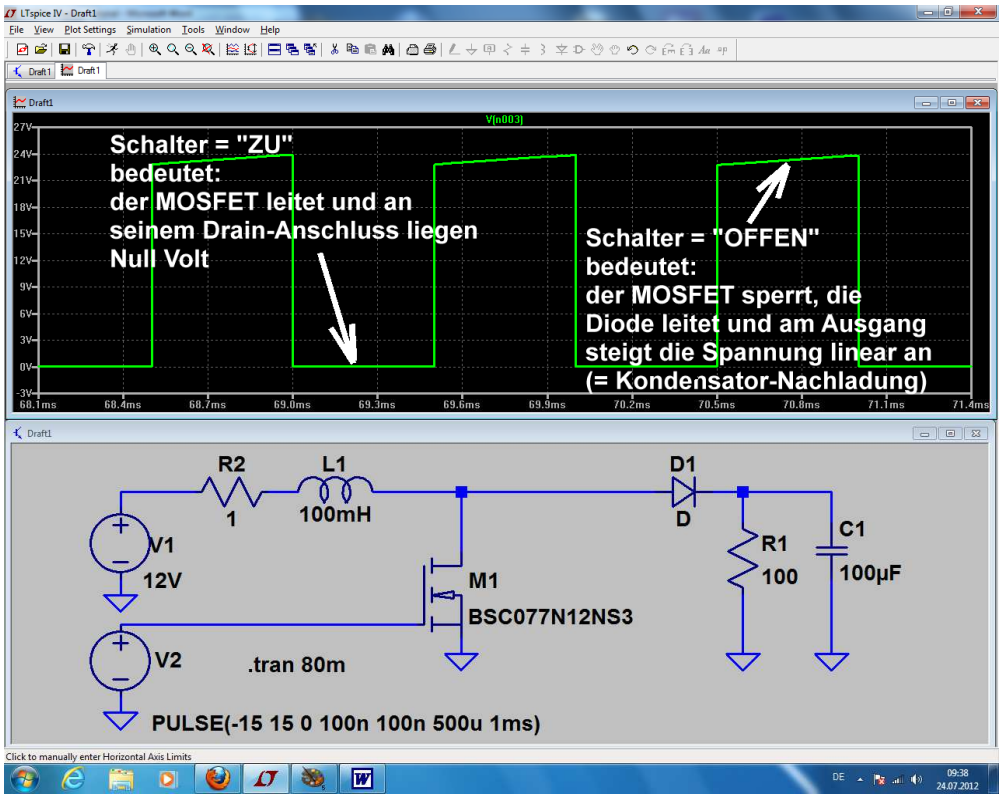
Wir steuern den Power-MOSFET am Gate mit einem symmetrischen Rechtecksignal an, das eine Frequenz von 1 kHz aufweist.
Eingeschaltet wird er mit +15 V, gesperrt dagegen mit –15 V.

Als Simulationszeitraum wählen wir 0…..80ms.



Gut zu erkennen ist der Mittelwert der Ausgangsspannung von etwa +23V und darauf ein kleiner Sägezahn-Anteil (= Auf- und Entladung des Ausgangs-Kondensators).

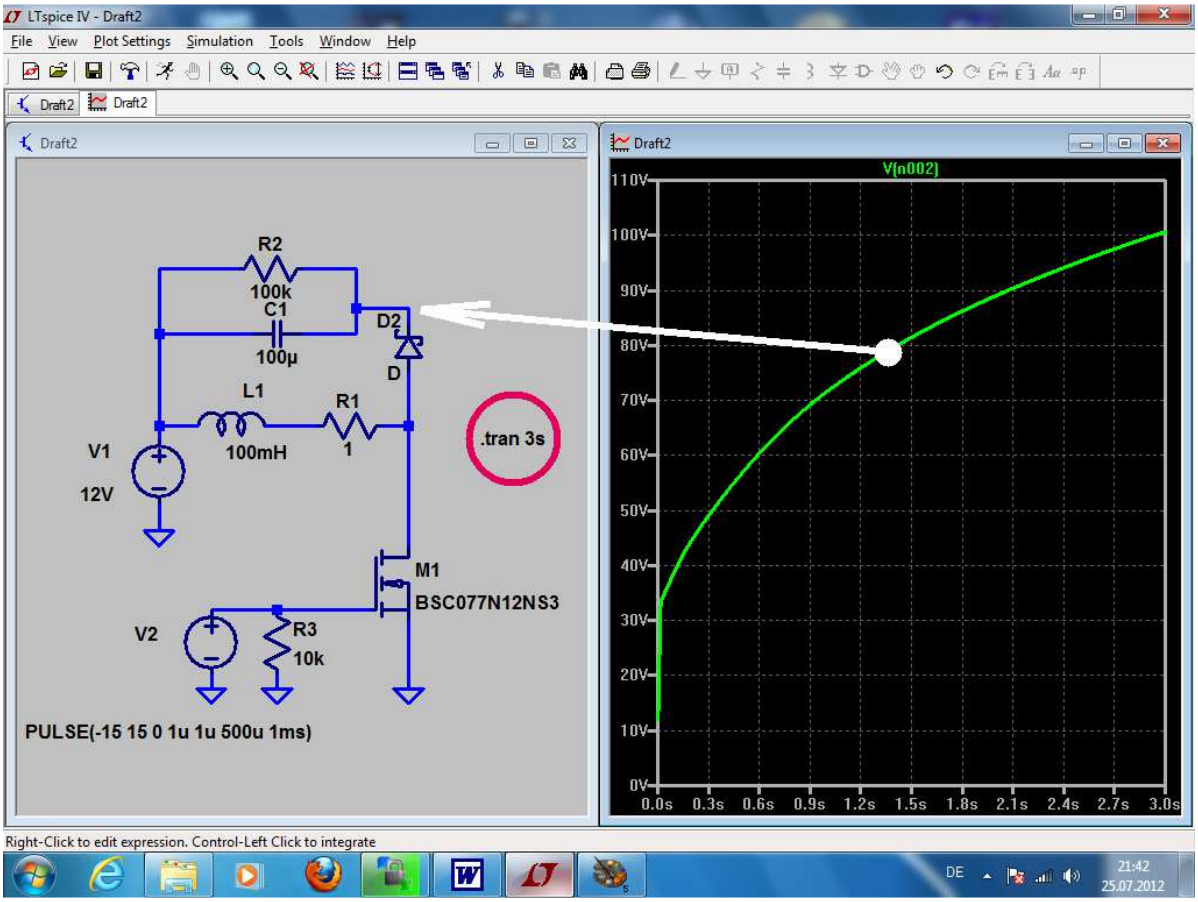
Werfen wir jetzt noch einen Blick auf die Drainspannung des FETs:



Wer Spaß daran hat, kann sich nun auch die Gate-Spannung auf den Schirm holen und / oder die Simulation für andere Werte der Ansteuer-frequenz, der Induktivität oder des Ladekondensa-tors wiederholen.

Hinweis:
Die Regelung oder Stabilisierung der Ausgangsspannung erfolgt bei dieser Schaltung immer über das Tastverhältnis (= Verhältnis von FET-Einschaltdauer zur Periodendauer)!

13.3. Der Flyback-Konverter (= Sperrwandler)



Genau hinsehen:
hier wird das rechte Ende der Drosselspule bei eingeschaltetem FET direkt an Masse gelegt, während das linke Ende an der (konstanten) Betriebsgleichspannung von +12 V liegt. Dadurch steigt der Spulenstrom linear an und in der Spule wird wieder mal im Magnetfeld Energie gespeichert.
Sperrt der FET plötzlich, so muss der Spulenstrom in alter Richtung weiterfließen. Aber da ihm der Weg über den FET versperrt ist, schließt sich der Stromkreis nun über die **Schottky-Diode „D“** und den Ladekondensator bzw. Lastwiderstand. Dieser Stromfluss hört dann auf, wenn die in der Spule gespeicherte „Energieportion“ voll an den Ausgangskreis übergeben wurde.

Das führt zu recht interessanten Eigenschaften der Schaltung:

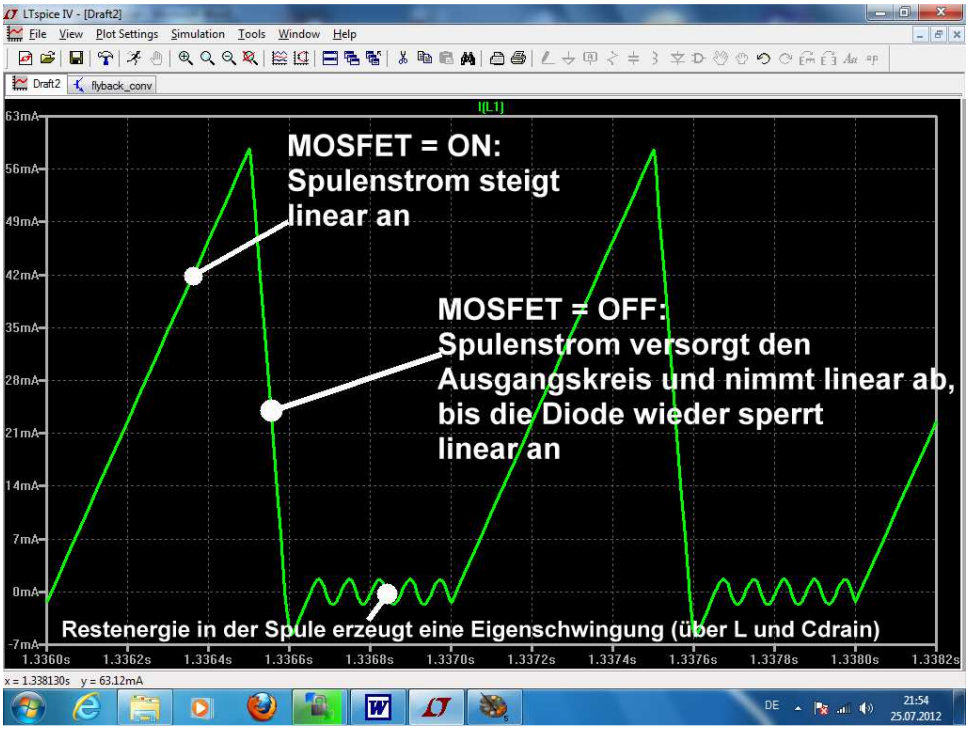
a) Die Schaltung darf am Ausgang ohne schlimme Folgen kurzgeschlossen werden, da die Energieübertragung zum Ausgang wie beim Wasserschöpfen abläuft: Eimer füllen (= Spule aufladen) / Eimer ausleeren (= Energie an Ausgang übergeben). Dadurch ist weder der FET noch irgendein Teil der Schaltung beim Kurzschluss gefährdet.

b) Allerdings gibt es noch weitere Effekte: entfernt man den Lastwiderstand, dann wird durch die übergebenen Energieportionen der Kondensator geladen und geladen und immer weiter geladen**... folglich läuft die Ausgangsspannung immer höher und höher, bis irgendetwas nachgibt!** Zur Demonstration dieses Effektes wurde der Lastwiderstand auf 100kΩ erhöht und selbst damit steigt die Ausgangsspannung nach 3 Sekunden immer noch weiter an!

c) Der **Pluspol der Kondensatorspannung** (= Ausgangsspannung) ist an der **Katode** der verwendeten Schottky-Diode, außerdem ist diese Ausgangsspannung im Normalfall immer viel höher als die Betriebsspannung!

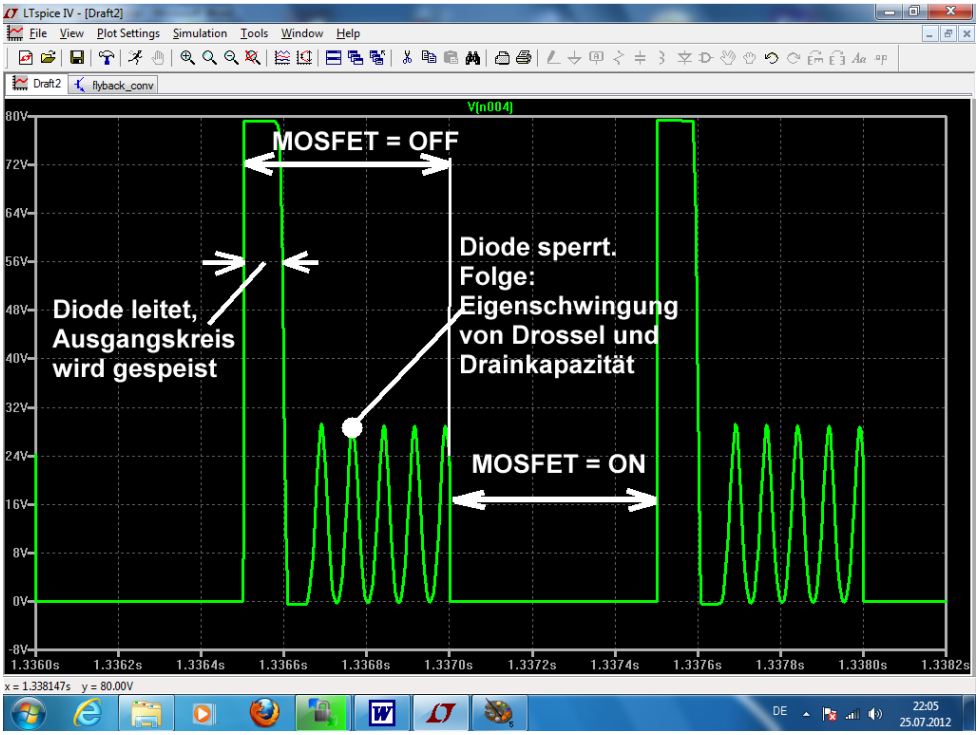
Hinweis:
Die Regelung oder Stabilisierung der Ausgangsspannung erfolgt bei dieser Schaltung immer über das Tastverhältnis (= Verhältnis von FET-Einschaltdauer zur Periodendauer)!

Die obige Schaltung wird nun gezeichnet und simuliert. Als Ansteuersignal dient wieder eine symmetrische Rechteckspannung mit +15 V Maximalwert, -15 V Minimalwert und einer Frequenz von 1 kHz.
Da die Schwellspannung der Diode nicht Null Volt beträgt, endet der Aufladevorgang des Ausgangskondensators bereits, bevor die „gespeicherte Magnetfeld-Energie in der Spule aufgebraucht ist."



Diese Restenergie tobt sich nun als gedämpfte Eigenschwingung (des Resonanzkreises, der aus L und der Drainkapazität des MOSFETs gebildet wird) aus!

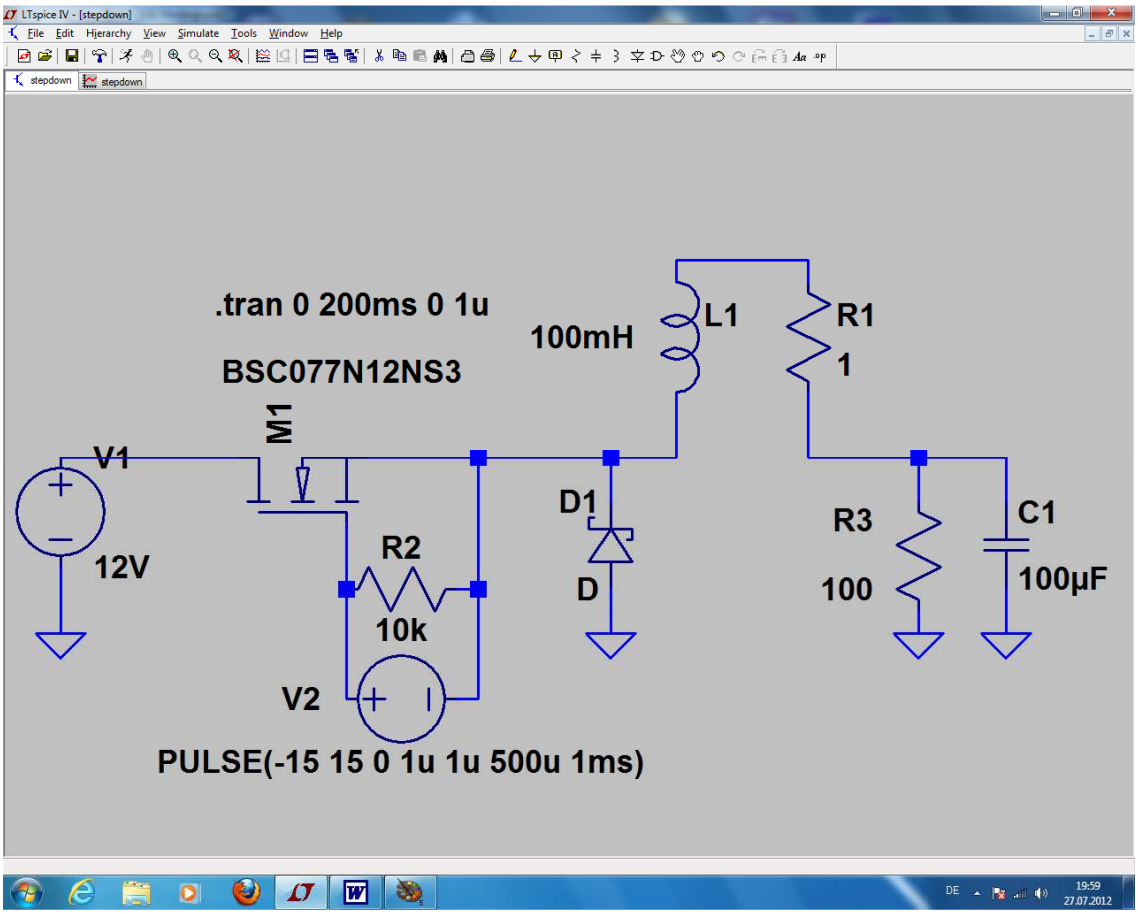
Das ist in der nebenstehenden Simulation gut zu sehen.



Genau so schön muss man das an der Drainspannung des MOSFETS sehen:

Die Eigenschwingung des Kreises aus Drossel-Induktivität und Drainkapazität des POWER-MOSFETs ist gut zu erkennen.

13.4. Der Step-Down-Konverter (= Abwärtswandler)



Bei dieser Schaltung ist der MOSFET als Schalter zwischen der Betriebsspannung und der Drossel eingefügt. Die Diode ist in Sperrrichtung zwischen dem Anfang der Drossel und Masse angeordnet (...sie wird hier oft als „Freilaufdiode“ bezeichnet).

Vorgänge in der Schaltung:

a) Wird der MOSFET eingeschaltet, dann fließt aus der Gleichspannungsquelle ein linear ansteigender Strom über den FET, die Drossel und den Ausgangskondensator bzw. Lastwiderstand nach Masse. Dadurch wird wieder im Magnetfeld der Spule Energie gespeichert.

b) Wird der FET gesperrt, so muss der Strom seinen Weg über die Freilaufdiode nehmen, damit sich der Stromkreis wieder korrekt schließt (…der Weg über die Spannungsquelle ist wegen des gesperrten MOSFETs unterbrochen). Die gespeicherte Magnetfeldenergie versorgt nun den Lastkreis. Bitte beachten: der Strom klingt wieder linear bis auf Null ab **und in dieser Zeit ist die Spannung an TPv2 (...wegen der leitenden Diode!) NEGATIV!!!**

Hinweise:

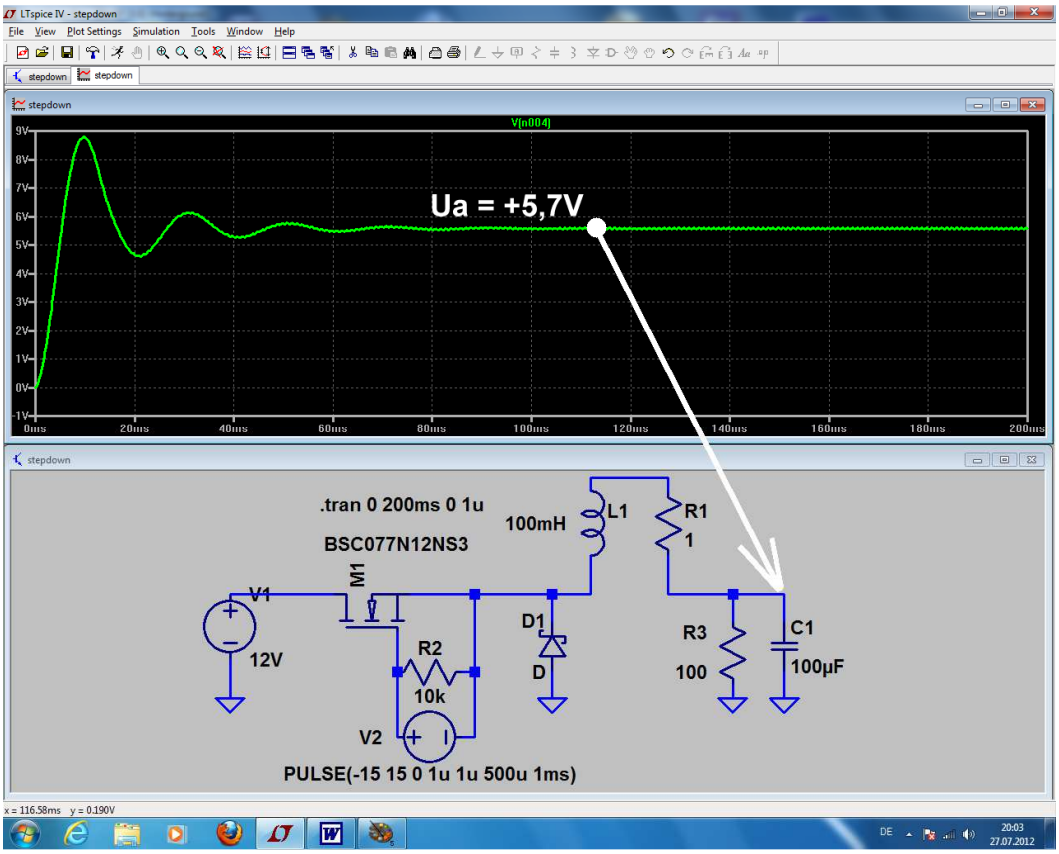
1) **Die Regelung oder Stabilisierung der Ausgangsspannung erfolgt bei dieser Schaltung immer über das Tastverhältnis (= Verhältnis von FET-Einschaltdauer zur Periodendauer)!**
2) **Die Ausgangsgleichspannung ist bei dieser Schaltung IMMER kleiner als die Eingangsspannung, ihren Wert erhält man, wenn man die Eingangsspannung mit dem Tastverhältnis multipliziert.**

Bei unserem Beispiel haben wir eine Einschaltzeit von 0,5ms und eine Periodendauer von 1ms vorgesehen. Deshalb sind am Ausgang

$$0{,}5 \times 12V = 6V$$

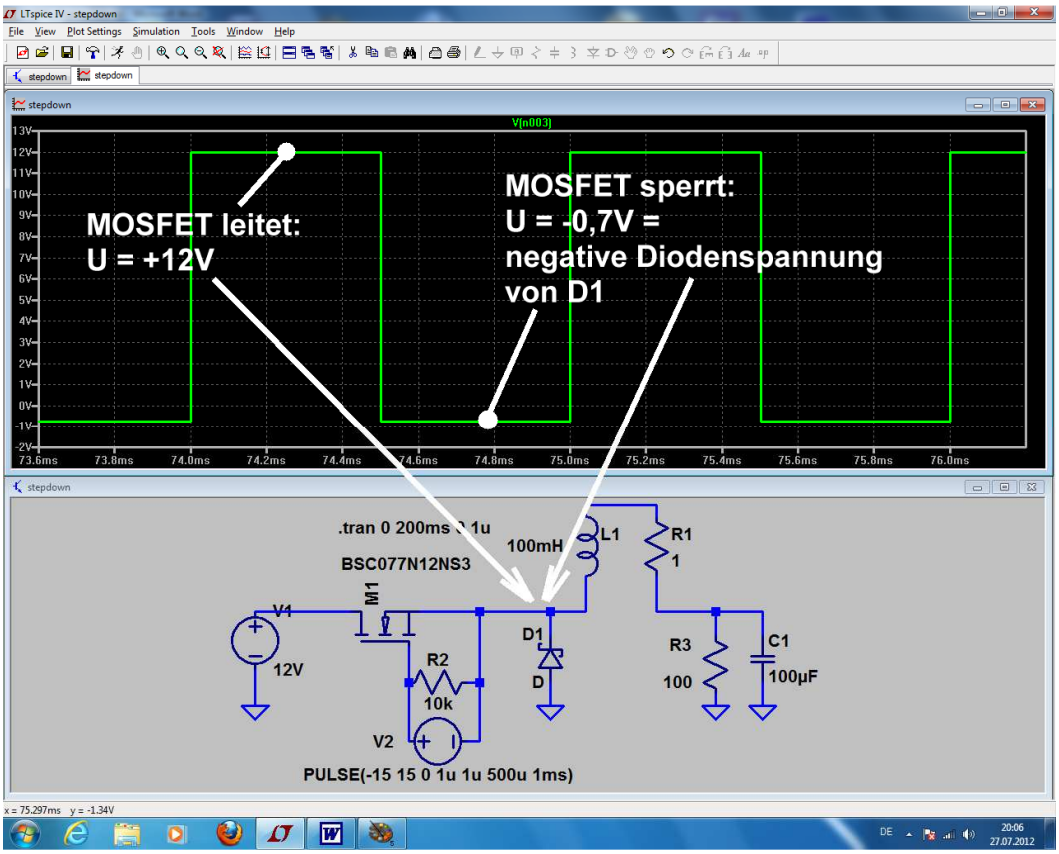
zu erwarten.

Sehen wir uns mal die Simulation der Ausgangsspannung an:



Na, also…..

Und so sieht die Spannung hinter dem MOSFET -- also an der Katode der Diode -- aus:



14. Achtes Projekt: Phasenanschnitt-Steuerung mit Thyristor

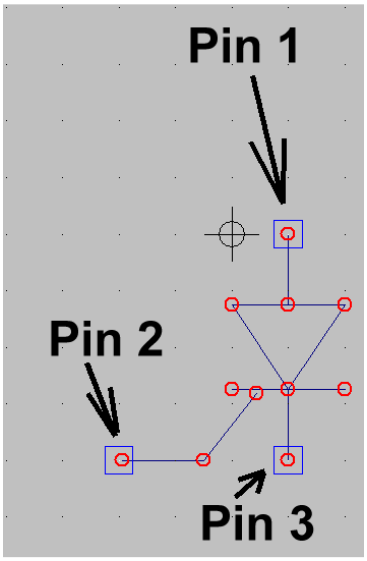
14.1. Das eingesetzte Thyristor-Modell

Recht gut bewährt hat sich für die Simulation der Thyristor-Typ **2N5171**
Es handelt sich hier um einen 20A / 600V – Thyristor, mit dem in der Praxis doch einiges anzufangen ist. Zur Beschaffung des SPICE-Modells suchen wir im Internet nach der Datei

thyristr.lib

Sie enthält den erforderlichen Subcircuit und wird unter dem Pfad **„lib / sub“ im LTSpiceIV-Verzeichnis** abgelegt.

Aber Vorsicht:
eingefügt. Das kann dann endgültig als „thyristr.lib“ am angegebenen Ort gespeichert werden. Wenn sie dagegen als HTML-File auf unserem Bildschirm auftaucht, wird sie erst geöffnet, dann der Inhalt markiert, in die Zwischenablage kopiert und in ein neues Blatt des Texteditors Und dann geht es mal wieder los:

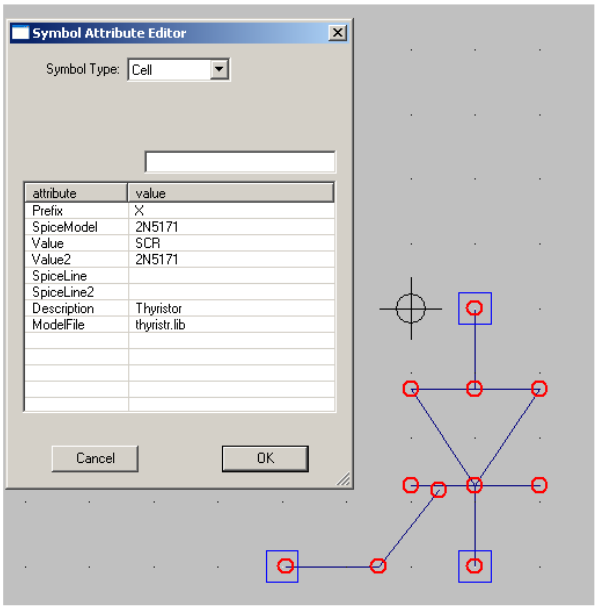


1. Schritt:
Unter „File“ wählen wir „**New Symbol**“ und öffnen anschließend das Symbol für die **Diode („diode.asy“ im Ordner „lib / sym).** Wir ergänzen das Schaltzeichen um den Gate-Anschluss und nehmen uns gleich eine wichtige Sache vor:

Entsprechend dem SPICE-Modell müssen wir streng die Zuordnung

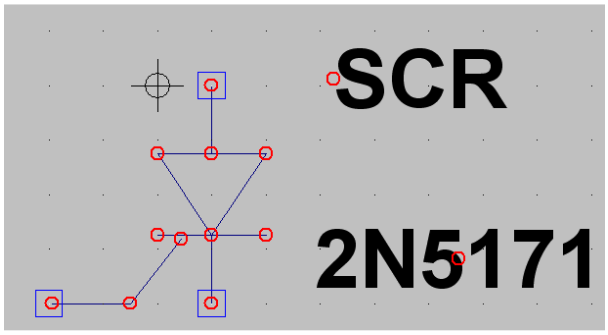
Anode = Pin 1
Gate = Pin 2
Katode = Pin 3

einhalten. Deshalb klicken wir mit der rechten Maustaste nacheinander auf jeden Anschlusspunkt und stellen sicher, dass diese Reihenfolge stimmt.



2.Schritt:
Jetzt öffnen wir (unter „Edit / Attributes“) die Option „Edit Attributes“ und erstellen sorgfältig die nebenstehende Attributstabellle.

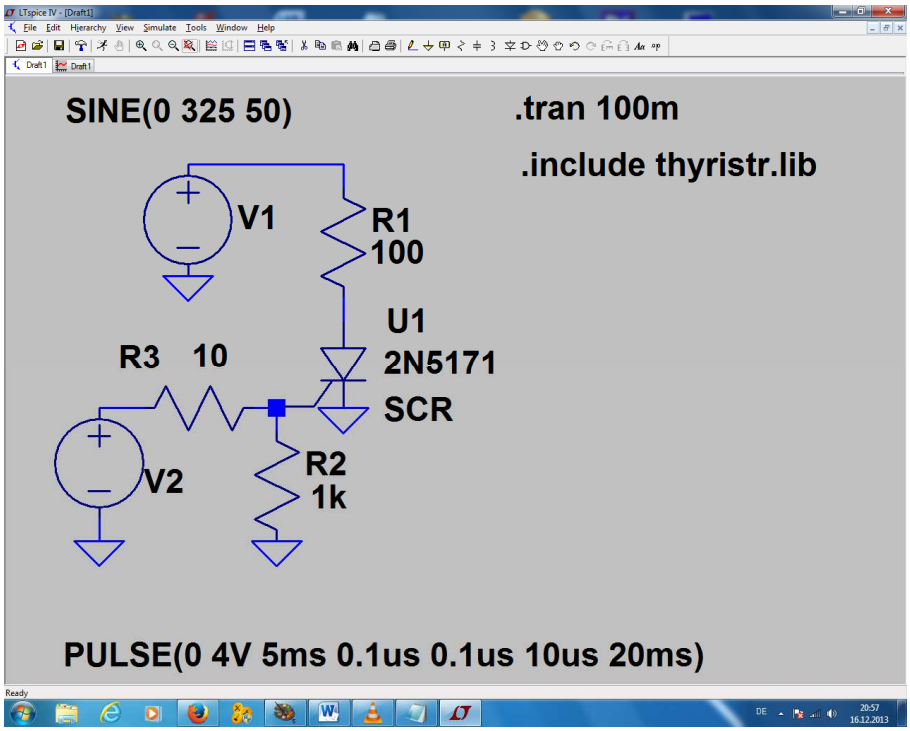
Prefix	**X**
SpiceModel	**2N5171**
Value	**SCR**
Value 2	**2N5171**
Description	**Thyristor**
ModelFile	**thyristr.lib**



3. Schritt:
Wir öffnen nochmals „Edit / Attributes“, wählen aber anschließend „**Attribute Window**“. In der auftauchenden Liste klicken wir auf „Value“ und platzieren anschließend den Hinweis „SCR“ neben dem Schaltzeichen. Das wird nochmals wiederholt, aber nun „SpiceModel“ gewählt. So sieht der Endzustand aus.

Jetzt muss nur noch das fertige Modell in einem **neuen Ordner „Thyristoren“ in „lib / sym“** unter der Bezeichnung „2N5171.asy“ abgelegt werden (..bitte wieder an die Administrator-Berechtigung denken….).

14.2. Schalten von Ohm’schen Lasten



Schaltungsbeschreibung:
Die Sinus-Spannungsquelle V1 (Spitzenwert = 325 V, Frequenz = 50Hz) speist eine Reihenschaltung aus dem Widerstand R1 (100 Ω) und dem Thyristor 2N5171. Vom Gate nach Masse ist meist ein Widerstand R2 (hier: 1kΩ) geschaltet. Über einen Strombegrenzungs-widerstand (R3 = 10Ω) steuert eine Pulsspannungsquelle das Gate des Thyristors an.

Erläuterungen:

a) eine Simulationszeit von 0….100ms wird durch die SPICE-Direktive **„.tran 100m“** festgelegt.

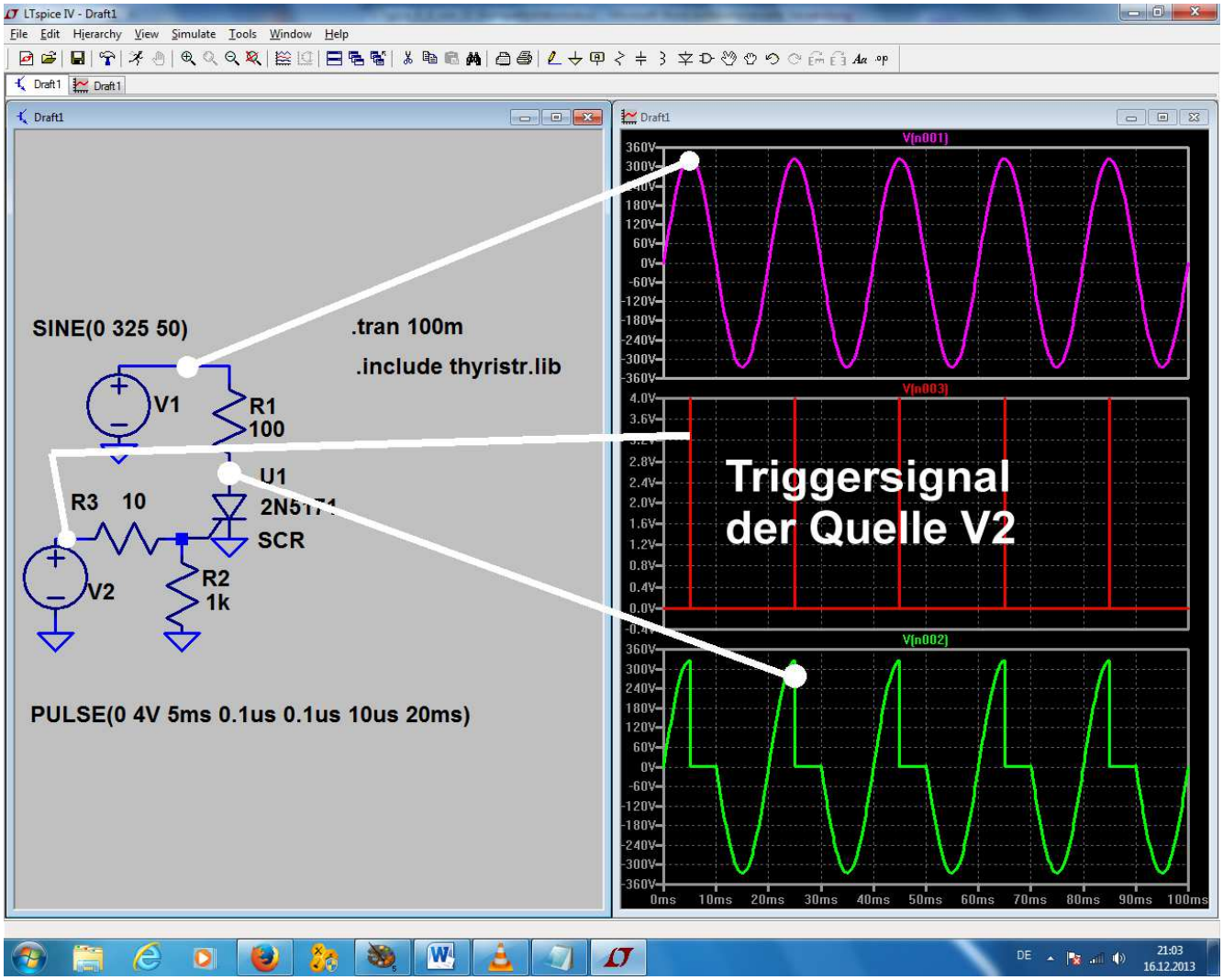
b) Mit „**.include thyristr.lib“** wird die Datei mit den Thyristor-SPICE-Modellen bereitgestellt.

c) Die Pulsspannung wird durch folgendes Attribut der Spannung V2 erzeugt:

PULSE (0V 4V 5ms 0.1us 0.1us 10us 20ms)

Das ergibt eine Minimalspannung von Null Volt, eine Maximalamplitude von +4V, eine Startverzögerung von 5ms, eine Anstiegs- und Abfallzeit von je 100ns, eine Pulslänge von 10 Mikrosekunden und eine Periodendauer von 20ms

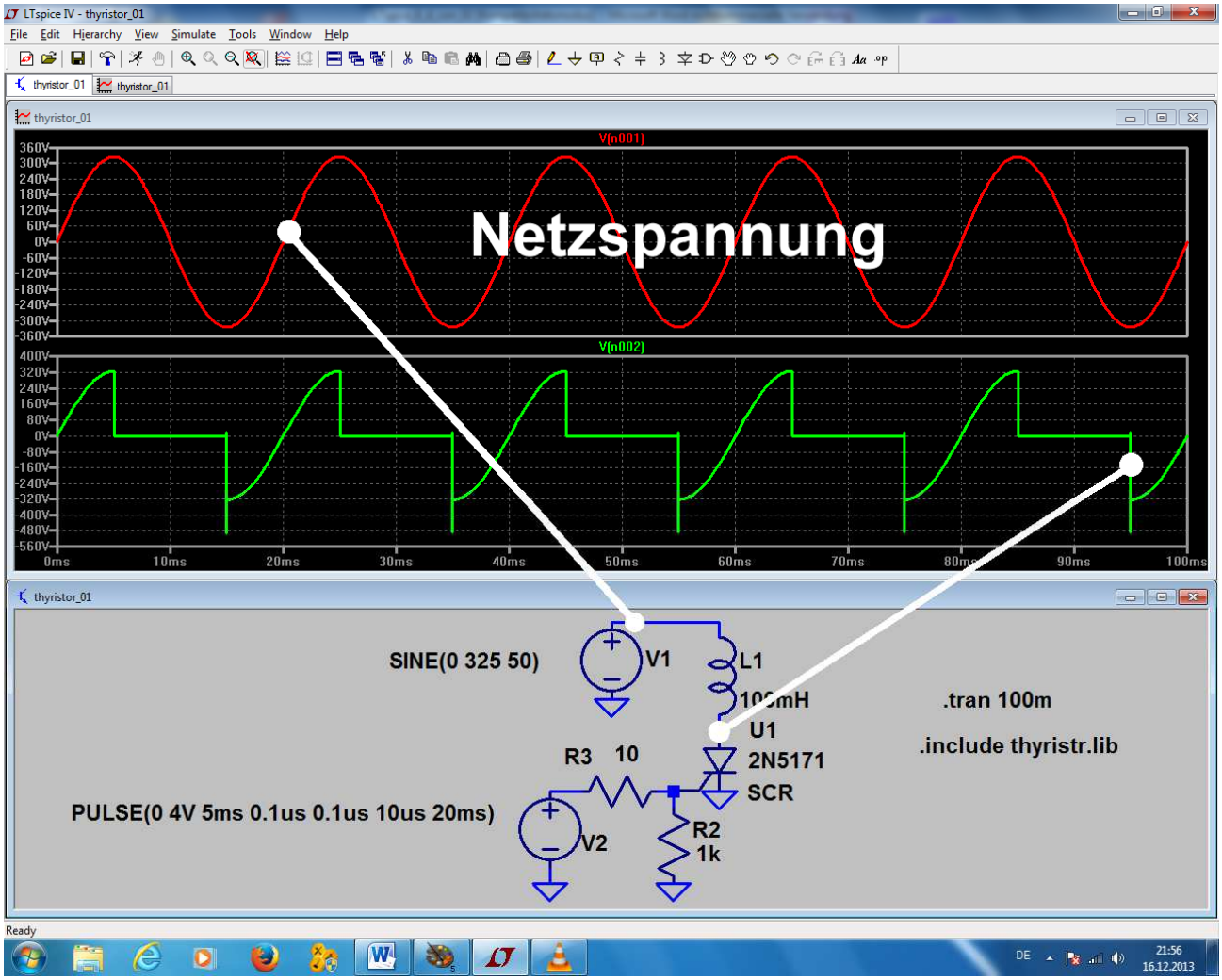
Und so sehen die Spannungen in der Schaltung aus:



14.3. Schalten von induktiven Lasten

Hierbei handelt es sich um einen besonders unangenehmen Betriebsfall (...denken wir nur an die vielen Elektromotoren mit Drehzahlregelungen, z. B. in Handbohrmaschinen....), denn sobald nach der Zündung des Thyristors der Strom im Stromkreis ansteigt, **speichert die Induktivität Energie in ihrem Magnetfeld**. Beim Nulldurchgang der Netzspannung möchte natürlich der Thyristor ausschalten. **Aber die Spule bemüht sich (wegen der gespeicherten Energie, die sie wieder loswerden möchte!) dagegen aufzukommen und versucht, den Strom in der alten Richtung weiterzutreiben.** Wie immer im Leben gibt so etwas Kollisionen und Ärger, der sich sehr schön durch die Simulation zeigen lässt.

Also schalten wir mal L = 100mH in Reihe zum Thyristor und drücken dann den Simulationsbutton. **Sehr gut ist zu sehen, wie der Thyristor jetzt nicht mehr wie vorher beim Nulldurchgang abschalten kann. Die Induktivität liefert nämlich noch solange Storm, bis die in ihr**

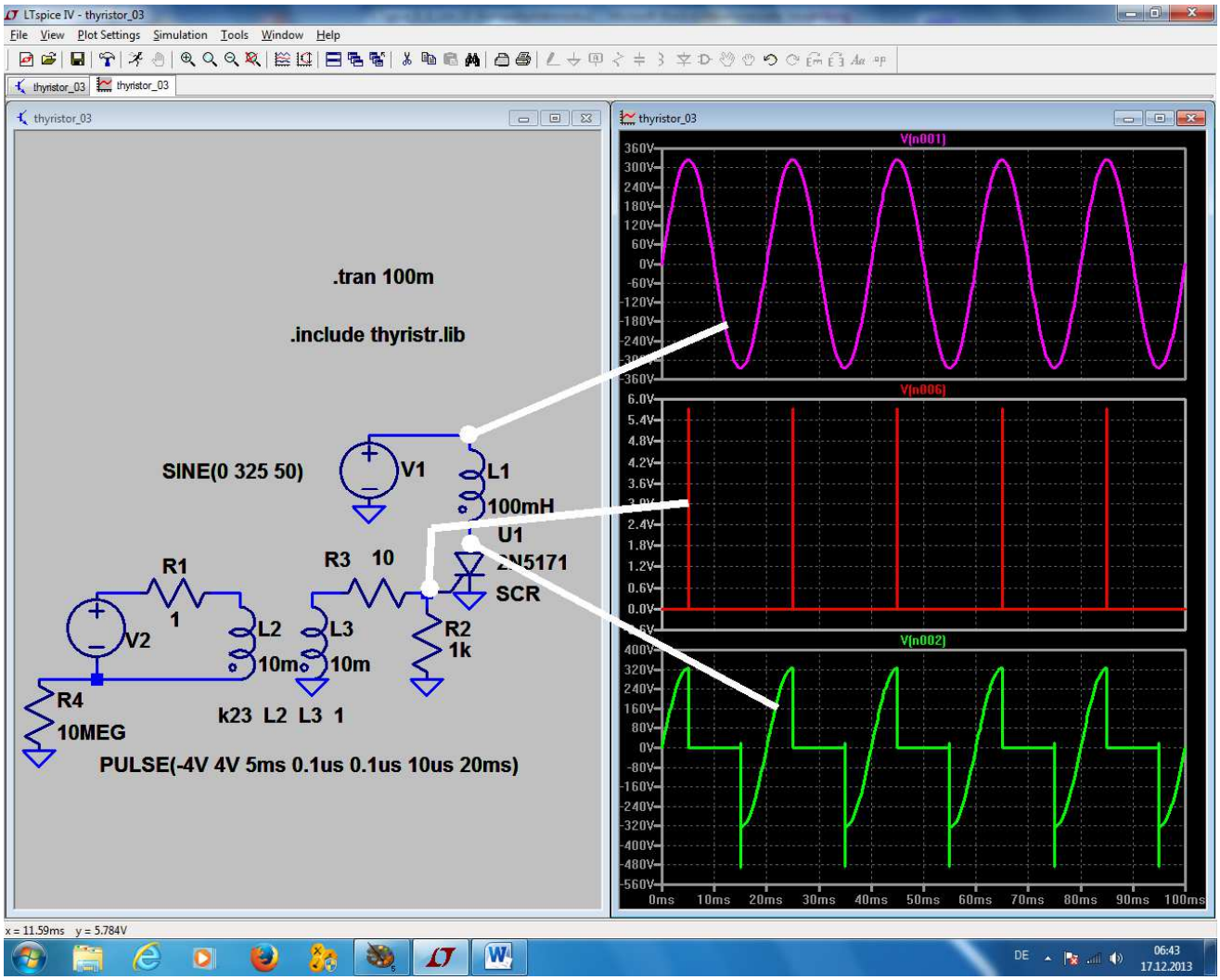


gespeicherte magnetische Energie verbraucht ist.

Ein kleiner Energierest erzeugt am Ende dann noch eine kurze gedämpfte Schwingung mit einer hübschen kleinen Spannungsspitze (Amplitude = ca. -500V).

14.4. Zündung des Thyristors über einen Gate-Transformator

In der Praxis fordert man sehr oft eine **galvanische Trennung zwischen Lastkreis und Zündkreis.** Deshalb erfolgt die Ansteuerung des Gates mit dem Zündimpuls über einen (meist fertig käuflichen) kleinen Transformator („Zünd-Trafo“). Also nehmen wir uns die vorige Schaltung her, bauen einen solchen Zündtrafo ein und sehen nach, ob alles noch wie gewünscht funktioniert.



Einige Details:

a) Die Zündimpuls-Spannungsquelle auf der Primärseite des Trafos ist nun galvanisch getrennt vom Lastkreis, aber SPICE verträgt keine „erdfreien Stromkreise“. Deshalb **muss** (als Gleichstromweg nach Masse bei der Startanalyse!) der **Widerstand R4 = 10 Mega-Ohm eingebaut werden**.

b) Was geändert werden **muss**, ist der **Verlauf des Zündimpulses, da der Trafo einen Gleichspannungsanteil im Zündsignal nicht übertragen kann**. Deshalb arbeiten wir nun mit einer symmetrischen Pulsspannung, die die Amplituden „+5V“ und „-5V“ aufweist.

c) Eine Spannungsquelle darf bekanntlich nie direkt an eine reine Induktivität gelegt werden -- das hatten wir schon. Deshalb wird der **Widerstand R1 mit 1 Ω zwischen Quelle und Primärseite des Trafos eingefügt.**

d) Wir verwenden den bekannten **idealen Transformator** und bauen ihn aus **zwei Induktivitäten „ind2“ mit je 10mH** auf. Da darf man natürlich die magnetische Kopplung nicht vergessen und **muss die Direktive** **k23 L2 L3 1** **für eine Kopplung von 100% im Schaltbild angeben.**

15. Neuntes Projekt: Digitale Schaltungssimulation

15.1. Was man vorher wissen sollte

Leider wird dieses Thema sowohl von der Online-Hilfe wie auch von den mitgelieferten Beispielen her etwas stiefmütterlich behandelt -- man muss also etwas Wissen und Erfahrung mitbringen und dann alle möglichen Mosaiksteine zusammentragen, um Erfolg zu haben.

Deshalb folgen gleich die wichtigsten Spielregeln:

Die mitgelieferte Symbolbibliothek **„[Digital]“** enthält nur eine Reihe von **„idealen Grundbausteinen“**. An ihnen fällt auf, dass sie alle

8 Anschlüsse („Pins“)

aufweisen. Dahinter steckt folgendes System:

a) Es gibt eine ganze Reihe Eingänge (z. B. beim AND-Gatter deren 5…). Nicht benötigte Eingänge werden einfach offen gelassen, denn dadurch sind sie automatisch abgeschaltet und nicht in die Simulation einbezogen.

b) Meist ist nicht nur der zugehörige logische Ausgang, sondern zusätzlich auch die Invertierung vorhanden.

c) Die logischen Pegel betragen „**Null Volt“ für die „Logisch Null“ und „+1 Volt“ für die „Logische Eins“**. Die intern programmierte Umschaltschwelle zwischen beiden Zuständen ist +0,5V.

Achtung: Wer auf andere Werte (z. B. den TTL-Pegel von 0V / +5V) umstellen will, der klickt mit der rechten Maustaste auf das Gatter-Symbol und trägt dann in der Zeile „value“ ein:

Vhigh=5V Vlow=0V

d) Ausgänge sollten nicht unbedingt „frei in der Luft hängen“. Sie können entweder mit einem Label versehen oder über einen Widerstand (empfehlenswert: R = 10kΩ) an Masse gelegt werden.

e) Man findet **keine Betriebsspannungsanschlüsse**, da mit „idealen Bausteinen“ simuliert wird.

Als Eingangssignal verwendet man entweder das „PULSE-“ oder das PWL-Signal.
Minimaler Spannungswert ist Null Volt, maximaler Spannungswert = +1Volt (…andere Werte: siehe oben…)
Als Anstiegs- und Abfallzeit reicht eine Zeit von 1 Nanosekunde.

Bei der Darstellung der Simulationsergebnisse sollte man sofort -- bevor überhaupt etwas im Ergebnisdiagramm zu sehen ist! -- mit der rechten Maustaste draufklicken und dann

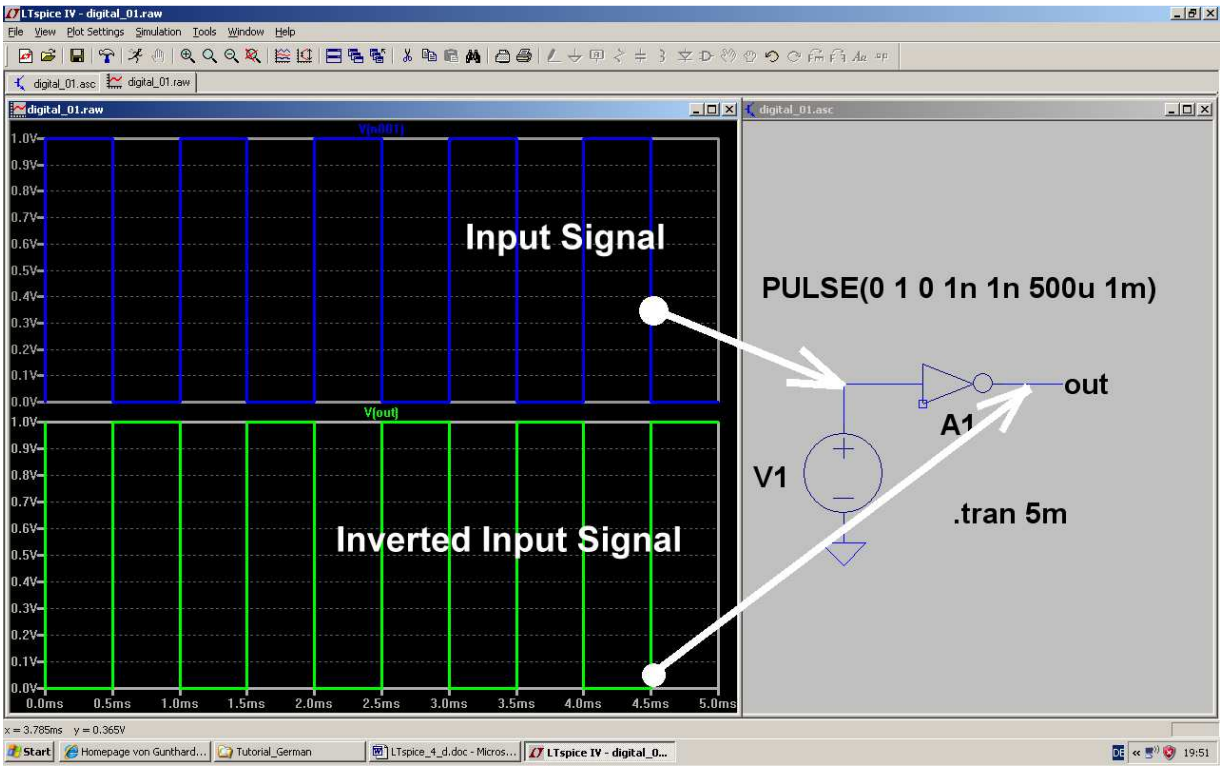
Add Plot Pane

wählen. Dadurch taucht ein weiteres Diagramm auf. Falls nötig, wiederholt man diese Prozedur solange, bis man genügend Diagramme für die Darstellung aller gewünschten Signale zur Verfügung hat. Wer das nicht tut, wird ganz schnell wahnsinnig….
Und die Darstellung einer Kurve in einem dieser Diagramme ist ja sehr einfach: rechts auf das Diagramm klicken, „Add Trace“ aktivieren und das gewünschte Signal auswählen.

15.2. Einfacher Anfang: die Umkehrstufe (= NOT oder Inverter)

Da fangen wir gleich mit der Schaltung an. Den Baustein holen wir uns als „**Digital / INV“** aus der Library und zur Ansteuerung verwenden wir eine symmetrische Rechteckspannung (Umin = 0V / Umax = +1V / Anstiegszeit = Abfallzeit = 1ns / Pulslänge = 500 Mikrosekunden / Periodendauer = 1ms). Simuliert wird von 0….5ms.

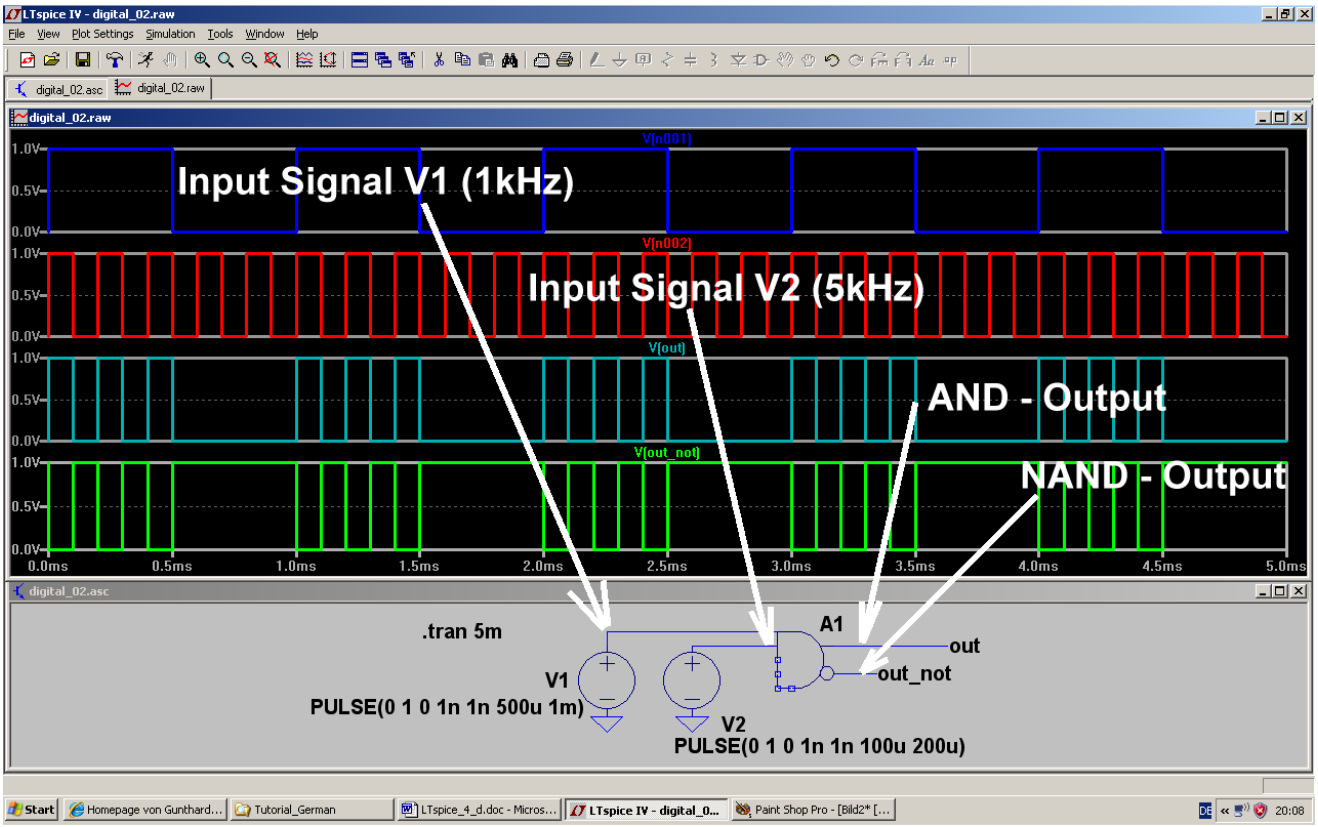
Am oberen Eingang wird angesteuert, der andere wird offen gelassen. Der Ausgang erhält den Label „out“ .Wie erwähnt, öffnen wir über „**Add Plot Plane**“ zwei Ergebnisdiagramme und stellen darin die Eingangs- bzw. die Ausgangsspannung dar. So sieht das Ergebnis aus, wenn man „**Tile vertically**“ im Menü „**Window**“ wählt:



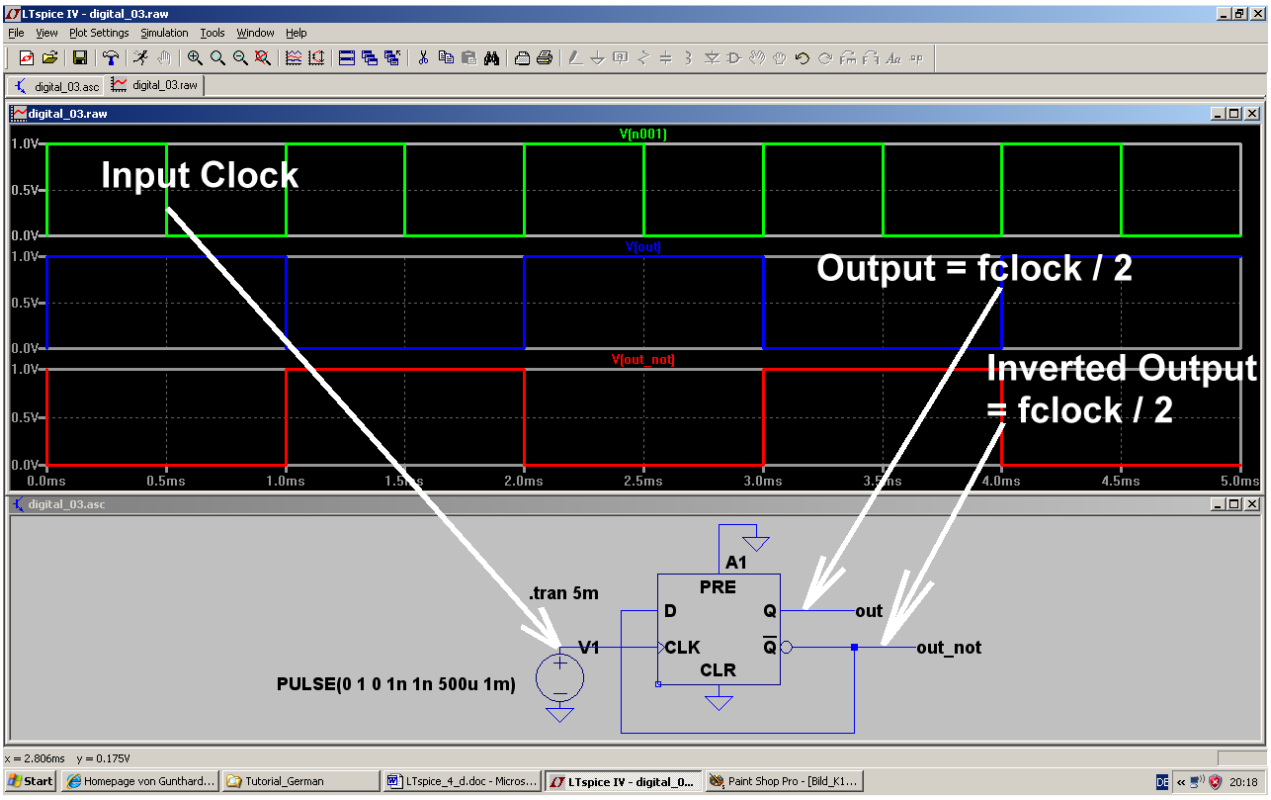
15.3. Der AND-Baustein

Wir wollen eine UND-Verknüpfung mit 2 Eingängen realisieren. Am einen Eingang liegt ein Digitalsignal mit f = 1kHz, am anderen dagegen eines mit f = 5kHz. Simuliert wird von 0….5ms. Wie zu Beginn des Kapitels erwähnt, müssen wir dazu die drei zusätzlich vorhandenen, aber nicht benötigten Eingänge offen lassen. Den vorhandenen beiden Ausgängen wurden wieder **Label zugewiesen**.

So sollte die Simulation aussehen:



15.4. Das D-Flipflop



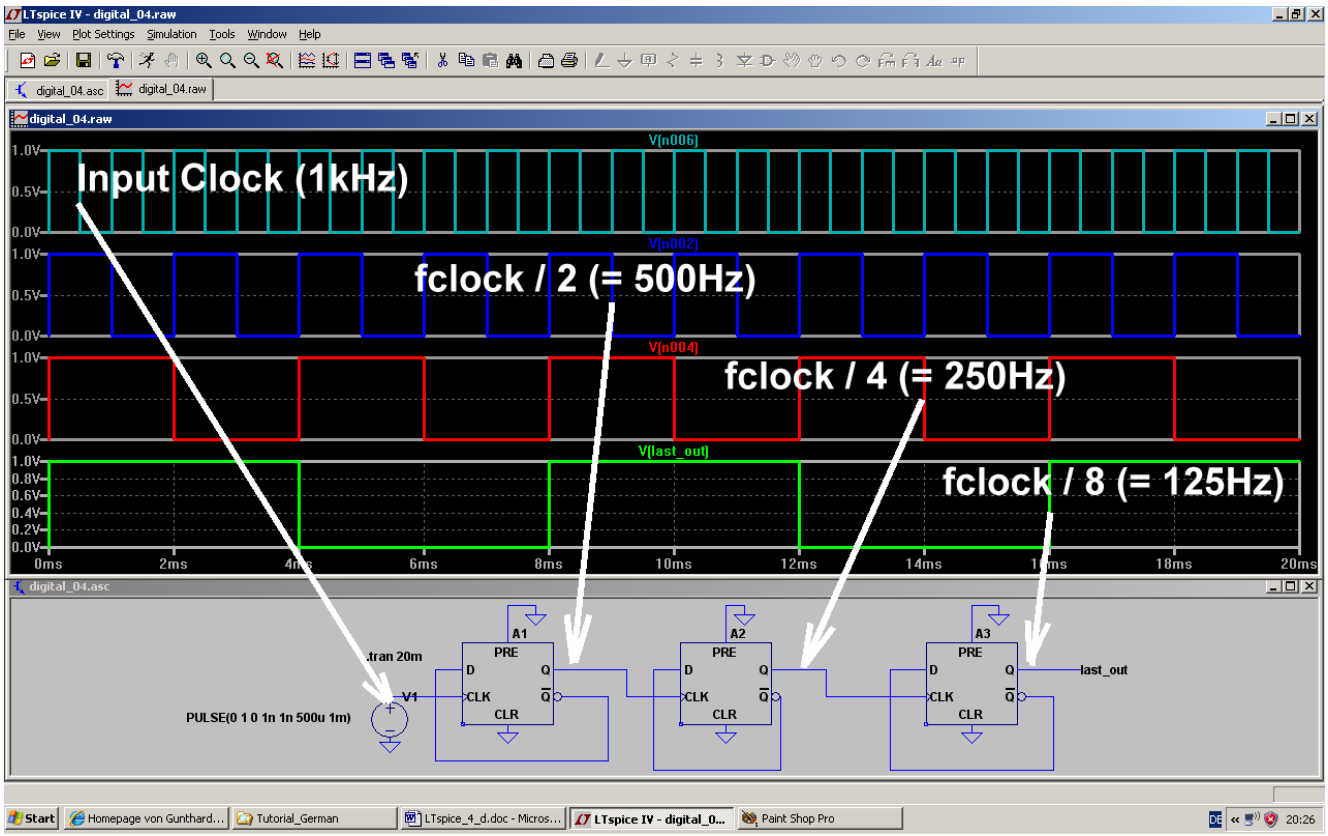
Die Schaltung stellt den bekannten Binär-Frequenzteiler dar, bei dem am Ausgang die halbe Eingangsfrequenz beobachtet werden kann. Dazu wird der invertierte Q-Ausgang mit dem D-Eingang verbunden. Die Input Clock-Frequenz beträgt 1kHz.

Die Signaleingänge „**Preset“ und „Clear“ dürfen nicht aktiv werden** und liegen deshalb an Masse.

Bei jeder positiven Flanke des Clocksignals wechseln die Ausgänge ihre Zustände.

15.5. Dreistufiger Frequenzteiler mit D-Flipflops

Das ist jetzt nur noch reine Arbeit, aber kein Problem mehr, denn es werden einfach zwei weitere identische Stufen nachgeschaltet.



Die Simulationszeit wurde auf 20ms erhöht, der Eingangs-Clock (= Input Clock Signal) besteht weiterhin aus einem Rechtecksignal mit der Frequenz f = 1kHz.

Wer dieses Kapitel als etwas dürftig empfindet, der hat recht -- aber es fehlen einfach die vielen, vielen Bausteine der TTL – oder CMOS-Serie, die das alles interessanter oder praxisnäher machen.

Da bleibt nur ein Weg, nämlich folgender Link:

http://tech.groups.yahoo.com/group/LTspice/files/%20Lib/Digital%2074HCTxxx/

Dahinter versteckt die LTspice-Usergroup auf Yahoo und dort sind auch die von Helmut Sennewald selbst erstellten Bibliotheken zur Simulation der gängigen 74xx-Bausteine abgelegt (Helmut: Danke für diese enorme Arbeit!).

ABER:
Das ist eine Community (= Gemeinschaft), in die man erst nach kostenloser Registrierung und über ein Passwort reinkommt. Dafür herrscht dort ein reger Online-Betrieb mit pausenlosen Fragen und Antworten und Hilfen und Hilferufen und Angeboten und Hinweisen und…und….und….
Ist aber faszinierend und dadurch wird nicht nur die Lösung irgendwelcher eigener Probleme erleichtert bzw. ermöglicht, sondern auch neben der Freude an der Technik das Gefühl vermittelt, unter Freunden und in guter Gesellschaft und in guten Händen zu sein.

Also: nichts wie los und rein…

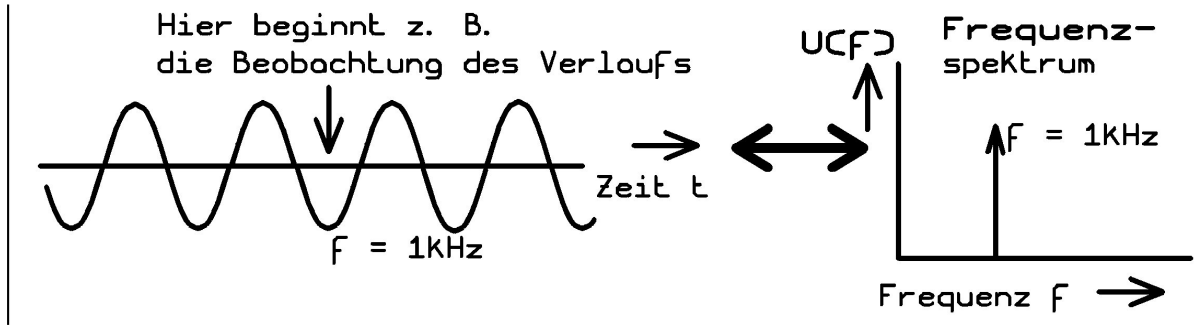
16. Zehntes Projekt: Signalformen und Oberwellen

16.1. Grundlagen

Selbst viele Laien wissen sofort die Antwort auf die Frage: „Wie sieht die Spannung aus, die in jedem Haus aus der Steckdose kommt?“ Sie lautet immer: „Na klar, ein Sinus mit 230 Volt Effektivwert und einer Frequenz von 50 Hz“. Aber bei der nächsten Frage:

„Welche verschiedenen Frequenzen stecken eigentlich in diesem Steckdosensignal?“

kriegt man selbst von Fachleuten oft die falsche Antwort. Das liegt daran, dass man erstens genau die „Mess- und Beobachtungsbedingungen“ beachten muss, denn damit ändert sich direkt die Lösung. Zweitens muss man sozusagen die große Lupe nehmen und das Signal mit der Lupe absuchen (...wie einen Hund nach Flöhen..) und auf jede Kleinigkeit oder Unregelmäßigkeit achten, denn die sind enorm wichtig.....Formulieren wir es mal ganz korrekt:



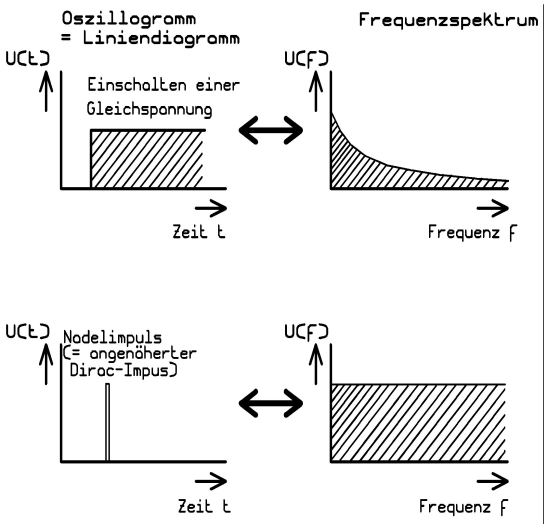
Nur eine rein sinusförmige Spannung (hier: f = 1 kHz), die vor sehr langer Zeit eingeschaltet wurde und die in absehbarer Zeit nicht aufhört (...also weiterläuft bis zu den Flügen von Raumschiff Enterprise im Jahr 2150....) besteht nur aus einer einzigen Frequenz.

Da diese Formulierung so gar nicht zu dem passt, was man kennt und weiß und täglich sieht, folgt hier die Lösung des Rätsels: Man muss nämlich sorgfältig zwischen „**Schaltvorgängen**“ und „**Verzerrungen der Kurvenform bei periodischen Signalen**“ unterscheiden.

Schaltvorgänge

Jede **einmalige Änderung** (= Einschalten, Ausschalten, Aufdrehen, Kleiner machen, kurz unterbrechen.....) erzeugt ein **einmaliges „Störsignal“ oder „Störspektrum“**, das vom Entstehungsort wegwandert. Das kennt jeder von uns als Knacken im Radio, wenn er plötzlich das Licht im Raum einschaltet....man hat das bloß noch nie so gesehen oder beachtet! Entscheidend in diesem Störsignal ist die Tatsache, dass ***der komplette Frequenzbereich einschließlich der sehr hohen Frequenzen während dieser kurzen Zeit lückenlos mit Signalen belegt ist***. Die tiefen Frequenzanteile breiten sich dann natürlich über Kabelverbindungen aus, die hohen Anteile werden dagegen schon von den als Antennen wirkenden Verbindungsdrähten der Schaltung als elektromagnetische Wellen abgestrahlt. Und das ist der erwähnte Knacks im Radio, der dann selbst in einem batteriebetriebenen Kofferradio zu hören ist! Glücklicherweise sinken die Amplituden der hohen Frequenzanteile bei den üblichen Schaltvorgängen sehr schnell ab, sonst könnten wir überhaupt nicht drahtlos kommunizieren.

Die Geschwindigkeit der Änderung beim „Schaltvorgang bestimmt wesentlich diese Amplitudenverteilung und bestätigt anschaulich das Schwäbische Prinzip des „ruhigen Angehens einer Sache“ (Schwäbisch: = „no net hudla“) -- und wenn sich etwas sehr langsam ändert, können wir diese Effekte fast gar nicht nachweisen. Aber es gibt ein Signal, in dem alle Frequenzen enthalten sind und vor dem sollte man sich auch in abgeschwächter Form hüten: es ist der „**Dirac-Impuls**“ mit unendlich kurzer Pulsdauer und unendlich großer Amplitude, denn er kann geradezu sagenhaft stören.



Hier zwei Beispiele:

Links die Oszillogramme („Liniendiagramme“) vom Einschalten einer Gleichspannung und vom Nadelimpuls, der den Dirac-Impuls annähern soll.

Rechts die zugehörigen Frequenzspektren, wobei für den Nadelimpuls eine sehr kleine Pulsdauer angenommen wurde. (Genauere Details folgen im Sampling-Kapitel).

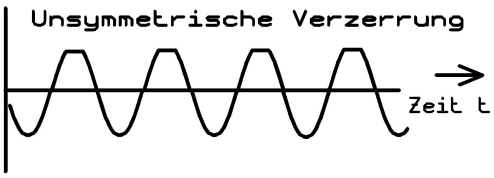
Man erkennt gut, dass es kein Entkommen vor diesen Störfrequenzen gibt, denn es existieren bei beiden Signalen **keine Lücken im Spektrum**!

Verzerrungen bei periodischen Signalen

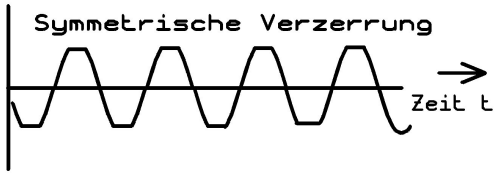
Da merken wir uns gleich den Schlüsselsatz:

Auch die allerkleinste, aber sich regelmäßig wiederholende Abweichung von der idealen Sinusform führt zum zusätzlichen Auftreten neuer Signale im Frequenzspektrum!

Diese neuen Frequenzen tragen den Fachausdruck „Harmonische“ oder „Oberwellen“ und sind stets **ganzzahlige Vielfache der Grundfrequenz** (Im Klartext heißt das: man findet nun plötzlich auch die doppelte, dreifache, vierfache, fünffache, sechsfache....Frequenz im Spektrum!).
Dabei gilt noch eine weitere wichtige Spielregel:



Ist nur eine Halbwelle von der Verzerrung betroffen ODER die positive und die negative Halbwelle sind unterschiedlich verformt, dann spricht man von ***unsymmetrischer*** Verzerrung. Man findet dann ***alle Oberwellen*** im Spektrum.



Ist das ursprünglich ideale Sinussignal aber ***symmetrisch*** verzerrt (= positive und negative Halbwelle sind exakt in derselben Weise verändert), dann gibt es ***keine geraden***, sondern nur ***ungerade Oberwellen*** (= außer der Grundfrequenz beobachtet man zusätzlich nur die dreifache, die fünffache, die siebenfache, die neunfache.....Grundfrequenz).

Abschließender Hinweis:
Sobald man am idealen Sinus irgendwas im ***Rhythmus eines anderen Signals verändert*** (= Amplitude, Phase oder Frequenz), gelangt man zu den entsprechenden

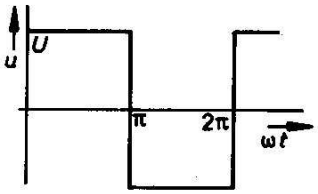
Modulationsarten.

Auch hier müssen folglich neue Signale entstehen. Aber damit sind wir bei einem anderen Thema…

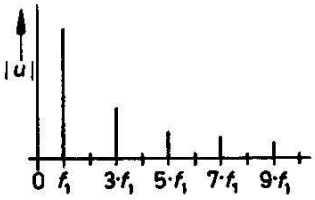
Auf dem nächsten Blatt folgt noch eine kleine Zusammenstellung einiger Signalformen und der zugehörigen Spektren:

Spektrum: Absolutwerte der auftretenden Frequenzen

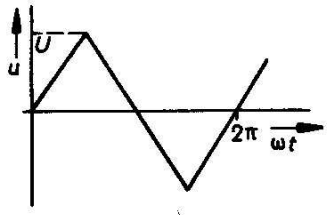
Rechteckspannung



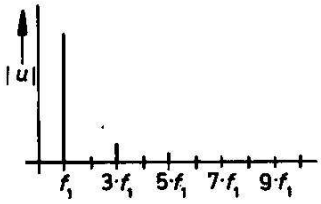
$$f(t) = \frac{4 \cdot U}{\pi}\left(\sin \omega \cdot t + \ldots + \frac{1}{3}\sin 3\,\omega t + \frac{1}{5}\sin 5\,\omega t + \ldots\right)$$



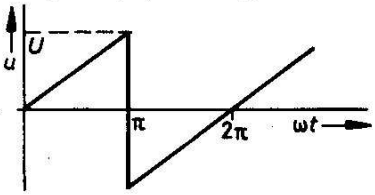
Dreieckspannung



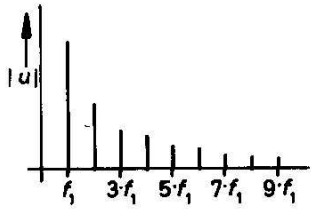
$$f(t) = \frac{8\,U}{\pi^2}\left(\sin \omega\, t - \ldots + \frac{1}{3^2}\sin 3\,\omega t + \frac{1}{5^2}\sin 5\,\omega t - \ldots\right)$$



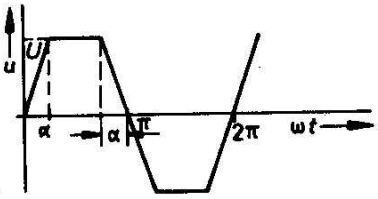
Sägezahnspannung



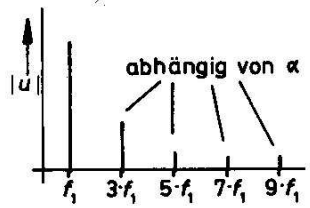
$$f(t) = \frac{2\,U}{\pi}\left(\sin \omega t - \frac{1}{2}\sin 2\,\omega t + \ldots + \frac{1}{3}\sin 3\,\omega t - \ldots +\right)$$



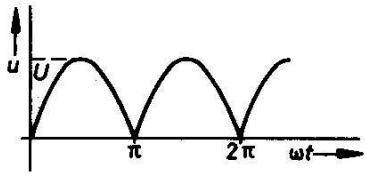
Trapezspannung



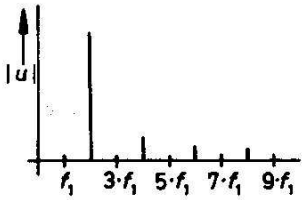
$$f(t) = \frac{4 \cdot U}{\pi \cdot \alpha}\left(\sin \alpha \cdot \underset{\text{tot}}{\sin \omega t} + \ldots + \frac{\sin 3\,\alpha}{3^2} \cdot \sin 3\,\omega t + \ldots + \frac{\sin 5\,\alpha}{5^2}\sin 5\,\omega t + \ldots\right)$$



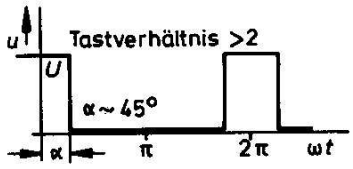
Doppelweg-Gleichrichtung



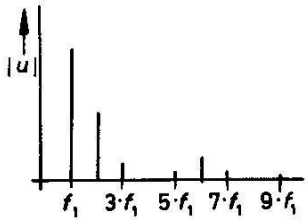
$$f(t) = \frac{4\,U}{\pi}\left(\frac{1}{2} - \frac{1}{1 \cdot 3}\cos 2\,\omega t - \ldots - \frac{1}{3 \cdot 5}\cos 4\,\omega t - \ldots - \frac{1}{5 \cdot 7}\cos 6\,\omega t - \ldots\right)$$



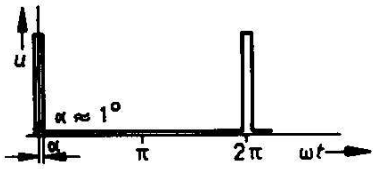
Rechteckspannung
Tastverhältnis > 2

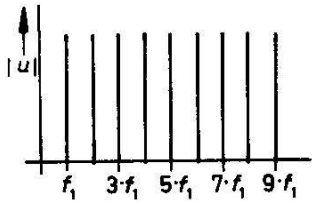


$$f(t) = \frac{2\,U}{\pi}\left(\frac{\alpha}{2} + \frac{\sin \alpha}{1}\cos \omega t + \ldots + \frac{\sin 2\,\alpha}{2}\cos 2\,\omega t + \ldots + \frac{\sin 3\,\alpha}{3}\cos 3\,\omega t + \ldots\right)$$



Nadelimpulse





16.2. Simulation des Spektrums eines einmaligen Pulses

Wir wollen mal prüfen, ob die Behauptung aus dem Grundlagenteil stimmt. Dazu verwenden wir ein Signal mit folgenden Daten:

Umin = 0V / Umax = 1V / Anstiegs- und Abfallzeit je 1 Mikrosekunde / Pulsdauer = 1Millisekunde.

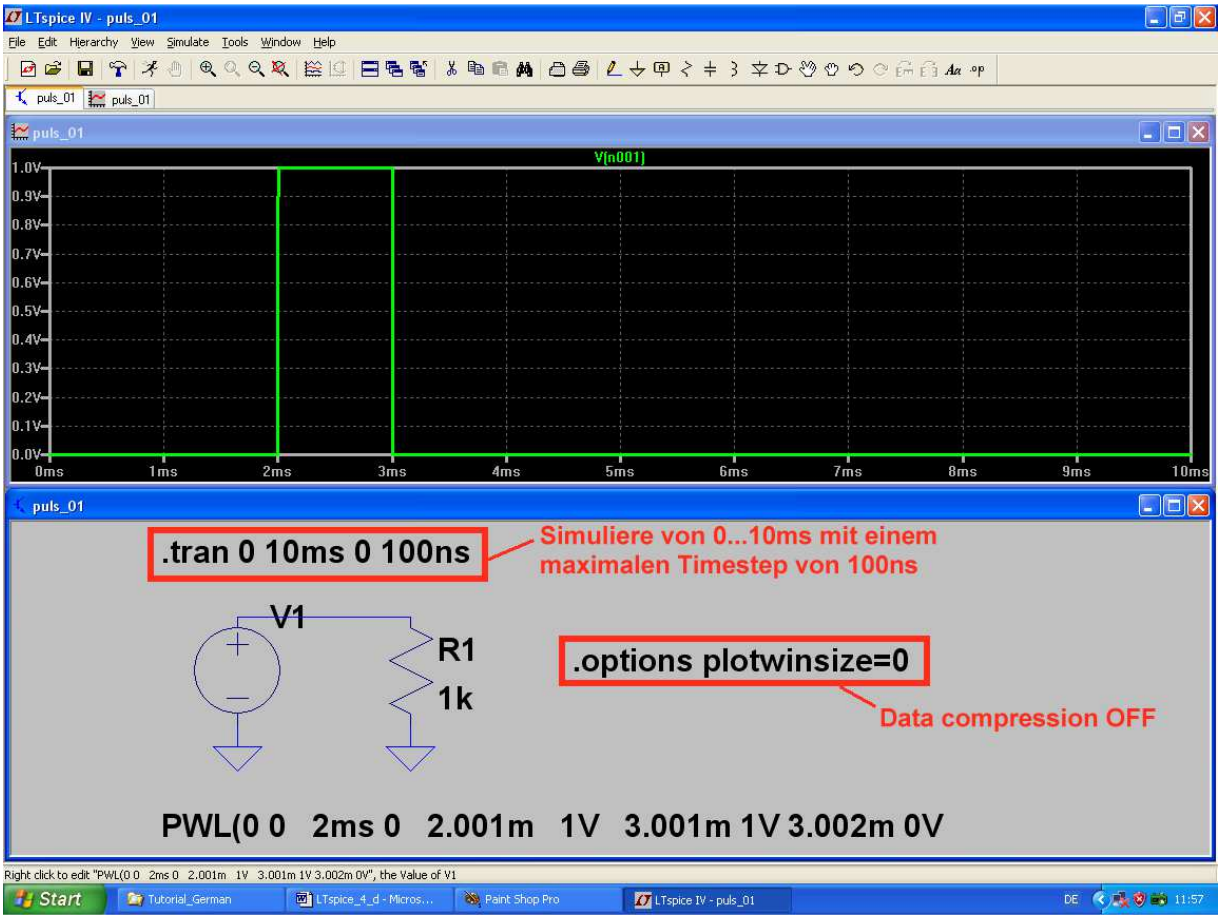
Zusätzlich bauen wir eine „**Delay Time“ nach dem Start von 2 Millisekunden ein** und simulieren bis 10 Millisekunden im Zeitbereich die Spannung am Lastwiderstand R1. Der größte „Time Step“ betrage 100 Nanosekunden. Für die FFT stehen uns dann 10ms / 100ns = 100 000 echte Samples zur Verfügung. Die Datenkompression schalten wir (mit Rücksicht auf die geplante FFT) durch die Direktive

.option plotwinsize=0

aus.
Den Signalverlauf bilden wir mit einer „**PWL“-Quelle** nach, der folgende Wertepaare zur Verfügung gestellt werden:

Null Volt	**bei Zeitpunkt Null**
Null Volt	**bei Zeitpunkt 2 Millisekunden**
1 Volt	**bei Zeitpunkt 2,001 Millisekunden**
1 Volt	**bei Zeitpunkt 3,001 Millisekunden**
Null Volt	**bei Zeitpunkt 3,002 Millisekunden**

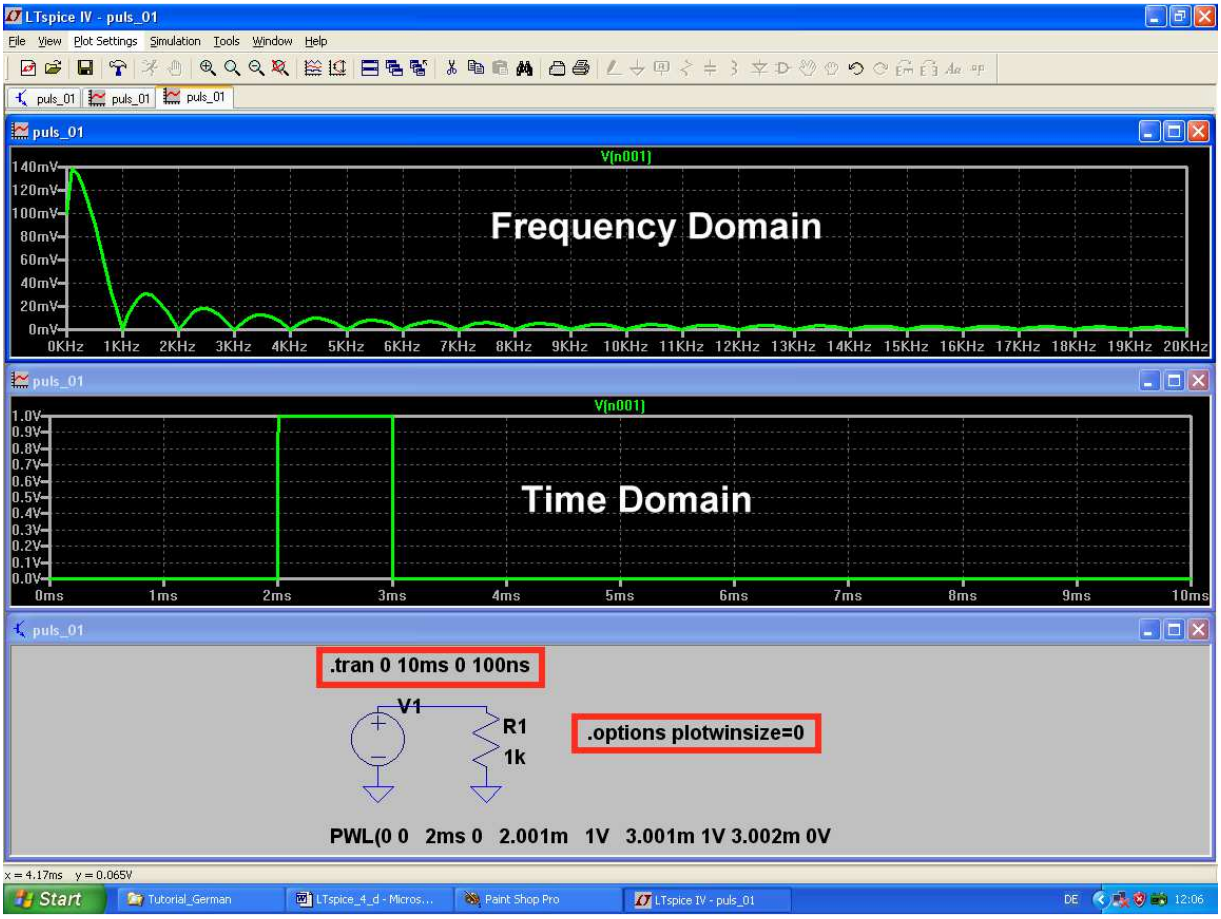
Simuliert man anschließend und wählt hinter „Window“ die Option „**Tile Horizontally**“, dann muss Folgendes zu sehen sein:



Jetzt klickt man mit der rechten Maustaste auf das Ergebnisdiagramm, ruft ganz unten „FFT“ auf und stellt darin „**65536 Sample Points**“ ein. Nach einem Klick auf OK wird anschließend beim Ergebnisdiagramm Folgendes geändert:

a) Fährt man mit dem Cursor auf die Zahlenteilung der **Senkrechten Diagramm-Achse und klickt**, dann kann man auf „**Linear**“ umstellen

b) Bei der **Waagrechten Achse** sorgen wir nach derselben Methode für eine **lineare Teilung, eine Startfrequenz von 0 Hz, einen Tick von 1kHz und eine Stoppfrequenz von 20kHz.**
Und zur Krönung wird nochmals unter „Window“ die Option „**Tile Horizontally**“ gewählt. Dann sehen wir dieses Bild:



Bei dem so erzeugten Frequenzspektrum finden wir drei wichtige und bekannte Tatsachen bestätigt:

a) **es gibt bei einem nichtperiodischen Signal keine einzelnen Spektrallinien, sondern der komplette Frequenzbereich ist (bis auf die Nullstellen) lückenlos mit Energie belegt.**
Deshalb zeigt das obere Diagramm den Verlauf der „Spektralen Leistungsdichte“.

b) Ihre Amplitude verläuft nach einer „sin x / x“-Funktion.

c) Die Nullstellen im Leistungsdichte-Verlauf liegen IMMER bei Vielfachen vom Kehrwert der Pulslänge, also bei

$$f_{NULL} = N \bullet \left(\frac{1}{t_{PULSE}} \right)$$

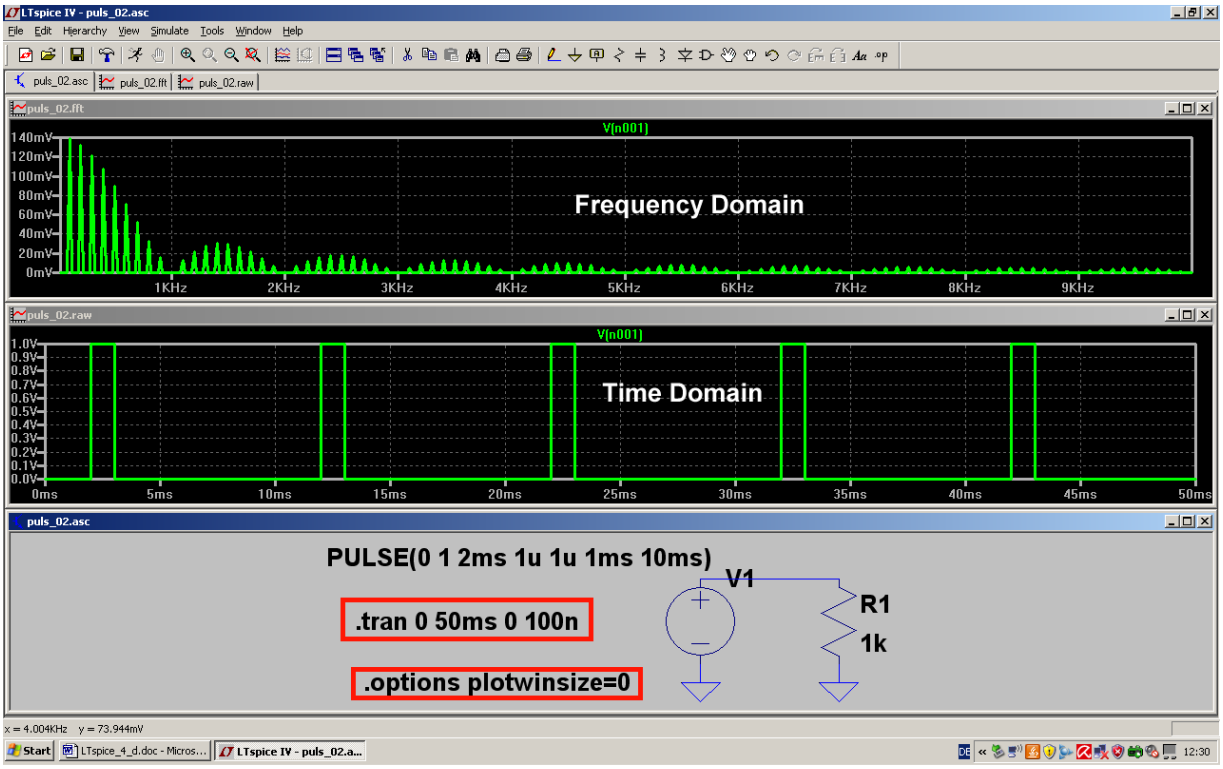
(Bei unserer Pulslänge von 1ms liegen die Nullstellen deshalb stets bei n x 1kHz)

Je kürzer der Puls wird, desto weiter wird folglich die erste Nullstelle zu höheren Frequenzen hin verschoben.

16.3. Simulation des Frequenzspektrums bei einem periodischen Pulssignal

Wir benützen denselben Spannungsverlauf wie vorher, lassen ihn aber **100x in der Sekunde wiederholen.** Deshalb müssen wir die PWL-Quelle durch eine **Pulsspannungsquelle** ersetzen. Außerdem erhöhen wir die **Simulationszeit auf 50ms, damit wir auf eine Frequenzauflösung von 1 / 50ms = 20Hz kommen.**
Dann wird die Spannung an R1 erst (ohne Datenkompression) simuliert und dann FFT-transformiert.
Für die FFT-Einstellungen und die Achsenteilungen des Spektrums benützen wir folgende Werte:
enkrechte Achse: Linear

Waagrechte Achse: Linear / Start = 0kHz / Tick = 1kHz / Stopp = 10kHz



Man erkennt:

a) Da es sich um ein **periodisches Signal** handelt, bekommen wir nun ein

„Linienspektrum“.

Es handelt sich um die berühmte „Grundwelle und ihre Oberwellen“. Die Grundfrequenz beträgt 100Hz, also haben die Einzellinien ebenfalls einen gegenseitigen Abstand von 100Hz.

b) Unsere Signalform (= Puls mit 1ms Länge) sorgt wie vorhin dafür, dass die Hüllkurve erneut einen „sin x / x „ – Verlauf aufweist. Die erste Nullstelle der Hüllkurve liegt dann wieder beim Wert

$$f_{NULL} = \frac{1}{\text{Pulslänge}}$$

und alle weiteren bei Vielfachen dieses Wertes (= 1kHz).

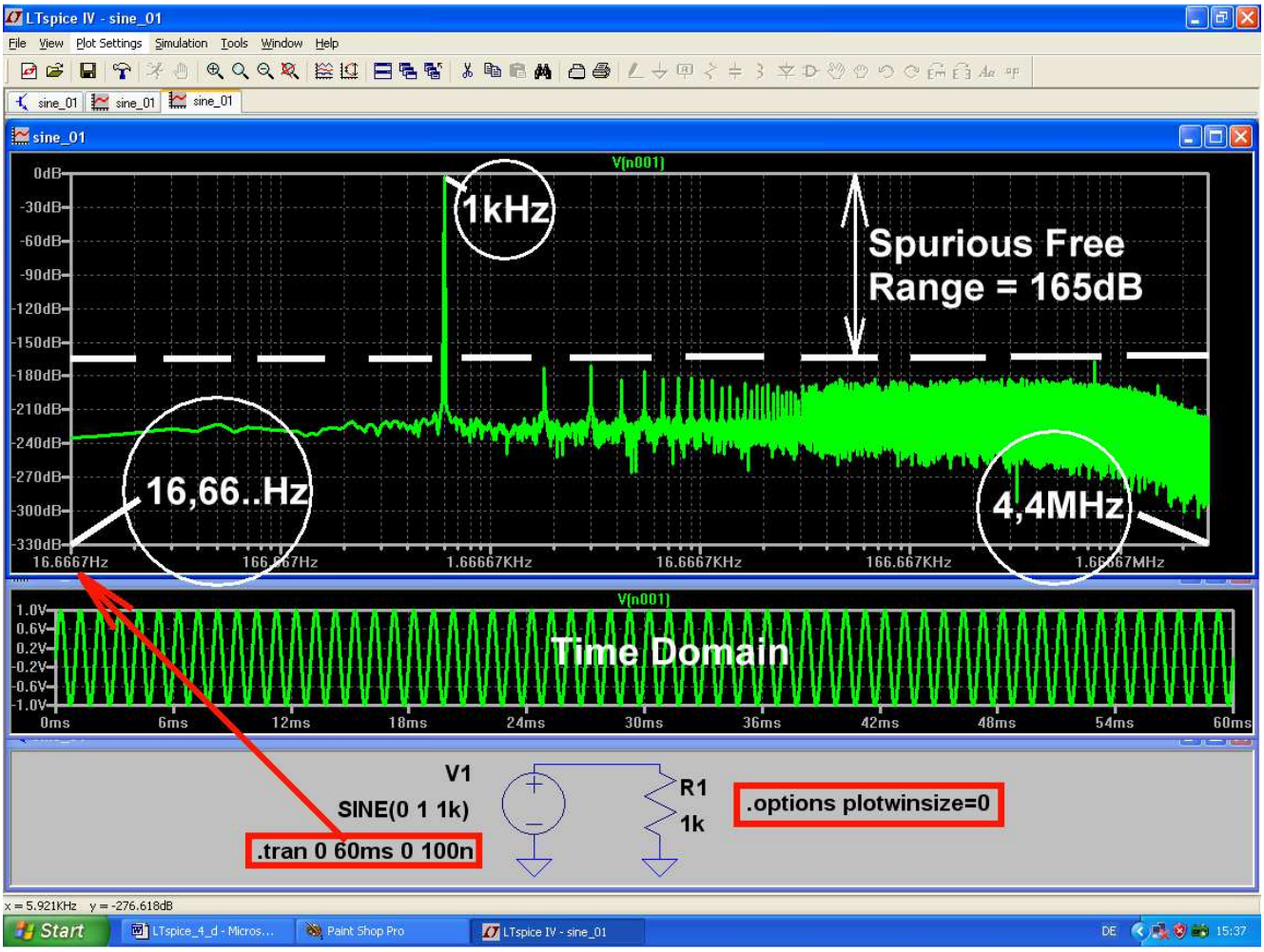
16.4. Der ideale Sinus

Sehen wir uns mal an, ob sich die Behauptung aus dem Grundlagenteil:

„Nur ein Sinussignal, bei dem weder Anfang noch Ende zu erkennen sind, enthält nur eine einzige Spektrallinie!“

beweisen lässt.

Simulieren wir deshalb eine 1kHz-Sinusspannung (Spitzenwert = 1V) für 60ms mit einem maximalen Timestep von 100ns. Damit erhalten wir 600 000 Samples und finden garantiert jede Unebenheit im Spektrum, da wir bei der FFT nun mit 524 288 Werten loslegen können. Auch die Datenkompression wird abgeschaltet. Und die Simulationszeit ist ein ganzzahliges Vielfaches der Periodendauer des Signals (Siehe Kapitel 5 / FFT), so dass auch diese Vorbedingung für ein hochwertiges FFT-Ergebnis stimmt.



Nun glaubt es Jeder: im Frequenzbereich von 17Hz bis 4,4MHz sind jetzt alle unerwünschten oder nicht echten Signalanteile oder Quantisierungsgeräusche oder…oder… um mindestens 165dB abgeschwächt und es bleibt nur die 1kHz-Linie übrig. Das ist nun wirklich ein nahezu perfekter Sinus…

16.5. Der einseitig begrenzte Sinus

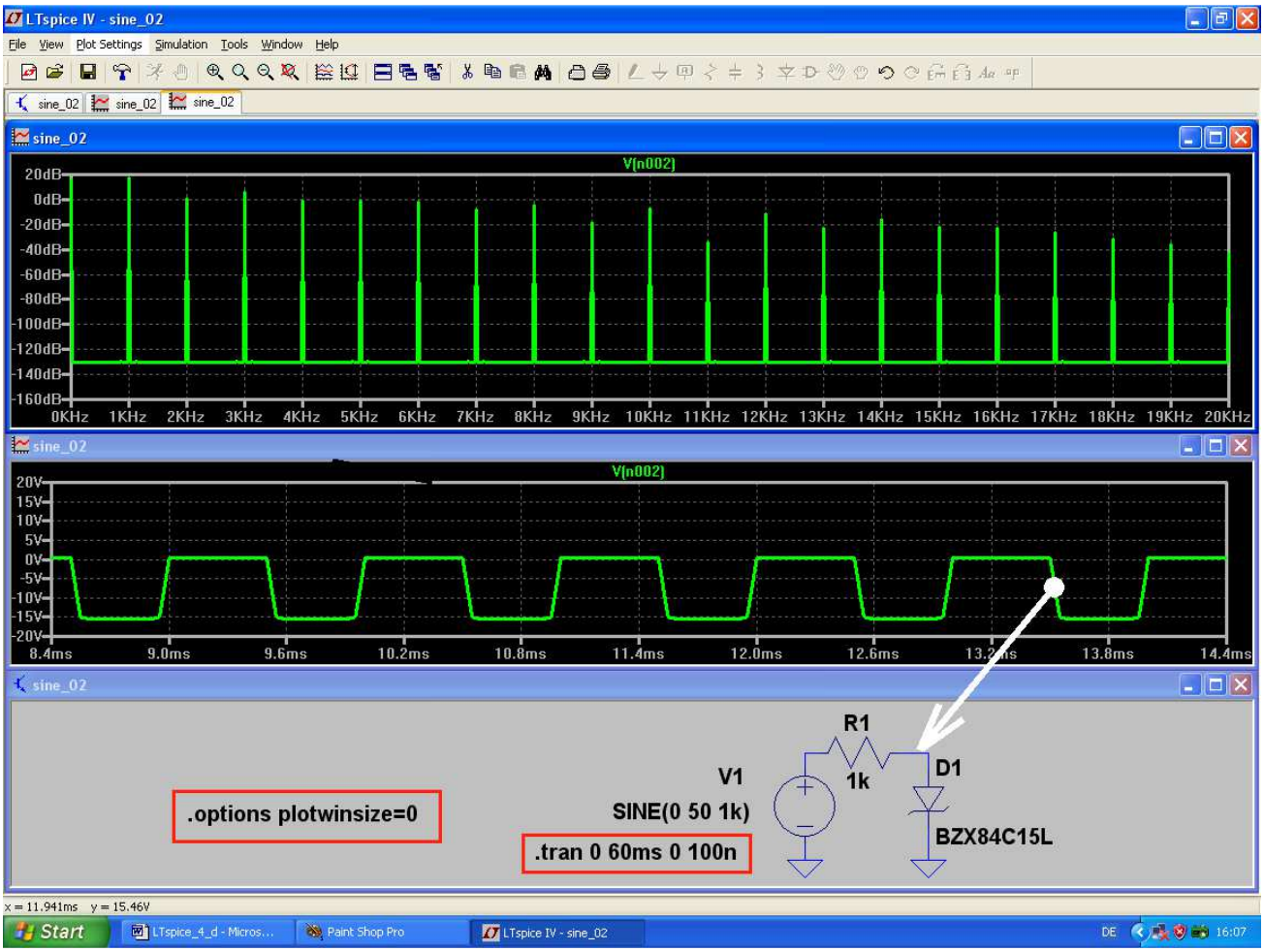
Das machen wir ganz brutal mit einer Zenerdiode, vor die ein Widerstand geschaltet wird. Die positive Halbwelle wird dadurch auf die Schwellspannung der „normalen Siliziumdiode mit 0,7V“ begrenzt. Und bei der negativen Halbwelle sorgen wir für eine Begrenzung auf ca. -15V durch die Verwendung des passenden Z-Diodentyps BZX84C15L.

Dazu muss man erst das Diodensymbol aus der Menüleiste holen und platzieren. Anschließend klickt man rechts auf das Schaltzeichen und öffnet das Menü „Pick new diode“. Dort findet sich die gesuchte Z-Diode.

Der Spitzenwert der sinusförmigen Eingangsspannung wird auf 50V erhöht, um die unsymmetrische Begrenzung deutlich sichtbar zu machen.

Simuliert wird wieder bis 60ms mit einem maximalen Timestep von 100ns und die Datenkompression ausgeschaltet. Das ergibt wieder 600 000 Samples für die FFT.

Dargestellt wird aber nur der Frequenzbereich von 0 bis 20kHz, um die geraden und ungeraden Harmonischen zu zeigen:



16.6. Der symmetrisch begrenzte Sinus

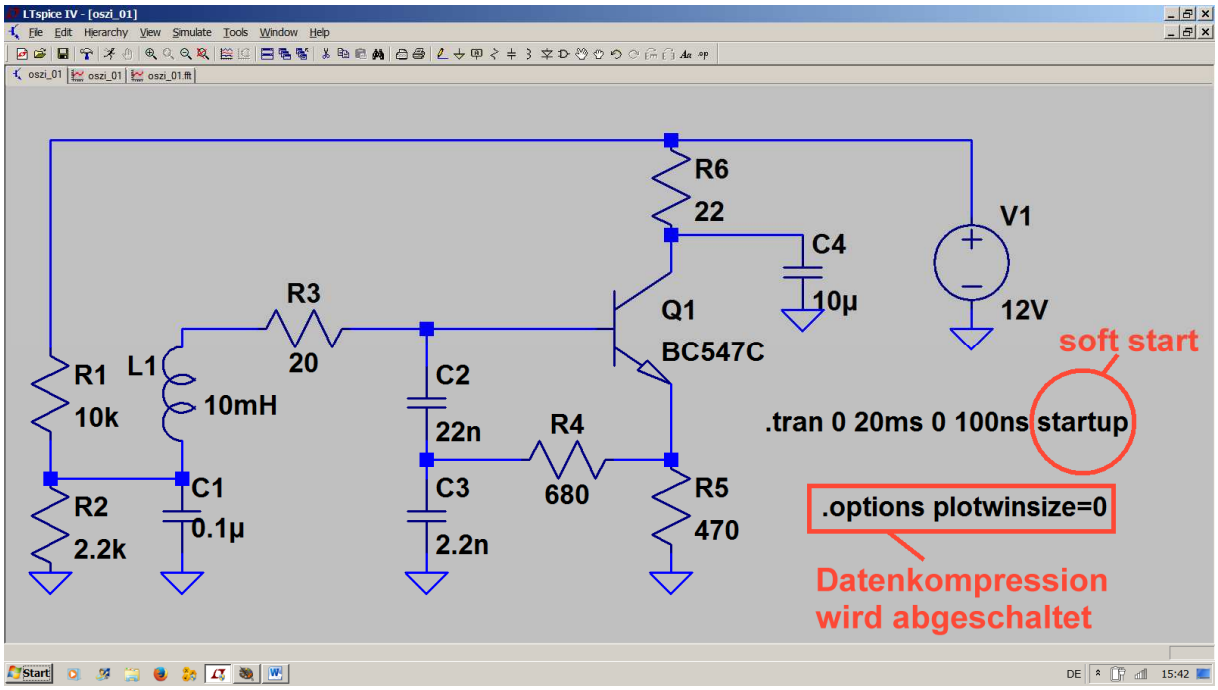
Eine sehr einfache Übung: wir schalten zwei gleiche Z-Dioden gegensinnig in Reihe und verwenden denselben Diodentyp BZX84C15L.

Wieder wird ein Sinus mit f = 1kHz / Upeak = 50V angelegt und bis 60ms mit einem maximalen Timestep von 100ns simuliert. Die Datenkompression ist abgeschaltet.

Auch beim Spektrum gilt dieselbe Kalibrierung wie im vorigen Beispiel und nun erkennt man sehr schön, dass diesmal nur ungeradzahlige Harmonische auftreten.

17. Elftes Projekt: Untersuchung eines Sinus-Oszillators

Wir wollen eine Sinus-Oszillatorschaltung (Colpitts - Oszillator mit npn - Transistor BC547C in Kollektorschaltung) untersuchen und erstellen dazu diesen Stromlaufplan:

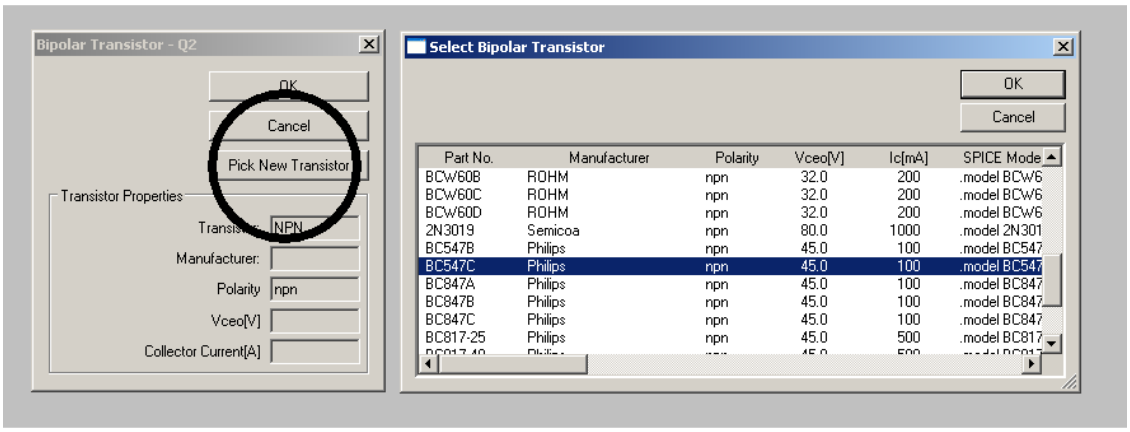


Bei den „**Simulation Settings**“ wählen wir „Time domain (Transient)“ und einen Simulationszeitraum von **0….20 Millisekunden**. Das ergibt später im Spektrum eine **Frequenzauflösung von 1/20ms = 50Hz.**

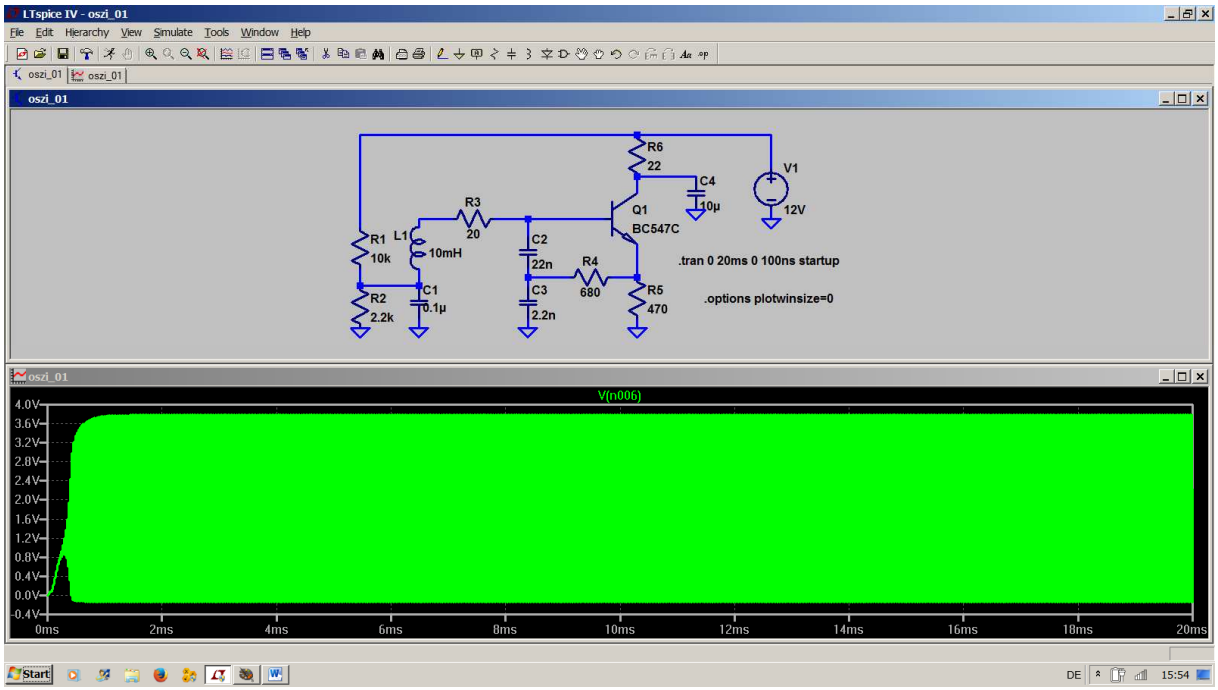
Bei der „**Maximum Step Size**“ geben wir den Wert **100ns** vor, damit unser SPICE das Anschwingen der Schaltung während der ersten 10 Mikrosekunden nicht übersieht. Außerdem erhalten wir damit eine echte Samplezahl von 20ms / 100ns = 200 000 und können bei der FFT mit mindestens 131072 Werten arbeiten. Dazu müssen wir aber auch die **Datenkompression** mit der Direktive **„.options plotwinsize=0“** ausschalten.

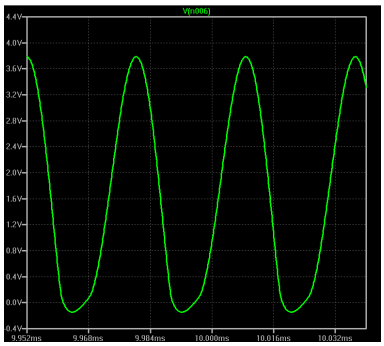
Damit die Schaltung in der Simulation überhaupt schwingt, müssen wir ihr einen „Schubs“ erteilen. Deshalb hängen wir die Aufforderung „**startup**“ an das Simulationskommando. Damit wird die Versorgungsspannung in 10 Mikrosekunden linear von Null auf +12 Volt hochgefahren und dadurch die Oszillatorschaltung mit einem „soft start“ zum Anschwingen angeregt.

Beim Transistor holen wir zunächst aus der Bibliothek das Symbol „**npn**“ und setzen es in der Schaltung ab. Sobald wir nun mit der Maus auf das Symbol rollen und rechts klicken, kommen wir an das Menü „**Pick New Transistor**“ heran. Darin findet sich der Typ „**BC547C**“, und mit OK wird er in unsere Schaltung übernommen.



Nun wird simuliert und dann sollte folgendes Ergebnis erscheinen, wenn man mit der „Prüfspitze“ den Emitter des Transistors berührt:





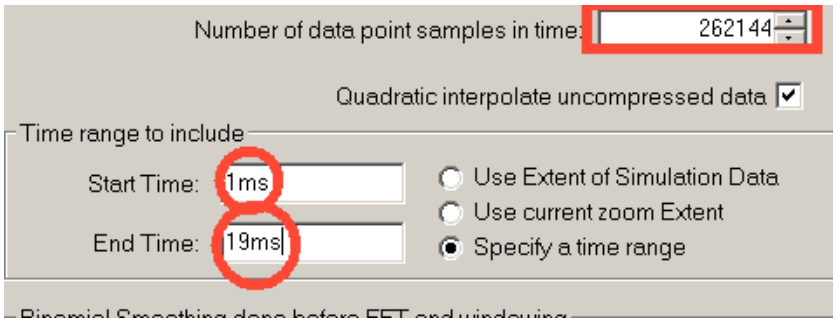
Interessant ist es nun, nur einige Schwingungen heraus zu zoomen. Dann kann man nämlich nicht nur die Kurvenform auf Verzerrungen kontrollieren, sondern auch (z. B. mit Hilfe des Cursors) **über die Periodendauer des Signals seine Frequenz bestimmen**:

Eine Messung der Periodendauer auf diese Art ergibt 28 Mikrosekunden. Dazu gehört eine Frequenz von 35,7kHz.

Bitte nachprüfen!

Aber:
Die Kurvenform ist noch nicht so richtig schön sinusförmig und das deutet darauf hin, dass die „Mitkopplung“ noch zu stark ist.

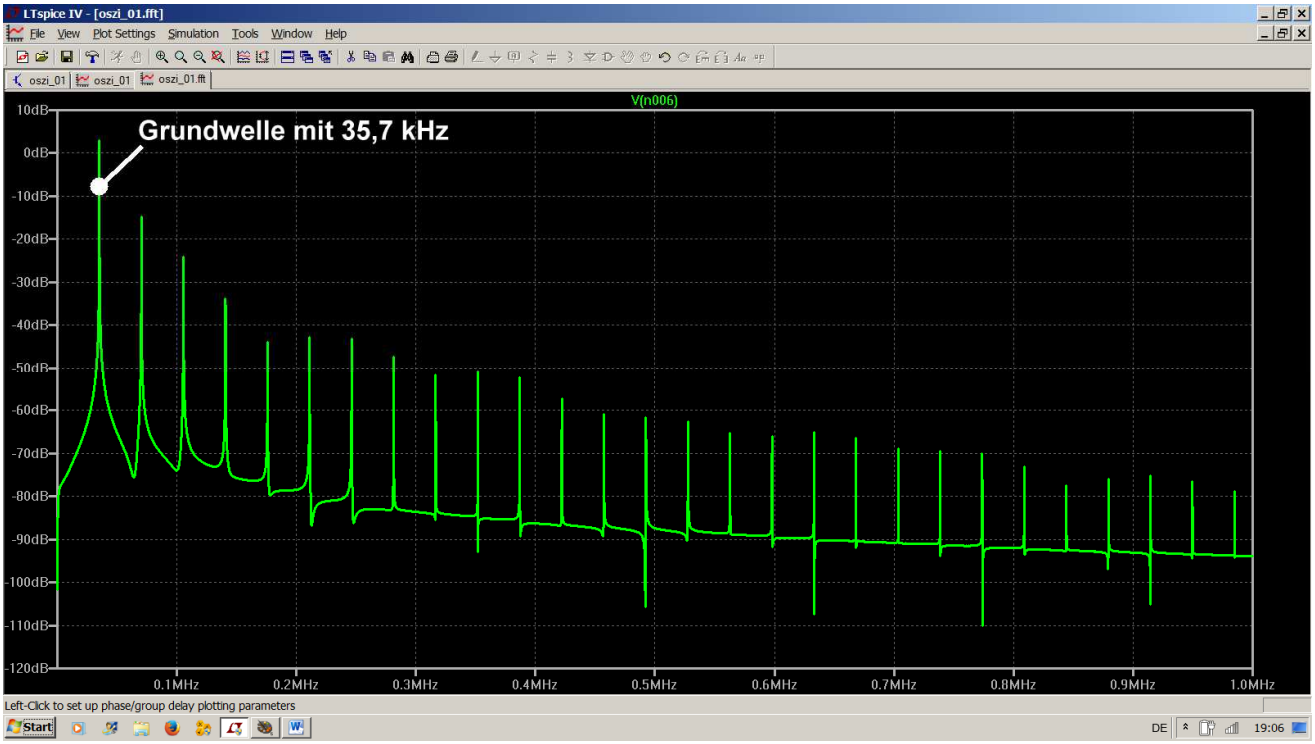
Sehen wir uns doch mal das Frequenzspektrum an.



Dazu klicken wir mit der rechten Maustaste auf das Ergebnisdiagramm und wählen (ganz unten) erst „View“ und dann „FFT“ mit **262 144** Punkten. Außerdem begrenzen wir den betrachteten Zeitbereich auf

1ms……19ms,

um einerseits den **Start- und Hochlauf- und Einschwingbereich**“ für die FFT auszuschließen (….gibt sonst unerwünschte überlagerte Störspektren…). Andererseits hören wir aus demselben Grund schon 1ms vor dem Ende der Darstellung auf, um auch den schroffen **Übergang von „EIN“ zu „AUS“ fernzuhalten**. Das gibt folgendes Ergebnis:



Die Darstellung ändern wir nach einem Klick auf die Zahlenteilung der waagrechten Achse und lassen uns einen **linearen Bereich von 0…1MHz mit einem Tick von 100kHz** zeigen:

Sehr schön ist hier zu sehen, wie die Amplituden der Oberwellen mit steigender Frequenz abnehmen. Auch die „Frequenz der Grundwelle“ lässt sich nun sehr leicht zu ca. 35,7 kHz bestimmen.

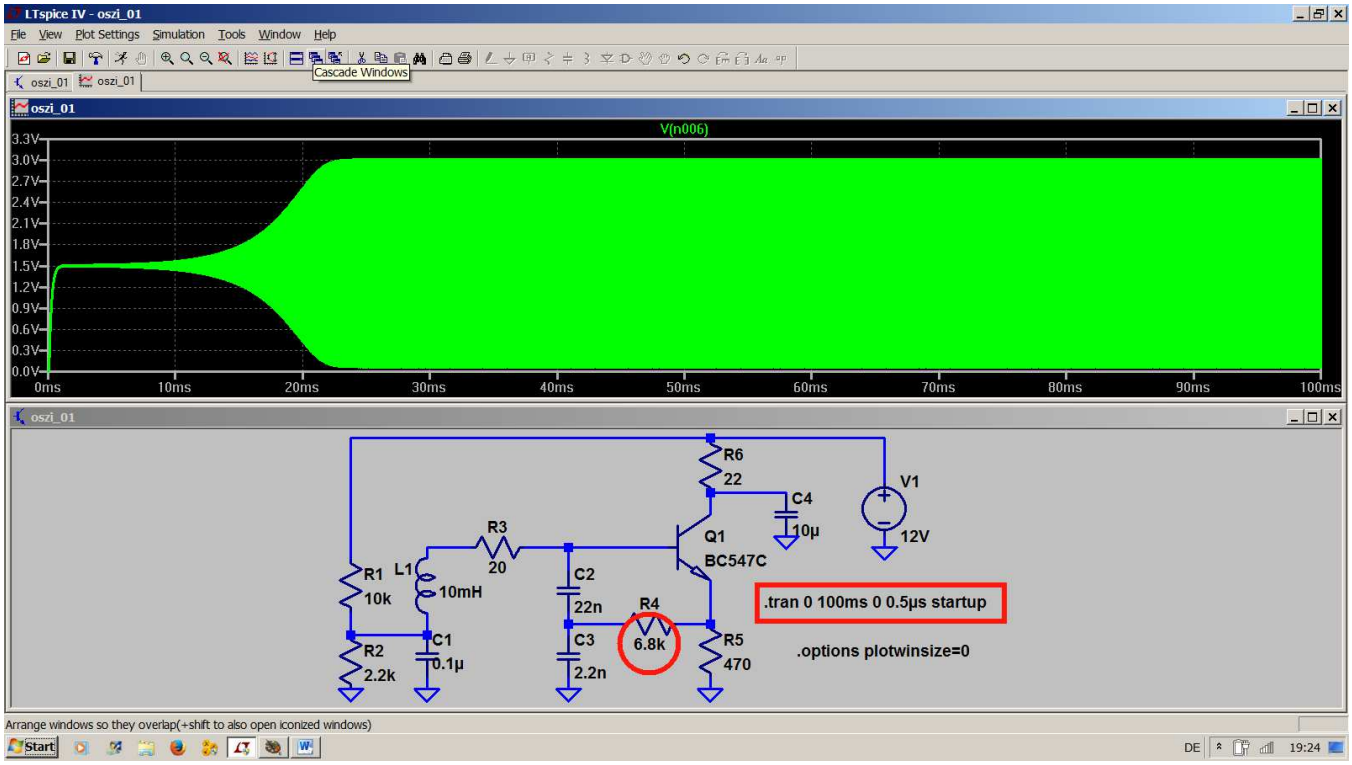
Allerdings ist die doppelte Frequenz um weniger als 20dB gedämpft und das war nach der kräftigen Unsymmetrie des verzerrten Signals zu erwarten (…wissen Sie noch: bei einem streng symmetrisch verzerrten Signal gibt es **nur ungeradzahlige Oberwellen**. **Das Auftreten geradzahliger Oberwellen deutet IMMER auf eine gestörte Symmetrie hin!)**

Übrigens:
Der Widerstand R5 / 680Ω stellt die Mitkopplung dar, denn über ihn fließt ein Teil des Emitter-Wechselstromes in den Schwingkreis, deckt die Verluste und bewirkt so eine Dauerschwingung. Erhöht man seinen Wert, dann wird diese Mitkopplung schwächer. Dadurch wird zwar die Kurvenform des Ausgangssignals verbessert und der Anteil der Oberwellen nimmt ab. Aber irgendwann reicht die Mitkopplung nicht mehr zum Ausgleich der Verluste und die Schaltung kann nicht mehr schwingen.

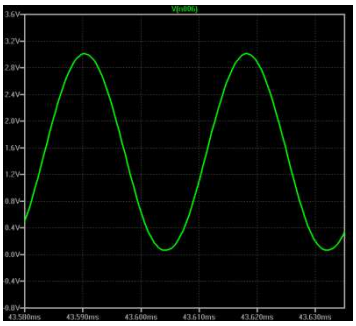
Aufgabe:
Versuchen Sie, diesen Punkt durch wiederholtes Erhöhen von R5 und anschließender Simulation möglichst genau zu finden. Beobachten Sie dabei auch die Verbesserung der Kurvenform und die Veränderung des zugehörigen Oberwellenanteils über das simulierte Frequenzspektrum.

Beachten Sie auch, dass das „Anschwingen“ umso länger dauert, je näher Sie diesem Punkt kommen -- denn dort startet der Oszillator überhaupt nicht mehr.
Dieser Punkt ist bei 6,8k schon fast erreicht. Deshalb wird man in der Praxis höchstens 5,6k als nächsten Normwert einsetzen.
Prüfen Sie diese Information nach.

Musterlösung:



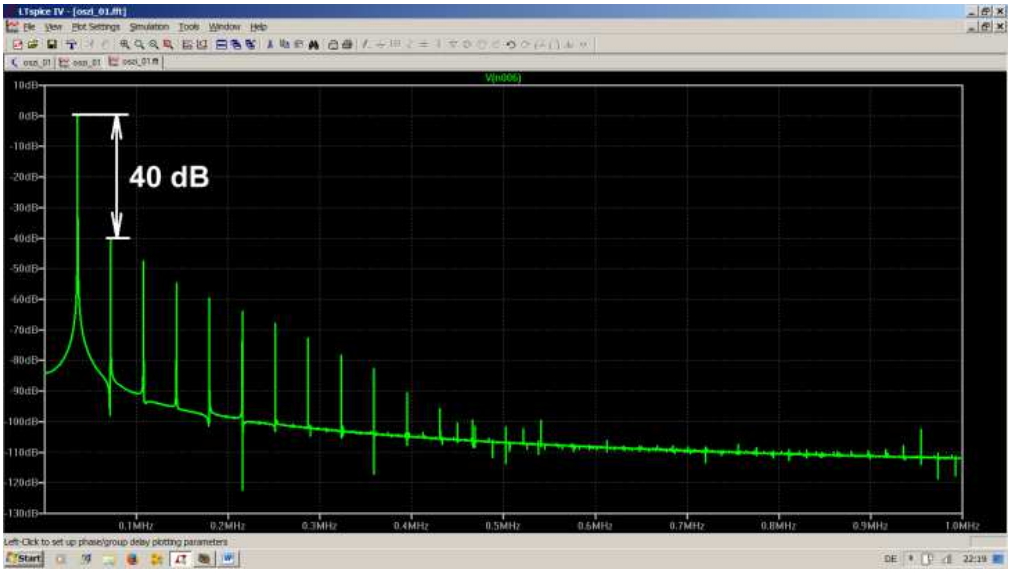
Da sieht man (mit R5 = 6,8k) deutlich das mühsamere Anschwingen….es dauert fast 25 Millisekunden….bei einer Simulationszeit von 100ms und einem Timestep von 500ns….



…aber die Kurvenform ist nun natürlich (fast) perfekt!
Und der Blick auf das Frequenzspektrum sollte dies bestätigen.

Allerding muss man wieder den verwendeten Zeitbereich bei der FFT auf die Zeit nach dem „Anschwingen“ (= ab 25ms), aber bis 1ms vor dem Ende der Darstellung begrenzen.

(= FFT für t = 25ms….99ms mit 262 144 Samples)



Die Verzerrungen im gesamten Frequenz-bereich sind zurückge-gangen und auch die Dämpfung bei der doppelten Frequenz ist auf 40dB gestiegen.

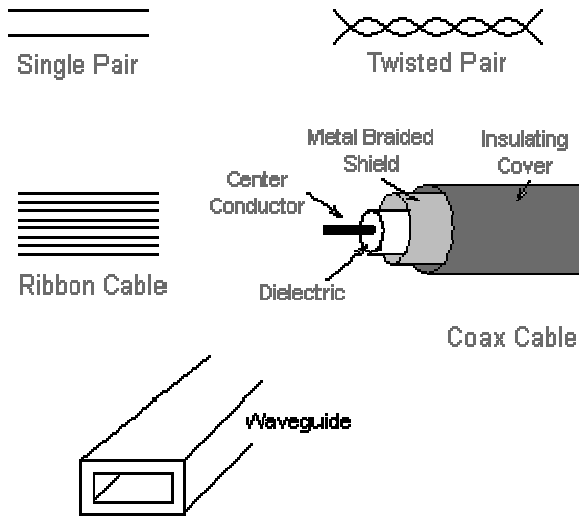
18. Zwölftes Projekt: Echos auf Leitungen

18.1. Leitungen -- nur zwei Drähte?

Wenn man sich einen einfachen Stromkreis mit einer Spannungsquelle und einem Lastwiderstand ansieht, scheint sich diese Frage nicht zu stellen: auf einem Draht kommt der Strom aus der Quelle und fließt zum Verbraucher, auf der anderen Strippe kehrt er zum unteren Anschluss des Generators zurück. Wo ist also das Problem?

Über HF-Leitungen und ihre Eigenschaften müssen wir dann reden, sobald Quelle und Last nicht mehr unmittelbar miteinander verbunden sind. Und bei hohen Frequenzen bedeutet „unmittelbar“ nur noch einige Zentimeter oder sogar schon Millimeter an nötiger Leitungslänge, sobald darauf elektrische Energie transportiert wird!

Sehen wir uns doch mal einige mögliche Ausführungsformen von „Leitungen zwischen Generator und Verbraucher “ an:



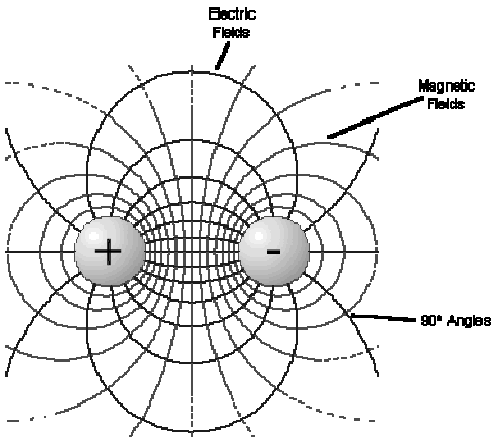
Das „**Single Pair**“ kann sich Jeder vorstellen: eine Hinleitung und eine Rückleitung, allerdings exakt in Kunststoff parallel geführt.

Das „**Twisted Pair**“ ist die Standardform bei modernen LANs, also Computernetzen (...und z. B. beim CAN-Bus)

Das **Koaxkabel** sehen wir als Antennenzuleitung bei jedem Fernsehempfänger

„**Ribbon Cable**“ ist ein vieladriges rundes Kabel oder ein vieladriges Flachbandkabel (z. B. in PCs zur Verbindung der Festplatte oder des CD-Laufwerks mit dem Mainboard)

Richtig geheimnisvoll wird es erst beim „**waveguide**“ (= Hohlleiter), denn dort werden die elektrischen und magnetischen Felder nicht zwischen Drähten, sondern in einem „Hohlraum“, also in Luft, geführt.



Sobald wir nun an eine Leitung (..wir nehmen als Beispiel das „Single Pair!“) eine elektrische Spannung anlegen, gehört dazu bekanntlich ein elektrisches Feld zwischen den beiden Drähten. Wie sich das verteilt und wie es verläuft, kann man im nebenstehenden Bild sehen:
Für uns ist dabei wichtig, dass das zugehörige Bauteil zu diesem Bild (positive bzw. negative Ladung und dazwischen das elektrische Feld) genau einem **Kondensator** entspricht.

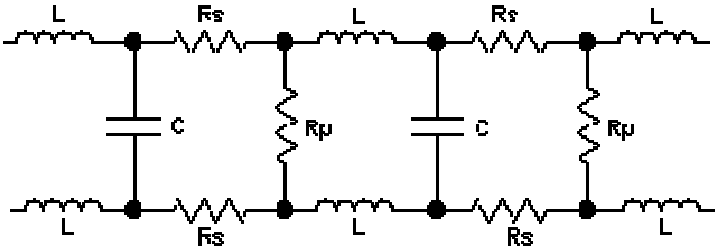
Wir merken uns:
Jedes noch so kurze Leitungsstück besitzt eine bestimmte Eigenkapazität, die linear mit der Leitungslänge zunimmt!

Sobald in den Leitungsdrähten ein Strom fließt, umgibt sich dieser Draht mit einem Magnetfeld und das zugehörige Bauteil ist die Spule (= Induktivität).

Wir merken uns:
Die Leitungsdrähte bewirken die Eigeninduktivität der Leitung, die ebenfalls mit der Länge zunimmt.

Natürlich darf man bei dieser Betrachtung nur sehr kurze Leitungsstücke annehmen, da es sich ja um „stufenlos verteilte“ Bauelemente handelt. Außerdem weisen die Leiterdrähte einen **Ohmschen Widerstand in Reihe** zur Induktivität auf, der wegen des „Skineffekts“ mit der Frequenz zunimmt.

Ebenso steigen die Verluste der Isolation zwischen den Drähten (= Dielektrikum des Kondensators) mit der Frequenz, die wir als **Parallel-Widerstand** zum Kondensator denken können. Das ergibt nebenstehendes Ersatzbild der Leitung, wenn wir sie gedanklich in lauter kurze Stücke aufteilen.



Sobald nun ein elektrisches Signal an die Leitung angelegt wird, sind mehrere Effekte zu beobachten:

a) Das Kabel benimmt sich für ein „eintretendes Signal“ (Fachausdruck: „Hinlaufende Welle = incident wave“) zunächst wie ein ohmscher Widerstand. Er trägt den Namen „Wellenwiderstand“ und lässt sich folgendermaßen berechnen:

$$Z = \sqrt{\frac{Induktivität}{Kapazität}}$$

wobei die Induktivität bzw. Kapazität für ein Leitungsstück mit einer bestimmten Länge (z. B. 10cm oder 1m) gelten. Übliche Werte in der gesamten Kommunikationstechnik ist Z = 50 Ohm, während die Fernseh- und Videotechnik 75 Ohm bevorzugt.

b) Wenn sich das Signal und damit die aufgenommene Energie im Kabel vom Generator wegbewegt, müssen dauernd die nächsten Kapazitäten über die kleinen Spulen auf- oder umgeladen werden. Das dauert so seine Zeit und deshalb ist die Ausbreitungsgeschwindigkeit auf dem Kabel kleiner als die Lichtgeschwindigkeit „c“. Es gilt:

$$v_{Kabel} = \frac{Lichtgeschwindigkeit}{\sqrt{\varepsilon_r}}$$

wobei ε_r die Dielektrizitätskonstante der Kabel-Innenisolation darstellt.

Material	Dielectric Constant (k)	Wave Velocity (relative to C)
Vacuum	1.00000	1.00000 C
Air	1.0006	0.9997 C
Teflon	2.10	0.690 C
Polyethylene	2.27	0.664 C
Polystyrene	2.50	0.632 C
Polyvinyl Chloride (PVC)	3.30	0.550 C
Nylon	4.90	0.452 C

Das bekannte RG58-Koaxialkabel mit Z = 50 Ohm enthält eine Polyäthylen-Innenisolation. Aus der nebenstehenden Tabelle sieht man, dass sich darin die Signale nur noch mit 66% der Lichtgeschwindigkeit ausbreiten...

c) Wie es mit der auf dem Kabel vom Generator in Richtung Verbraucher wandernden Energie weitergeht, sehen wir uns im nächsten Kapitel genauer an.

18.2. Echos auf Leitungen

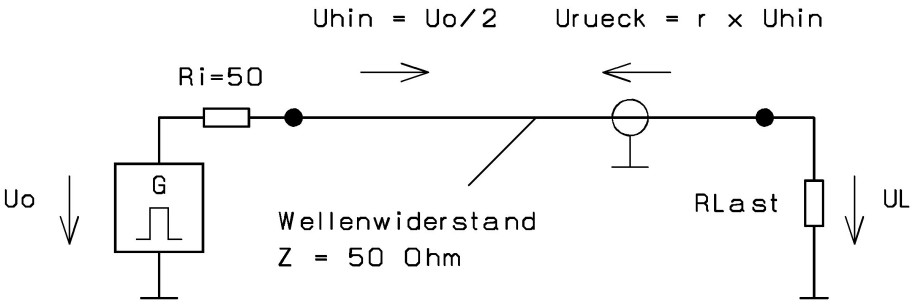
Bei hohen Frequenzen lassen sich Ströme und Spannungen nicht mehr so einfach messen, außerdem klappt auch die Sache mit Leerlauf- oder Kurzschluss-Messung zum Bestimmen des Innenwiderstandes einer Quelle nicht mehr richtig. Die Systembeschreibung und -berechnung geht deshalb dort von Größen aus, die auch **bei höchsten Frequenzen (z. B. bei f = 100GHz) leicht messbar sind und man benützt deshalb schon ab weniger als 10MHz ein völlig anderes Denkmodell:**

Überall im System wird derselbe „**Systemwiderstand**“ verwendet (üblich sind 75Ω bei der Unterhaltungselektronik und Videotechnik, dagegen 50Ω bei den meisten anderen Gebieten). Er gilt für den Innenwiderstand der Speisespannungsquellen, die Ein- und Ausgangswiderstände der verwendeten Baugruppen, den Wellenwiderstand aller Verbindungskabel und für alle Abschlusswiderstände.

Der Kern dieser Sache ist also das Prinzip der perfekten Leistungsanpassung (mit Ri = Ra) im kompletten System!

Durch geeignete Messgeräte (=Richtkoppler) misst man nun die Abweichungen von diesem Idealfall und drückt sie durch „**Reflektionsfaktoren**“ aus. Dazu sollte man jedoch die grundsätzlichen Vorgänge auf Leitungen genau kennen.

Annahme:
Eine Signalquelle schickt einen kurzen Puls in ein Verbindungskabel zu einem beliebigen Lastwiderstand oder irgendeiner Baugruppe.



Dann laufen folgende Vorgänge ab:

a) Bei größerer Kabellänge „**merkt man wegen der endlichen Signal-Ausbreitungs-geschwindigkeit (in Luft sind es in einer Nanosekunde 30cm) im ersten Moment noch nichts vom Verbraucher“**.

b) Folglich weist das Kabel einen **Eingangswiderstand von 50Ω** auf und es bildet mit dem Innenwiderstand der Quelle einen Spannungsteiler. Dadurch herrscht also zunächst „Leistungsanpassung“ (weil Ri = Ra) und die vom Kabeleingang aufgenommene **maximal abgebbare Wirkleistung** (zu der die Spannungsamplitude Uo/2 gehört) macht sich mit der Kabelgeschwindigkeit auf den Weg in Richtung Verbraucher.

c) Kommt diese „hinlaufende Wirkleistung“ (sie wird immer als **„hinlaufende Welle“** bezeichnet) am Verbraucher an, so wird sie nur dann voll absorbiert, wenn auch hier Leistungsanpassung (R_{LAST} = Z = 50Ω) herrscht. Jede Abweichung des Lastwiderstandes vom Systemwiderstand bedeutet, dass nun keine Leistungsanpassung mehr möglich ist. **Folglich wird die „überschüssige Energie in Richtung Signalquelle zurückgeschickt“ (= reflektiert) und wir können plötzlich zusätzlich eine „rücklaufende Welle“ auf der Leitung beobachten!**

Für diesen Sachverhalt wurde der Begriff des **Reflektionsfaktors „r“** eingeführt und wir erhalten damit:

$$r = \frac{(Z_{Last} - Z)}{(Z_{Last} + Z)} = \frac{U_{rück}}{U_{hin}} \qquad \text{bzw.} \qquad U_{rück} = r \bullet U_{hin}$$

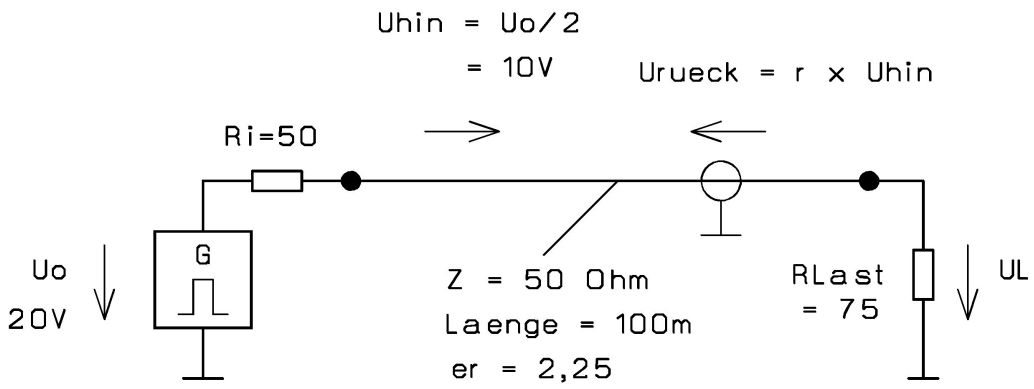
Für die **Spannung am Lastwiderstand** können wir dann schreiben: $U_{Last} = U_{hin} + U_{rück}$

Hinweis:
Für die auf dem Kabel wandernden Wellen muss ja -- da es sich hierbei um elektrische Energie handelt -- , das Ohm'sche Gesetz gelten. Damit kann bei Bedarf der zugehörige Strom berechnet werden:

$$Z = \frac{U_{hin}}{I_{hin}} \qquad \text{bzw. } Z = \frac{U_{rück}}{I_{rück}}$$

Machen wir uns das doch mal alles an einer hübschen kleinen Übungsaufgabe klar:
Ein Pulsgenerator mit dem Innenwiderstand 50Ω liefert im Leerlauf an seinen Klemmen kurze Impulse mit der Folgefrequenz 1kHz, der Pulsbreite 10ns und der Amplitude 20V. Er wird nun über ein 100m langes Kabel (Z = 50Ω, Typ RG58) mit einem Abschlusswiderstand von 75Ω verbunden. Die Dielektrizitätskonstante der Kabel-Innenisolation beträgt ε = 2,25.

Bestimmen Sie die Signalverläufe
a) am Kabeleingang b) in Kabelmitte c) am Kabelende



Lösung:

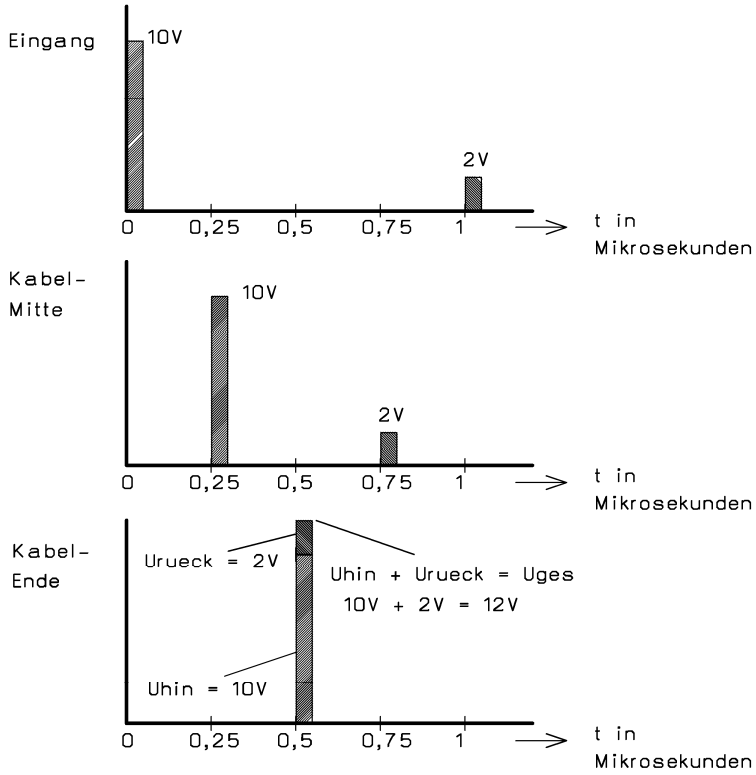
a) Bestimmung des **Reflektionsfaktors:** $r = \frac{R_{LAST} - Z}{R_{LAST} + Z} = \frac{75\Omega - 50\Omega}{75\Omega + 50\Omega} = +0{,}2$

b) Bestimmung der **Kabelgeschwindigkeit:** $v_{Kabel} = \frac{c}{\sqrt{\varepsilon_R}} = \frac{3 \bullet 10^8 m}{\sqrt{2{,}25} s} = 2 \bullet 10^8 \frac{m}{s}$

c) **Signal-Laufzeit** für 100 m Kabellänge: $t_{Delay} = \frac{100m \bullet s}{2 \bullet 10^8 m} = 0{,}5 \bullet 10^{-6} s$

d) **Hinlaufende Welle**: $U_{hin} = \frac{U_0}{2} = \frac{20V}{2} = 10V$

e) **Rücklaufende Welle**: $U_{rück} = r \bullet U_{hin} = 0{,}2 \bullet 10V = 2V$



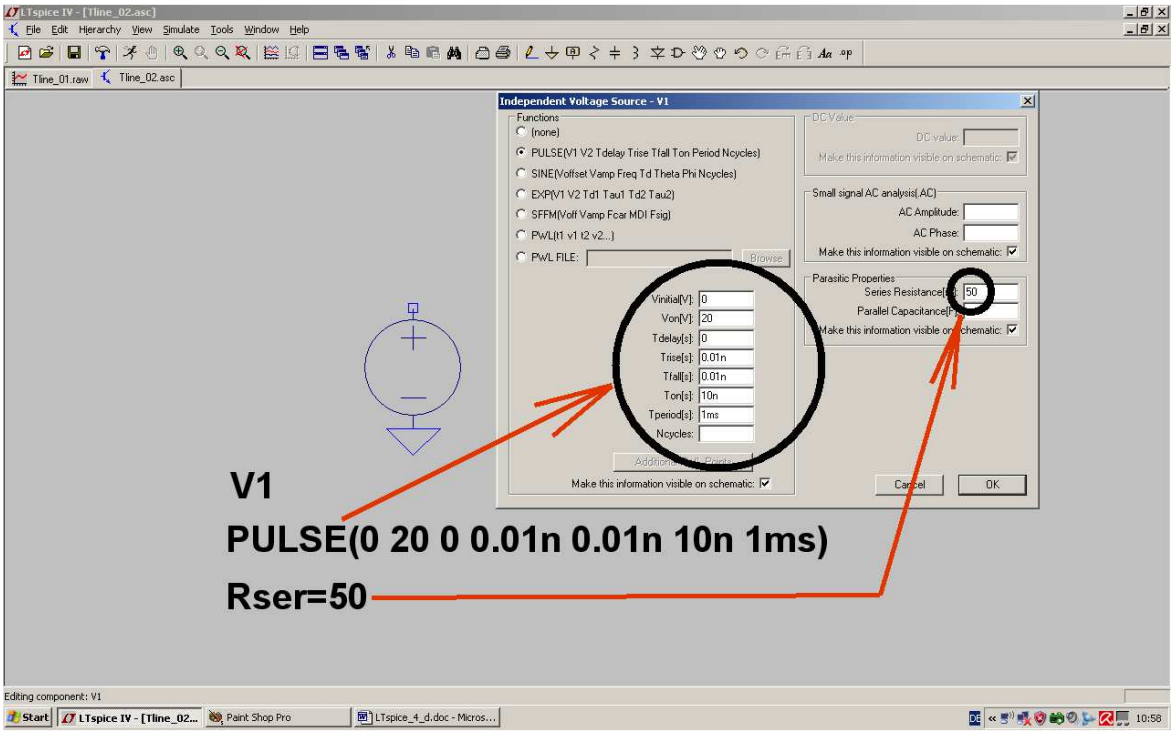
Am **Eingang** beobachtet man zunächst die hinlaufende Welle mit einer Amplitude von 10V (= halbe Urspannung). Nach 2 x Kabellaufzeit taucht das Echo (= rücklaufende Welle) auf.

Nach 0,25 Mikrosekunden hat die hinlaufende Welle die **Kabelmitte** erreicht. Das Echo kommt 0,5 Mikrosekunden später vorbei.

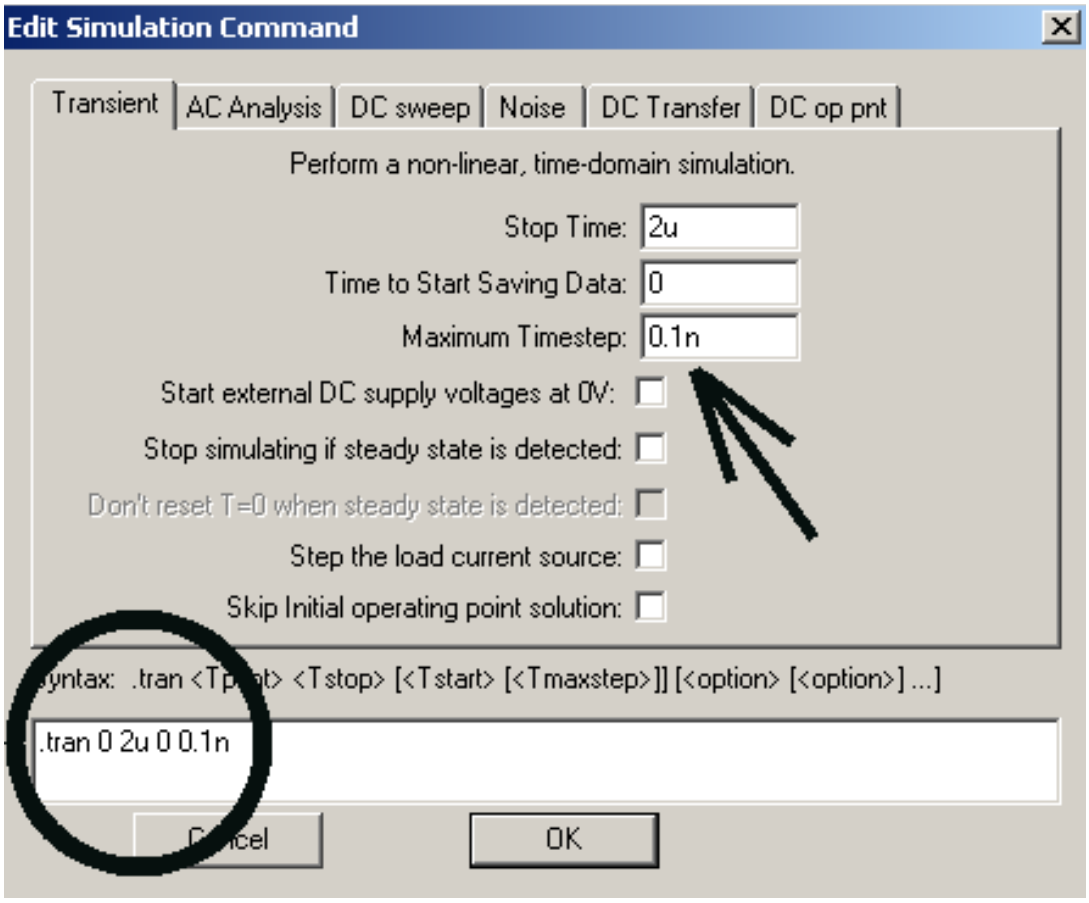
Genau nach 0,5 Mikrosekunden erreicht die hinlaufende Welle das **Kabelende**. Die angelieferte maximale Wirkleistung kann jedoch nicht vollständig an den „falschen" Abschlusswiderstand abgegeben werden. Deshalb entsteht sofort eine „rücklaufende Welle" mit 2V, durch die alle „überschüssige Energie" zurück zur Quelle transportiert wird.
Für einen kurzen Augenblick misst man also dort eine Spannung von 10V +2V = 12V.

18.3. Simulation des vorigen Rechenbeispiels mit LTSpice

Dazu fangen wir mit der **Spannungsquelle** an, die wir uns aus dem Bauteilvorrat als „**voltage**" holen. Nach einem rechten Mausklick auf das platzierte Schaltzeichen programmieren wir ein **Pulssignal mit der Urspannung von 20V, einer Pulsdauer von 10ns und einer Anstiegs- bzw. Abfallzeit von 0,01ns. Die Periodendauer wird zu 1ms gewählt und -- nicht vergessen!! -- ein Innenwiderstand von 50 Ohm eingestellt:**



Bei der Einstellung der Simulationszeit müssen wir **die maximale Timestep-Schrittweite begrenzen, denn die Anstiegs- und Abfallflanke sind mit 10 Picosekunden extrem kurz**. Also sorgen wir dafür, dass der größte mögliche Schritt **100 Pikosekunden** beträgt:



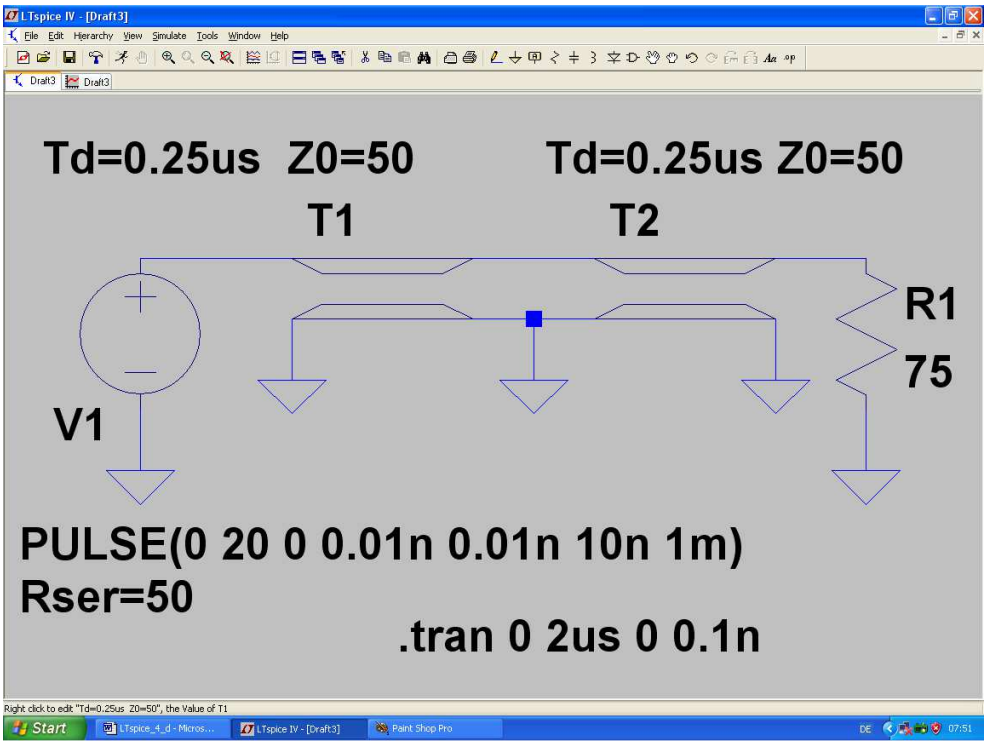
Jetzt geht es an die „**Transmission Line**“. Sie findet sich als „**tline**“ im Bauteilvorrat. Allerdings müssen wir vor dem Einsatz noch etwas rechnen, denn wir können bei ihr zwar die „**Kabellaufzeit“ als „delay time“** programmieren, aber **nicht die Kabellänge**.

Also bestimmen wir mit der mechanischen Kabellänge von 100m und der Kabelgeschwindigkeit von 200 000 km/sec diese Delay Time:

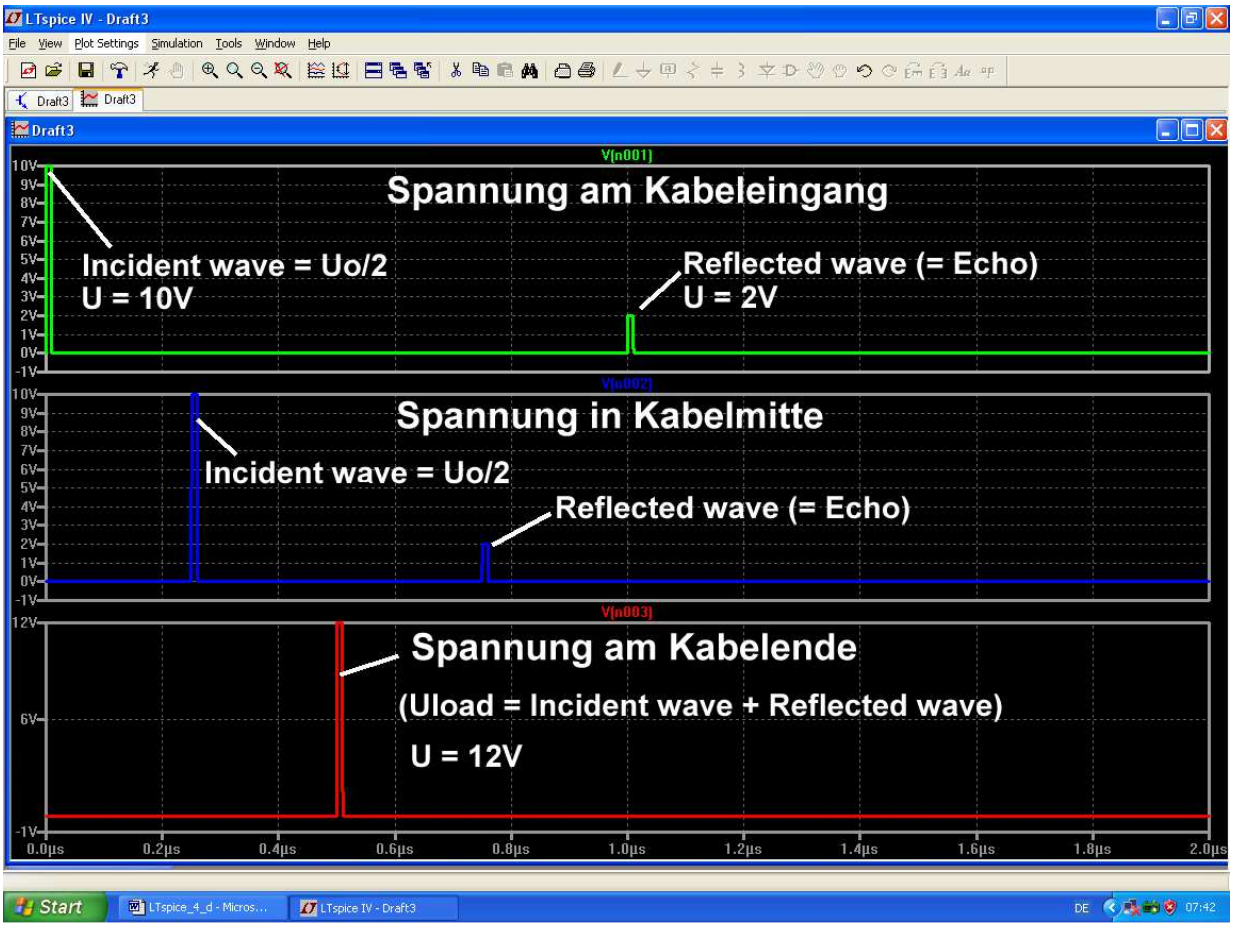
$$t_{delay} = \frac{l}{v_{cable}} = \frac{100m \bullet s}{2 \bullet 10^8 m} = 0{,}5\mu s$$

Diese Delay Time müssen wir auf zwei gleich lange und direkt miteinander verbundene 50Ω-Leitungsstücke aufteilen. Jedes Stück ergibt eine Verzögerungszeit von 0,25 Mikrosekunden und damit stimmt die Sache mit den 0,5 Mikrosekunden Gesamtzeit wieder. Aber nur so kommen wir an die Kabelmitte heran, um dort die Spannung zu messen.

Jetzt fehlt nur noch ein Abschlusswiderstand von 75Ω. Und mit einer Simulationszeit von 2 Mikrosekunden sollte unser Bildschirm so aussehen (Bitte nochmals alle Einstellungen kontrollieren…):

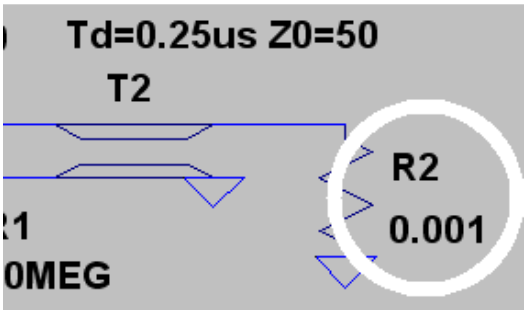


Nach der Simulation werden die drei messbaren Signale der Schaltung in getrennten Diagrammen dargestellt. Dann sieht das so aus und beim Vergleich mit Kapitel 13.2 kann man nur sagen: „Na also…“.



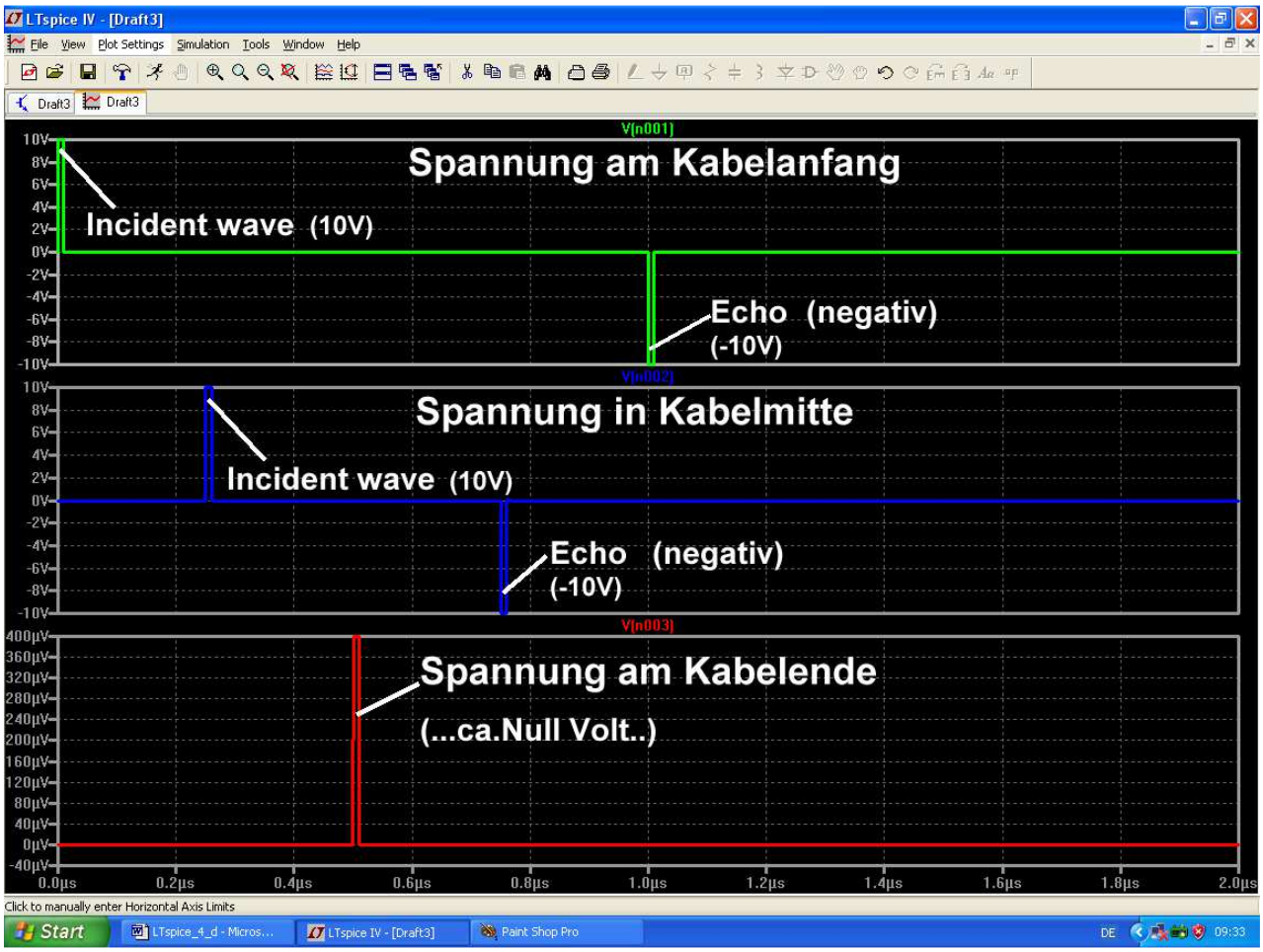
18.4. Leerlauf oder Kurzschluss als Last am Kabelende

Da nehmen wir unsere bisherige Simulationsschaltung her und ändern nur den Wert des Abschlusswiderstandes.



a) **Kurzschluss:**
Ein Wert von 0,001Ω (also 1 Milli-Ohm) ist hier realistisch und sinnvoll.

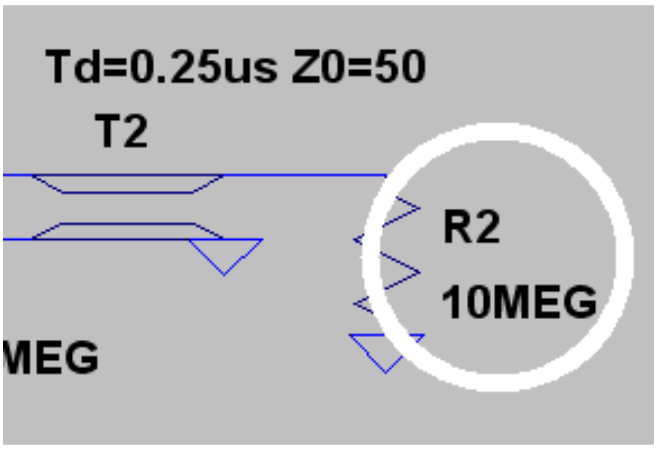
Damit erhalten wir folgende Signale:



Da am Kurzschluss beim Leitungsende die Spannung gleich Null sein muss, wird keine Leistung abgenommen. Folglich kehrt die ankommende Welle (= „incident wave“) ihre Polarität um und läuft als „reflected wave“ zum Generator zurück.

Am Restwiderstand des Kurzschlusses (= 1 Milli-Ohm) beobachten wir einen kurzen Puls mit einer Amplitude von 400 Mikrovolt.

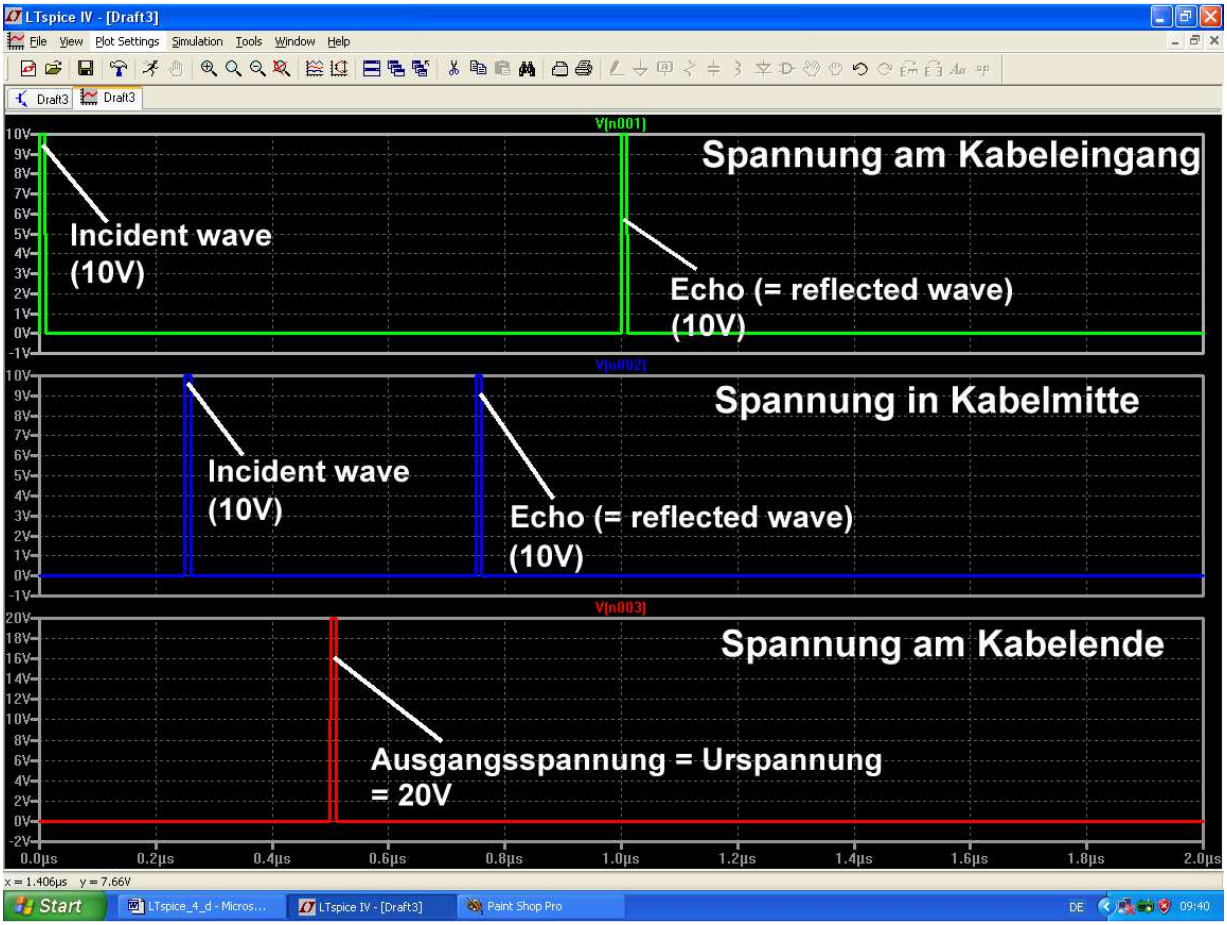
b) Leerlauf am Kabelende



Jetzt reicht ein Widerstand mit 10MΩ für den fast idealen Leerlauf aus.

Achtung:
Bitte den Widerstand R2 NICHT entfernen, um den Leerlauf noch perfekter zu machen. Das Programm antwortet sonst sofort mit „node is floating..“ und bricht die Simulation ab!

Und das sehen wir:



Auch jetzt wird am Kabelende wieder keine Leistung abgenommen (denn es fließt praktisch kein Strom in den Lastwiderstand). Also wird alles wieder zum Generator zurückgeschickt.
Das entspricht den Verhältnissen bei einer Spannungsquelle, an die keine Last angeschlossen ist. Deshalb misst man am Kabelende kurzzeitig die Leerlaufspannung (Urspannung) des Generators.

18.5. Verwendung von Kabel mit Verlusten (Beispiel: RG58 / 50Ω)

18.5.1. Wie simuliere ich RG58-Kabel?

Dazu ist ein anderes Bauteil erforderlich, welches als

ltline

(= lossy transmission line) bereits im Bauteilvorrat von LTSpice enthalten ist.

Allerdings gibt es ein kleines Problem:
Die zugehörigen Leitungseigenschaften müssen in einem selbst geschriebenen Modell-File untergebracht werden. Dieses Modell-File (..wir taufen es „RG58.mod“) wird anhand der aus dem Internet beschafften Kabeldaten zusammengebastelt und anschließend mit dem ltline-Symbol verknüpft.

Im LTSpice-Handbuch findet sich zu diesem Thema eine kurze Information und das Ergebnis muss z. B. so aussehen:

```
.model   RG58   LTRA(len=100   R=1.5   L=250n   C=100p)
```

Erklärung:

„.model“	ist die erforderliche SPICE-Syntax für ein Modell
„RG58“	heißt unser Kabel und das ist nun der Name, unter dem es in der Schaltung verwaltet wird
„LTRA“	heißt „Lossy Transmission Line“ und die folgende Klammer enthält die Kabeldaten
„len=100“	bedeutet, dass wir mit einer Kabellänge von 100 „Längeneinheiten“ arbeiten wollen. Bei unserem Modell beträgt die gewählte Längeneinheit einen Meter, und damit entspricht das einer Kabellänge von 100m
„R=1.5“	gibt an, dass wir pro Längeneinheit mit einem Verlustwiderstand (Serienwiderstand) von 1,5Ω rechnen müssen
„L=250n“	ergibt eine Induktivität von 250nH pro Längeneinheit (hier: für 1m)
„C=100pF“	ergibt eine Kapazität von 100pF pro Längeneinheit (also für 1m)

Diese Zeile wird mit einem Texteditor geschrieben und als „**RG58.mod**“ im Verzeichnis „**Programme / LTC / LTspiceIV / lib / sub**“ abgespeichert.

Übrigens: die Daten „100pF pro 1m“ und „250nH pro 1m“ findet man für RG58 im Internet. Die Verluste mit „1,5Ω pro 1m“ sind eine erste grobe Schätzung für den Einstieg. Wie man damit zum richtigen Wert kommt, kriegen wir gleich….

18.5.2. Simulation der Kabeldämpfung bei 100MHz

Dazu holen wir uns erst mal wieder eine passende Information aus dem Internet.

Frequenz *frequency* MHz	Dämpfung typ. Meßwerte *attenuation typ. measured values* dB/100m	max. Dämpfung *max attenuation* dB/100m	min. Rückfluß-dämpfung *return loss* dB	$P_{40°C,max}$ W	$U_{40°C,max}$ V
50	10,1	11		210	105
100	14,6	16		145	90
200	21,6	24		95	75
300	27,5	30		75	65
450	36,0	40		55	55
500	38,1	42	20	55	55
800	52,8	58		40	45
900	57,5	64		35	45
1000	62,3	70		30	40
1800	102,4	113		20	30
2000	108,5	120		20	30

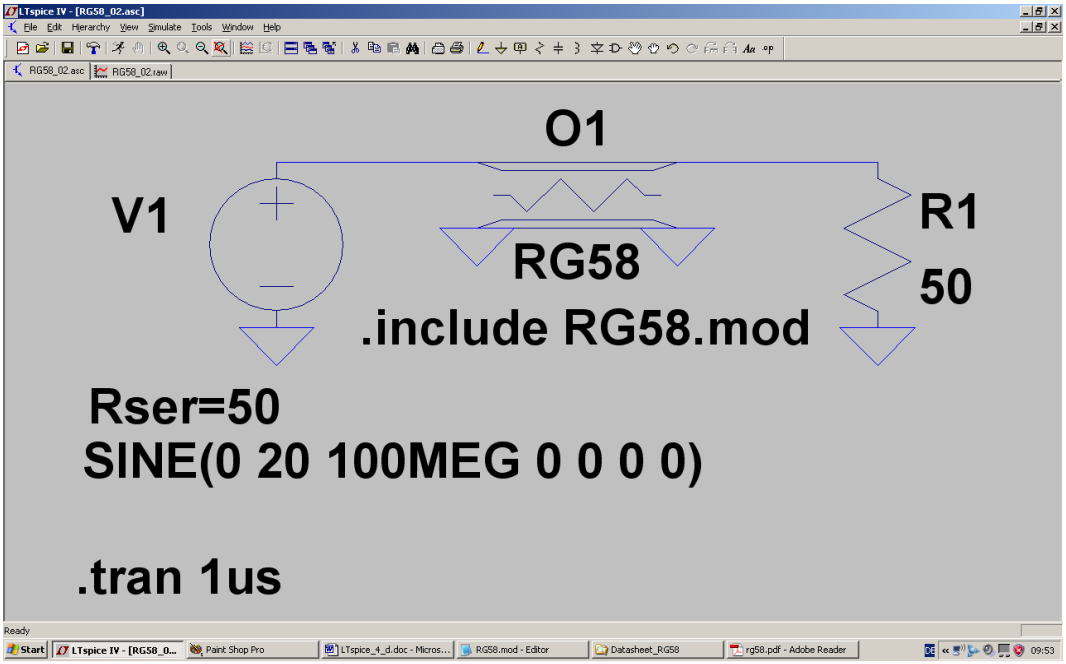
Wie man sieht, ist die Kabeldämpfung sehr stark frequenzabhängig. Man müsste deshalb für jede Arbeitsfrequenz eine eigene Simulation mit dem passenden Tabellenwert durchführen. Machen wir das mal für eine Arbeitsfrequenz von 100MHz:

Schritt 1:
Laut obiger Tabelle beträgt die typische Kabeldämpfung bei f = 100MHz etwa 14,6dB pro 100m. Damit ist die Ausgangsspannung am Kabelende um den Faktor

$$10^{\frac{-14{,}6dB}{20dB}} = 0{,}186$$

kleiner.

Schritt 2:
Wir versorgen einen 50 Ω-Abschlußwiderstand über ein 100 m langes RG58-Kabel und speisen eine sinusförmige Spannung ins Kabel ein. Die Spannungsquelle hat einen Innenwiderstand von 50 Ω und eine Urspannung von 20 V. Damit -- und mit der verlustbehafteten Transmission Line „ltline“ aus dem Bauteilvorrat erstellen wir folgende Schaltung:



Bitte beachten:

a) Auf die untere Bezeichnung „**LTRA**“ des abgesetzten Leitungs-Schaltzeichens wird „mit rechts“ geklickt und dann die Modellbezeichnung „**RG58**“ eingetragen.

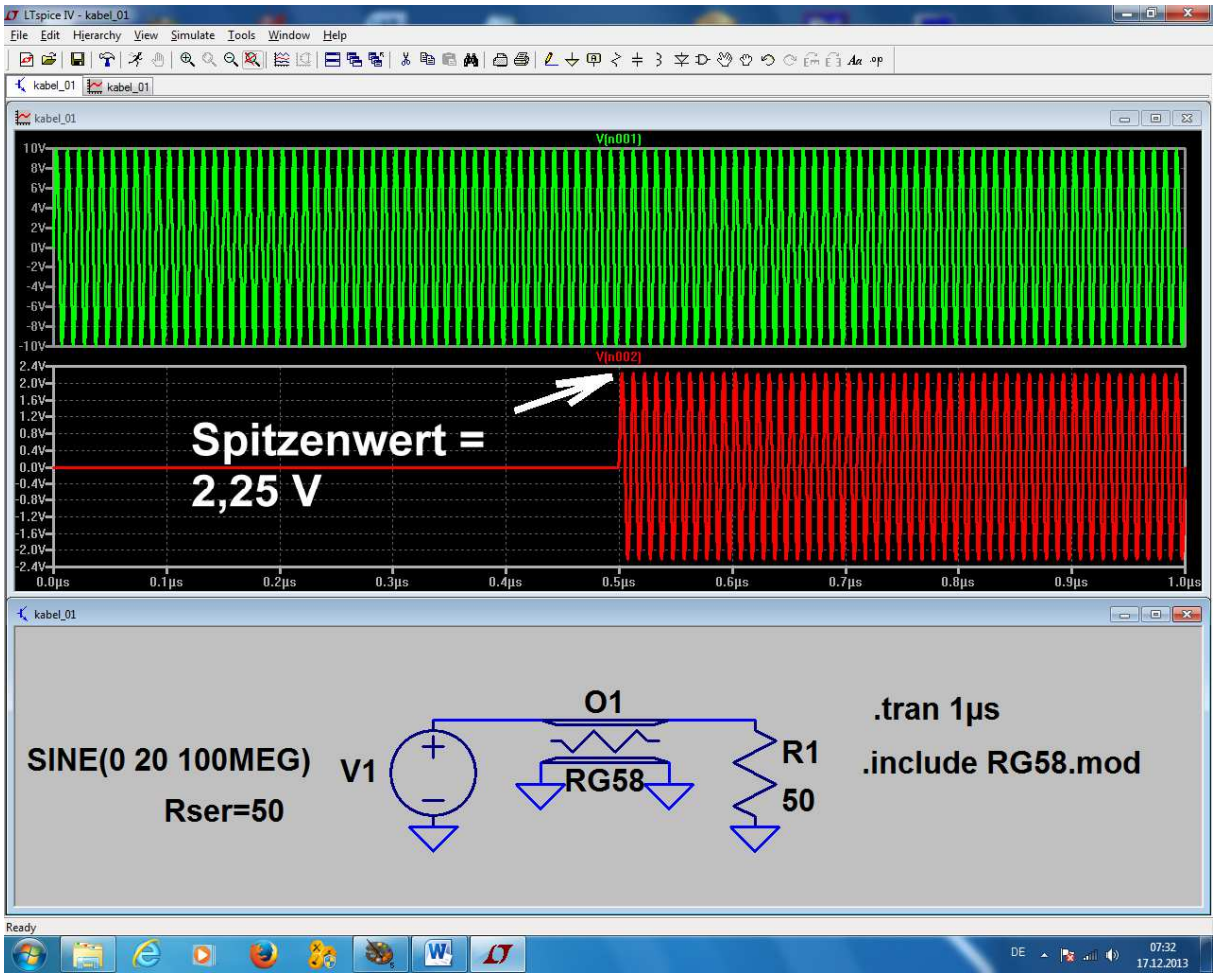
b) Über „**Edit“ und „Spice Directive**“ wird dafür gesorgt, dass die Anweisung

.include RG58.mod

auf dem Bildschirm erscheint.

c) Wir programmieren eine **Simulationszeit von 0….1Mikrosekunde.**

Schritt 3:



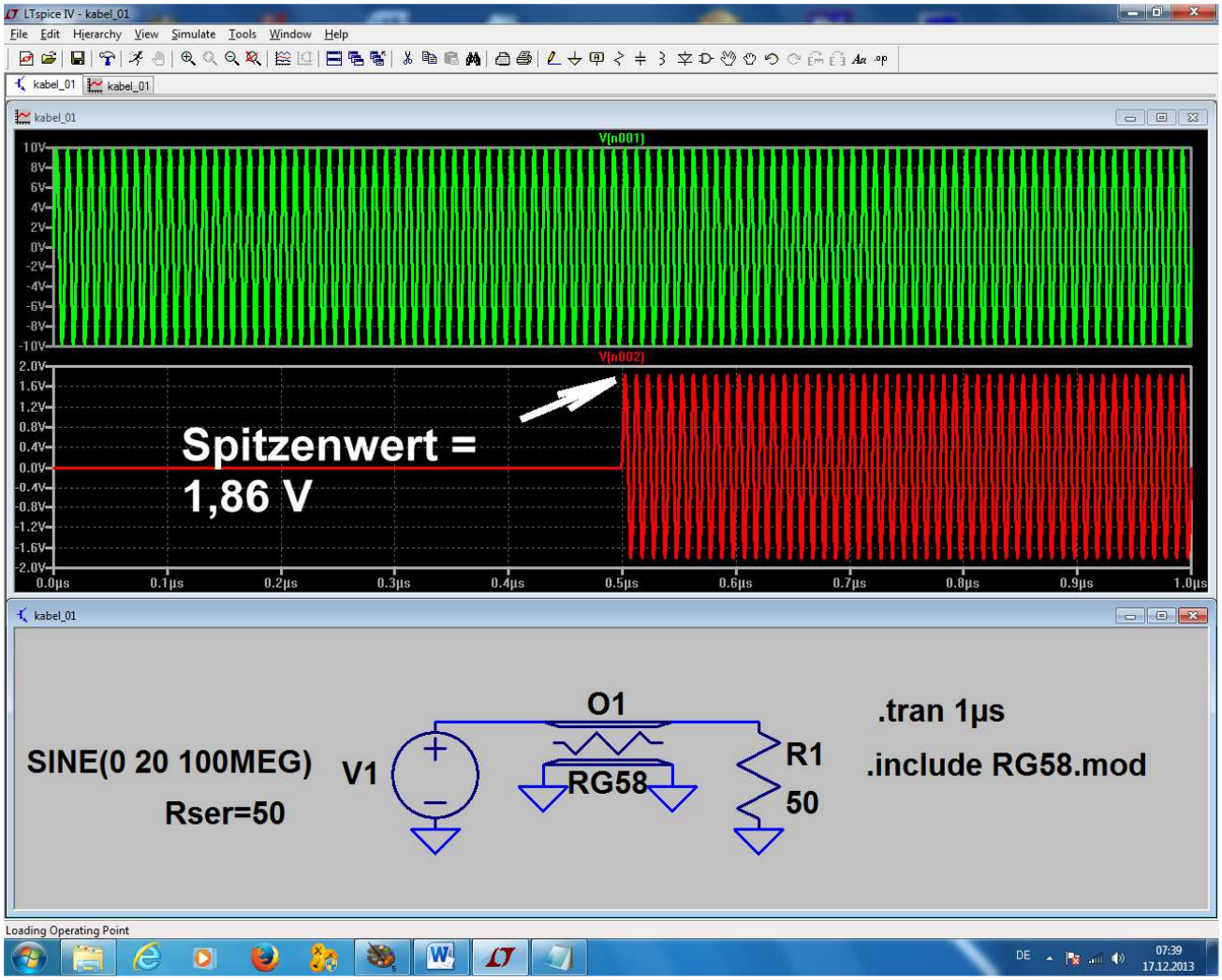
Nach der Simulation werden die Eingangs- und die Ausgangsspannung in zwei getrennten Diagrammen dargestellt:

Man kann nun sehr schön die Signal-Laufzeit von 0,5 Mikrosekunden erkennen, die zu einer Kabellänge von 100m gehört. Außerdem sieht man, dass der Spitzenwert des Ausgangssignals mit ca. 2,25V noch zu hoch für die angegebene Dämpfung von 14,6dB pro 100m ist (es sollten 1,86V sein…).

Schritt 4:
Deshalb öffnen wir mit dem Texteditor unser RG58 – Modellfile und erhöhen etwas den dort angegebenen Wert für „R“. Dann wird damit simuliert und kontrolliert. Einige Wiederholungen dieser Prozedur mit unterschiedlichen Werten bringen uns schließlich zum

Optimum mit R = 1,7.

Damit erhalten wir das gewünschte Ergebnis (Siehe nächste Seite).

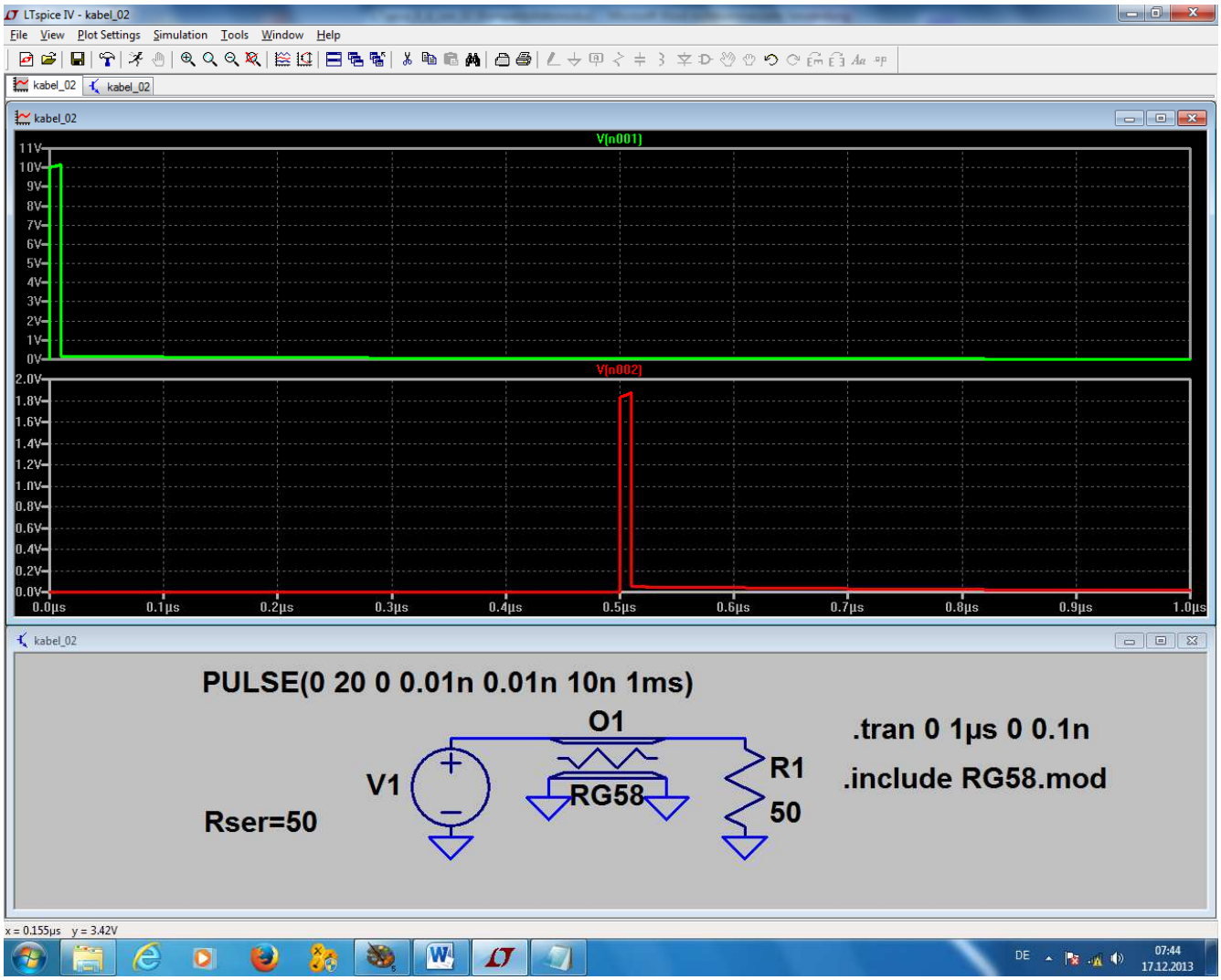


So passt nun alles…

18.5.3. Speisung der RG58-Leitung mit einer Pulsspannung

Legen wir an unsere Schaltung mit der korrekten Dämpfung doch mal die in den vorigen Beispielen eingesetzte Pulsspannung mit den bekannten Daten an:

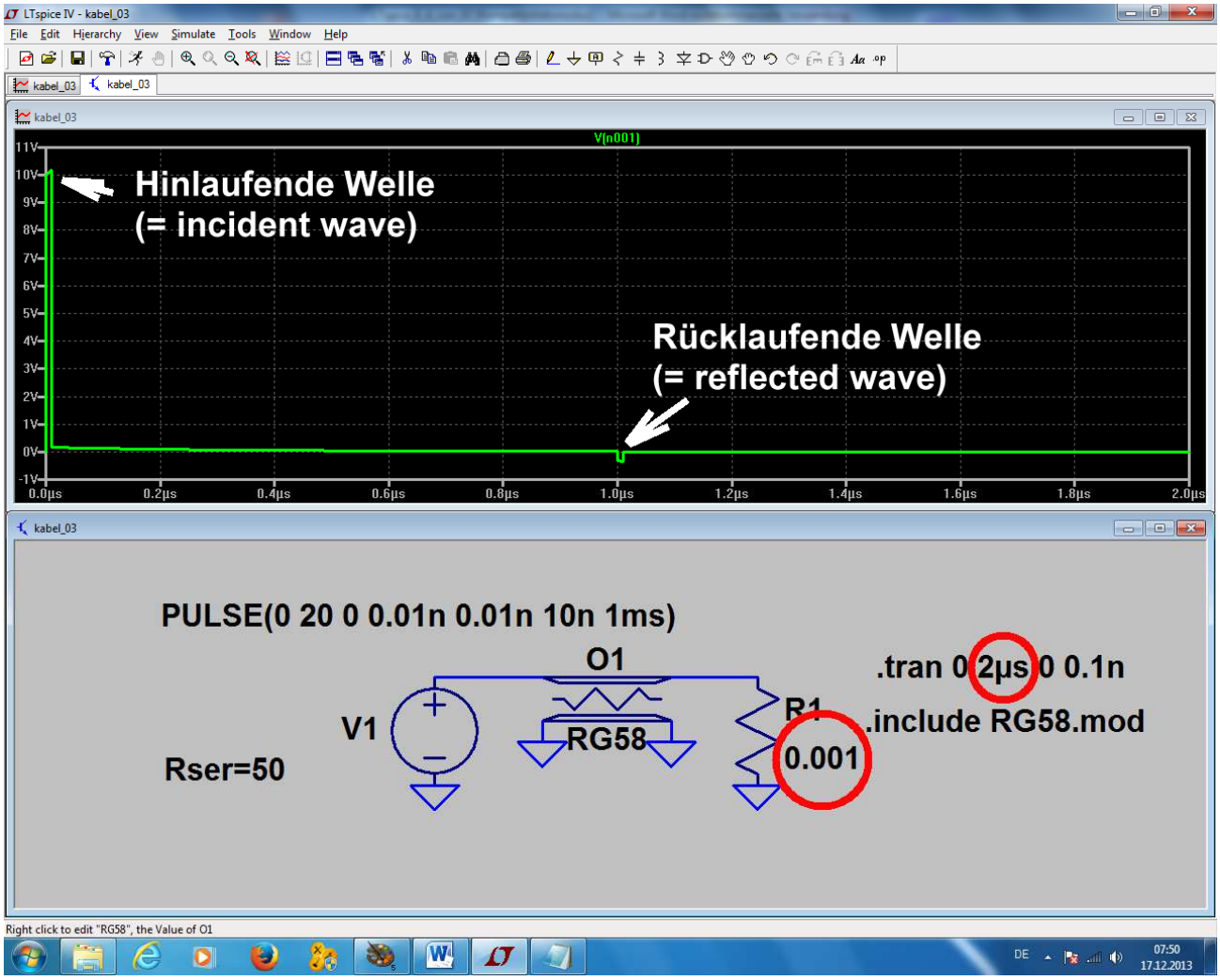
Urspannungs-Maximalwert	**= 20V**
Urspannungs-Minimalwert	**= 0V**
Pulsdauer	**= 10ns**
Anstiegszeit	**= 0,01ns**
Abfallzeit	**= 0,01ns**
Periodendauer	**= 1ms**



Auch hier weist die Ausgangsspannung den erwartetem Spitzenwert von 1,86V auf -- kein Wunder, denn die Dämpfung wird sich nicht ändern, da „R = 1,7 Ω / m“ konstant bleibt. So werden alle im Pulssignal enthaltenen Oberwellen um denselben Faktor gedämpft und damit wird zwar die Pulsamplitude kleiner, aber die Kurvenform bleibt erhalten.

18.5.4. Ein Kurzschluss am Ende der RG58-Leitung

Dafür wählen wir wieder einen **Abschlusswiderstand von 1 Milli-Ohm**, müssen aber auch die **Simulationszeit auf 2 Mikrosekunden** erhöhen. Sonst sieht man das Echo nicht, das ja nach seiner Entstehung nochmals die gesamte Kabellänge durchlaufen muss. Natürlich wird es dadurch zusätzlich gedämpft….



Bei der Darstellung des Ergebnisses brauchen wir nur die Spannung am Kabeleingang zu zeigen, denn an einem Kurzschluss gibt es nicht viel zu sehen…

19. Dreizehntes Projekt: S-Parameter

19. 1.Jetzt nochmals Echos, aber mit System

Der Energietransport auf der Leitung geschieht natürlich immer durch Leistungen. Um aber bei diesen Leistungen wieder mit **Spannungen** rechnen zu können, zieht man einfach die **Quadratwurzel aus der Leistungsformel** und tauft das Ergebnis **„Welle“**.

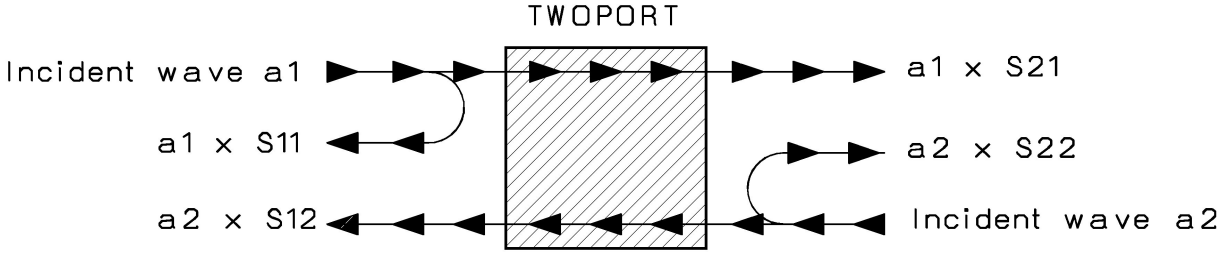
Das ergibt die „**hinlaufende Welle a**“ zu $$\mathbf{a} = \sqrt{\mathbf{P_{hin}}} = \sqrt{\frac{\mathbf{U_{hin}}^2}{\mathbf{Z}}} = \frac{\mathbf{U_{hin}}}{\sqrt{\mathbf{Z}}} = \frac{\left(\frac{\mathbf{U_0}}{\mathbf{2}}\right)}{\sqrt{\mathbf{Z}}}$$

Entsprechend gilt für die „**rücklaufende Welle b**“: $$b = \sqrt{P_{rück}} = \sqrt{\frac{U_{rück}^2}{Z}} = \frac{U_{rück}}{\sqrt{Z}}$$

Außerdem spricht man bei den verschiedenen Anschlüssen eines Bausteins nun von „**PORTS“ (oder „Toren**“). Ein einfacher Baustein mit Eingang und Ausgang heißt dann „**TWO PORT**“ oder „Zweitor“, ein einfaches Bauteil (z. B. eine Diode oder ein Widerstand oder eine Antenne) wäre ein „**ONE PORT** DEVICE“ oder ein „EINTOR“. Die verwendeten TWOPORT-Baugruppen (wie Filter, Verstärker, Abschwächer, Weichen....) weisen aber alle eine Verstärkung oder Dämpfung auf. Das bedeutet:

Die hinlaufende Welle eines Ports wird bei einem „TWOPORT“ stets auch ein Signal am zweiten Port erzeugen. Dieser Vorgang heißt „Transmission“ und wird durch einen eigenen Parameter beschrieben.

Wir wollen nun die eben besprochenen Überlegungen auf ein Zweitor, nämlich einen Verstärker anwenden und das Durcheinander der verschiedenen laufenden Wellen in einer „FLOW CHART“ darstellen:



Erläuterungen zur Eingangsseite:
Die Ansteuerung des Eingangs durch die Signalquelle drücken wir durch die **hinlaufende Welle a_1** aus (= „incident wave“). In der Praxis werden wir dann als Folge ein Echo als **rücklaufende Welle b_1** (= reflected wave) auf der Eingangsleitung beobachten können. Sie lässt sich durch einen Richtkoppler von der hinlaufenden Welle trennen und besteht aus **zwei Anteilen**, nämlich

1) aus einem **reflektierten Anteil von a_1**, der von den Abweichungen des Eingangswiderstandes gegenüber Z = 50Ω herrührt und

2) einem zweiten Anteil, der durch **Rückwirkungen des Ausgangs** (bei dem sich ja auch irgendwelche Signale a_2 herumtreiben) **auf den Eingang** erzeugt wird.

Damit lässt sich das Echo so ausdrücken: $$b_1 = a_1 \bullet S_{11} + a_2 \bullet S_{12}$$

Bedeutung und Messung der verschiedenen Koeffizienten:
Wird der Ausgang korrekt mit R_{Last} = Z abgeschlossen, dann müssen wir dort nicht mehr mit Echos rechnen, die zum Verstärkerausgang zurücklaufen. Also wird a_2 = Null und wir können plötzlich unseren Faktor S_{11} bestimmen:

$$S11 = \frac{b_1}{a_1} \qquad \text{für } a_2 = \text{Null}$$

Aber das kennen wir doch, denn damit ist **S11 nichts anderes als unseres Eingangs - Reflektionsfaktor r aus dem vorigen Rechenbeispiel -- bei diesem Zweitor natürlich für korrekten Abschluss am Ausgang (= input reflection)**. Leider ist er in der Praxis immer komplex, deshalb wird in den S-Parameter-Dateien der Halbleiterhersteller immer „**Betrag" (= MAGNITUDE = MAG) und „Phase" (= ANGLE = ANG) bei verschiedenen Frequenzen angegeben!**

Die Größe **S12** ist dann der „**Rückwärts-Übertragungsfaktor**" (= reverse transmission) vom Ausgang zurück zum Eingang. Er liefert uns Informationen über die **Rückwirkungen** im Zweitor, wenn der Eingang nicht angesteuert und zusätzlich korrekt mit Z = 50Ω abgeschlossen wird. Auf den Ausgang wird in diesem Fall mit einem Signalgenerator die Leistung a_2 „draufgeblasen". Dann gilt:

$$S_{12} = \frac{b_1}{a_2} \qquad \text{für } a_1 = \text{Null}$$

Und nun nehmen wir uns die Ausgangsseite vor:
Wir denken uns auf der rechten Seite einen Signalgenerator, der den Ausgang des Zweitors mit der Wellengröße a2 quält. Mit einem Richtkoppler können wir dann in der Ausgangsleitung folgendes „Echo b2" messen (das wieder aus zwei Teilen besteht):

$$b_2 = a_1 \bullet S_{21} + a_2 \bullet S_{22}$$

Wird der Eingang des Zweitors gerade nicht angesteuert und korrekt mit 50 Ohm abgeschlossen, dann ist a_1 automatisch Null. Also kann ein aus dem „Ausgang des Zweitors herauslaufendes Signal" nur durch eine Reflektion am nicht perfekten Innenwiderstand entstanden sein. **Damit ist S_{22} nichts anderes als der Ausgangs - Reflektionsfaktor (**output reflection**) des Zweitores** (natürlich bei korrektem Abschluss des Einganges mit Z = 50Ω.....)!

$$S_{22} = \frac{b_2}{a_2} \qquad \text{für } a_1 = \text{Null.}$$

Nun bleibt nur noch der linke Summand übrig, aber der macht uns keine Probleme:
Wird nur der Eingang von einer Quelle mit 50Ω Innenwiderstand angesteuert und der Ausgang korrekt mit Z = 50Ω abgeschlossen, dann haben wir die klassische Verstärkerschaltung vor uns. Dann ist nämlich a_2 = Null.

S_{21} ist folglich nichts anderes als die „Spannungsverstärkung des Zweitores bei korrekter Anpassung aus Ausgang" (= forward transmission)!

$$S_{21} = \frac{b_2}{a_1} \qquad \text{für } a_2 = \text{Null.}$$

Behutsamer Hinweis:
Wir arbeiten in Wirklichkeit ja stets mit Leistungen, und eine Leistung ändert sich bekanntlich mit dem Quadrat der Spannung. Also bekommen wir die **Leistungsverstärkung** eines solchen Zweitores im Idealfall als **$(S_{21})^2$** und **nicht als S_{21}**.....also Vorsicht!

Behutsamer Hinweis:
Wir arbeiten in Wirklichkeit ja stets mit Leistungen, und eine Leistung ändert sich bekanntlich mit dem Quadrat der Spannung. Also bekommen wir die **Leistungsverstärkung** eines solchen Zweitores im Idealfall als **$(S_{21})^2$** und **nicht als S_{21}**.....also Vorsicht!

Hinweise für die Praxis:

Aus den Eingangs- bzw. Ausgangs - Reflektionsfaktoren lassen sich natürlich bei Bedarf die zugehörigen Widerstandswerte bestimmen. Bitte aber beachten, dass diese im Normalfall komplex sind und wir deshalb einen Wirk- und einen Blindanteil erhalten:

z. B. Eingangswiderstand $$Z_{11} = Z \bullet \frac{(1+S_{11})}{(1-S_{11})}$$

oder der Innenwiderstand $$Z_{22} = Z \bullet \frac{(1+S_{22})}{(1-S_{22})}$$

Genauso lassen sich aus den gemessenen komplexen Widerstandswerten die Reflektionsfaktoren bestimmen, aber da greift man einfach zum Smith-Diagramm oder -- noch besser! -- zu modernen HF-CAD-Programmen (APLAC, PUFF; ANSOFT-Designer SV...) oder passenden Tools aus dem Internet. Das geht viel schneller und besser (sobald man sich damit auskennt...).
Zum Abschluss dieses Kapitels noch als Beispiel ein S-Parameter-File im „S2p-Format“, heruntergeladen aus dem Internet für den Mikrowellen-Verstärker-IC vom Typ INA-03184“ der Firma Agilent:

```
!        INA-03184        S PARAMETERS
!        Id = 10 mA        LAST UPDATED 07-22-92
# ghz S ma r 50
0.0      .32      180      19.2      0       .014      0       .55       0
0.05     .32      179      19.14    -3       .014      3       .55       0
0.10     .32      176      19.05    -7       .014      4       .57      -3
0.20     .32      172      19.05    -14      .014      6       .55      -5
0.40     .32      165      18.78    -29      .014     10       .53     -11
0.60     .32      158      18.71    -43      .015     11       .51     -14
0.80     .32      151      18.53    -57      .015     13       .51     -17
1.00     .32      144      18.18    -72      .016     21       .50     -20
1.20     .30      135      18.27    -86      .016     25       .50     -23
1.40     .31      126      18.10    -102     .017     30       .49     -29
1.60     .30      117      17.92    -117     .018     38       .48     -34
1.80     .26      102      17.49    -135     .019     44       .45     -41
2.00     .22       92      16.62    -153     .020     49       .40     -50
2.50     .09       91      12.88     168     .021     57       .26     -48
3.00     .14      160       8.79     134     .023     65       .22     -33
3.50     .24      151       5.92     108     .025     69       .26     -33
4.00     .29      139       4.18     87      .029     81       .28     -43
```

Analysieren wir nun die einzelnen Zeilen:

a) Alles, was mit einem **Ausrufezeichen** beginnt, ist ein **Kommentar** und dient nur zur Information

b) Die Zeile **# ghz S ma r 50“** hat folgende Bedeutungen:

„**ghz**“: der erste Wert in jeder Zeile ist die **Messfrequenz in GHz**.

„**S“**: anschließend folgen die vier S-Parameter in der Reihenfolge **S11 S21 S12 S22**

„**ma**“: jeder S-Parameter wird in der Form „erst die Amplitude (= **Magnitude**) und dann der Phasenwinkel (= **Angle**)“ angegeben.

„**r 50**“ Der **Systemwiderstand**, mit dem gemessen und gearbeitet wurde, ist ein **Ohmscher Widerstand mit 50 Ohm.**

Sehen wir uns zum Abschluss mal die **Frequenz f = 1GHz** im File an. Die vier S-Parameter lauten:

S11 = 0,32 / 144 Grad (der Betrag des Eingangsreflektionsfaktors liegt bei 0,32 und weil Blindanteile dabei sind, bekommen wir weder genau Null noch genau 180 Grad Phasenlage beim Echo...).

S21 = 18,18 / -72 Grad **(am Ausgang erhalten wir am Abschlusswiderstand mit 50 Ohm eine Spannung, die um den Faktor 18,18 größer ist als die hinlaufende Welle -- das ist also direkt die Spannungsverstärkung....)**

S12 = 0,016 / 21 Grad (Bläst man auf den Ausgang mit einem Generator drauf und schließt dabei den Eingang einfach nur mit 50 Ohm ab, dann erreichen 1,6 % der auf den Ausgang zulaufenden Welle den Eingang. Das sind die berühmten „Rückwirkungen“)

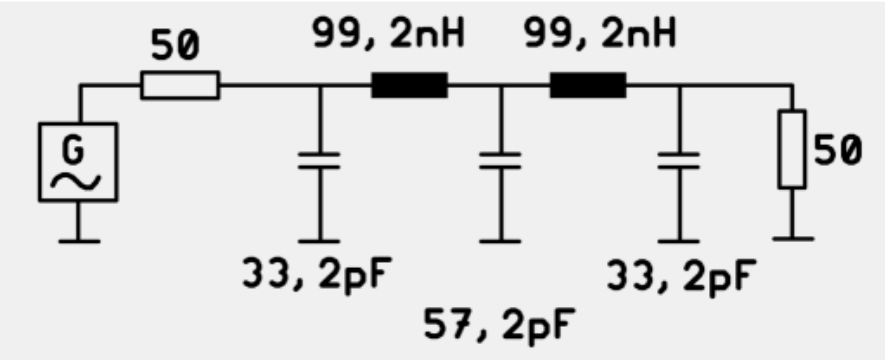
S22 = 0,5 / -20 Grad (Auch der Innenwiderstand des Bausteins weist leider keine exakten 50 Ohm auf. Wer nun die 20 Grad Phasenwinkel einfach zu Null setzt, käme auf 150 Ohm....bitte nachrechnen!)

19.2. Praxisbeispiel: 110MHz – Tschebyschef – Tiefpassfilter (LPF)

Wir wollen an einem **Tschebyschef-Tiefpass mit einer Grenzfrequenz von 110MHz die Simulation der S-Parameter demonstrieren.** Bei diesem Filtertyp nimmt man eine Welligkeit der Dämpfung im Durchlassbereich in Kauf, um einen steileren Übergang vom Durchlass- in den Sperrbereich zu erhalten.

Folgende Eigenschaften werden vorgegeben::

„Ripple“-Grenzfrequenz	fg = 110 MHz
Spulenarme Ausführung	
Filtergrad	n = 5
Systemwiderstand	Z = 50Ω
„Ripple“ (= Welligkeit der Dämpfung im Durchlassbereich) = 0,1 dB. (Zu diesem Ripple-Wert gehört ein Maximalwert von S11 = -16,4 dB)	



Für die Berechnung solcher Filter gibt es etliche Programme im Internet (= Online LC Filter Calculator). Damit erhält man folgende Bauteilwerte:

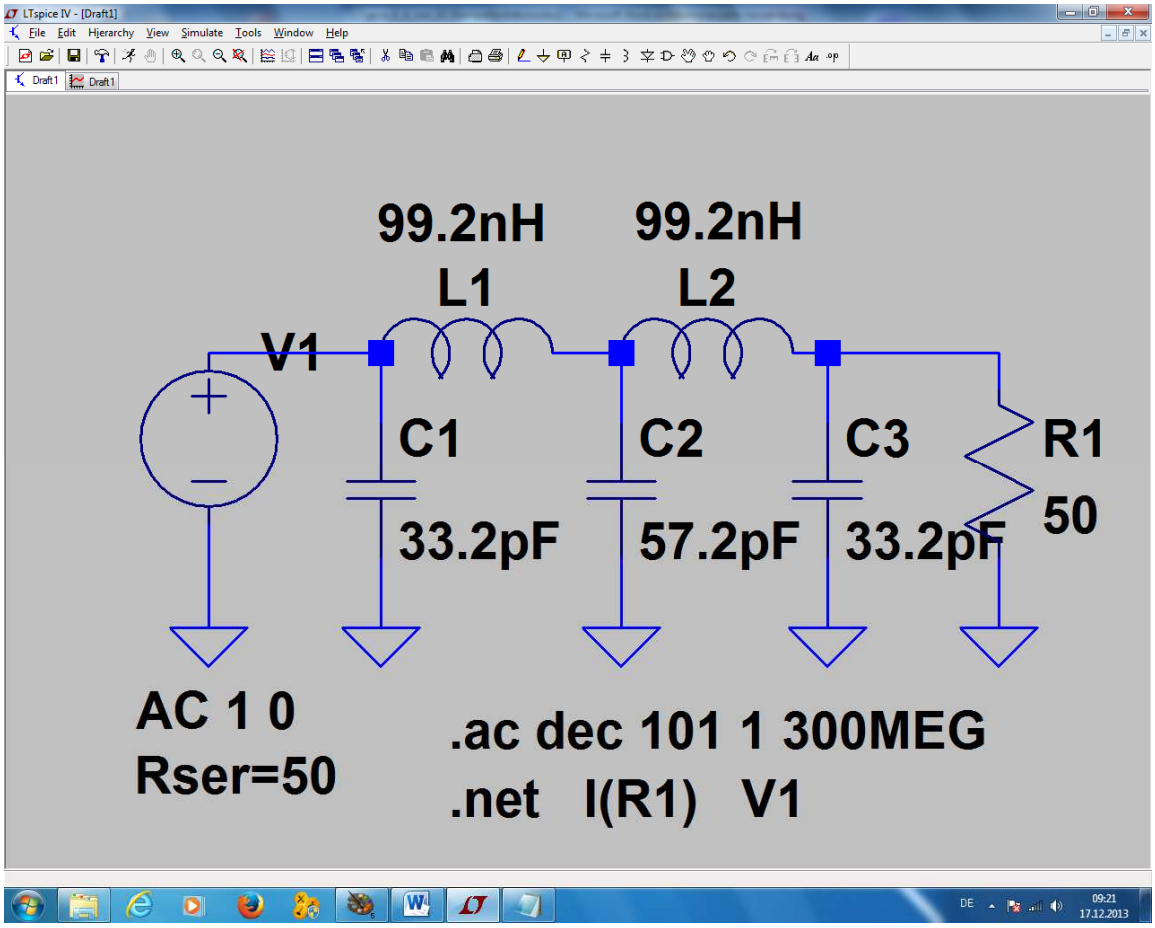
C1 = C3 = 33,2 pF

C2 = 57,2 pF

L1 = L2 = 99,2 nH

Nun kann man die Simulationsschaltung zusammenstellen:

Erläuterungen:

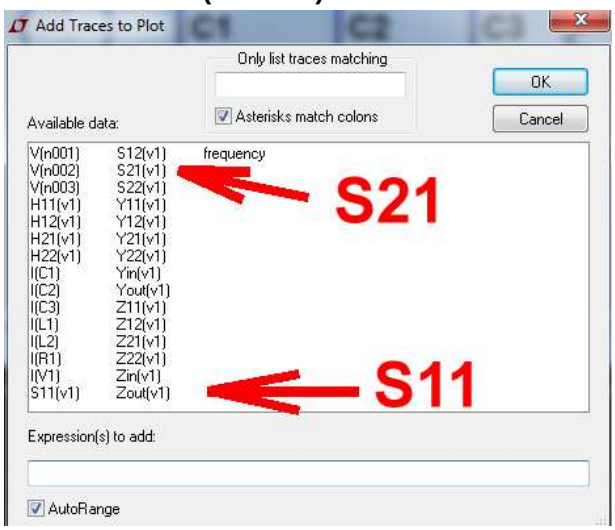


a) Die Direktive **.net I(R1) V1** löst eine **Netzwerksberechnung** aus, bei der das Ausgangssignal auf das angelegte Eingangssignal (= Quelle V1) bezogen wird.

b) Dazu programmieren wir einen dekadischen Sweep von 1Hz bis 300 MHz mit 101 Punkten pro Dekade mit der Anweisung **.ac dec 101 1 300MEG**

c) Bei den Eigenschaften der Spannungsquelle V1 stellen wir den üblichen Amplitudenwert von 1V und die Phasenlage von Null Grad durch die Angabe **AC 1 0** ein.

d) Zusätzlich aktivieren wir bei den Eigenschaften der Quelle V1 (wieder nach einem rechten Mausklick auf ihr Schaltzeichen…..) das Kästchen mit der Angabe des Innenwiderstandes von 50 Ohm **(Rser=50)**



Jetzt wird simuliert und anschließend geht es so weiter:

Wir haben zunächst ein noch leeres Ergebnisdiagramm vor uns und klicken deshalb mit der rechten Maustaste darauf. Im auftauchenden Menu wählen wir **„Add Trace“ und darin S11.**

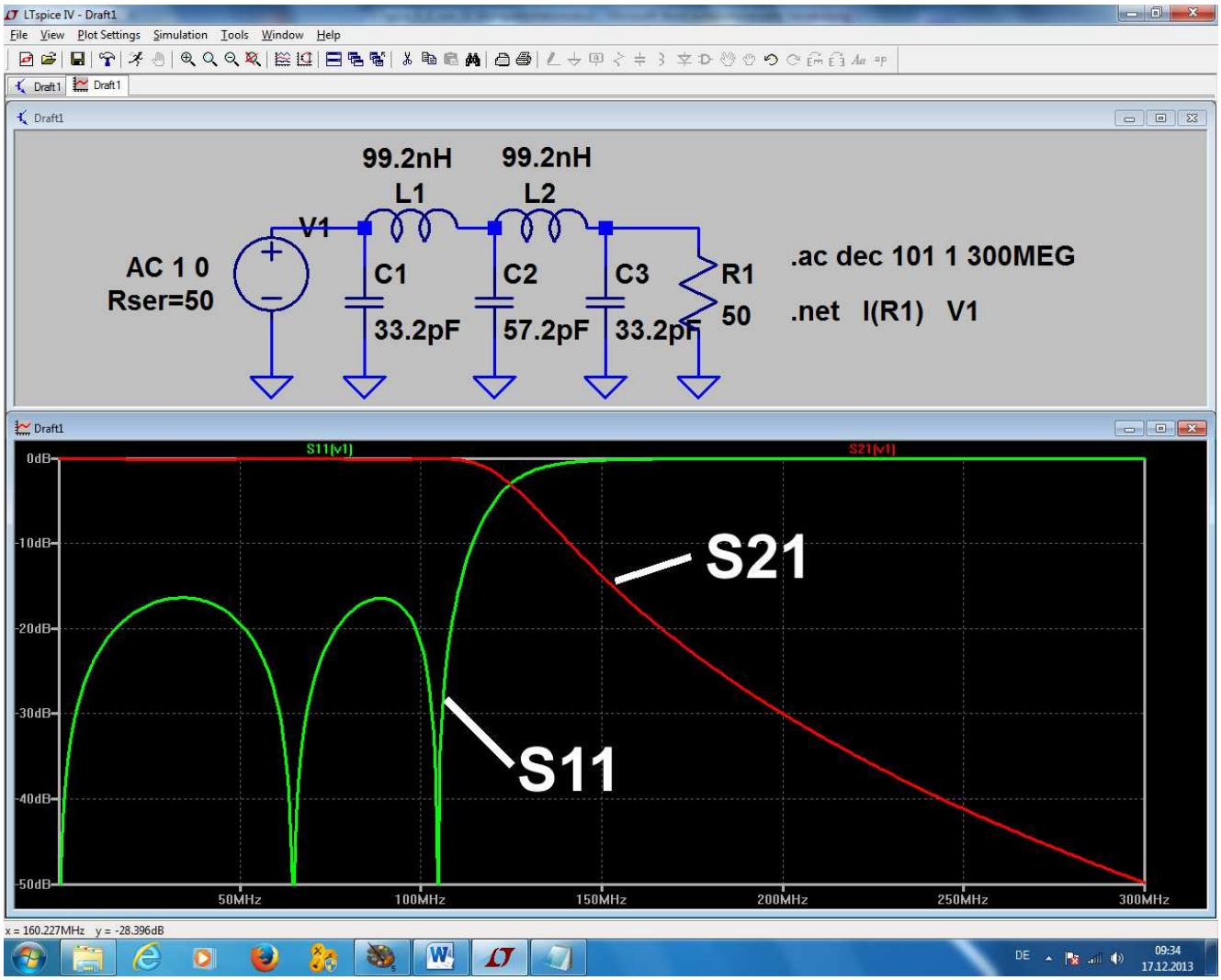
Anschließend wird die Prozedur wiederholt, aber diesmal auf S21 gewählt.

Was dann auftaucht, ist ja schon ganz brauchbar, aber noch nicht sehr schön. Außer der „Magnitude in dB“ ist nämlich auch noch der Phasenverlauf dargestellt und der interessiert uns im Augenblick nicht besonders. Auch die „dB“-Teilung der linken Achse ist noch verbesserungsfähig.

Also fahren wir mit der Maus auf die **Teilung der rechten, senkrechten Achse**, bis das Lineal erscheint. Dann klicken wir einmal darauf und wählen im auftauchenden Menü **Don’t plot phase.**

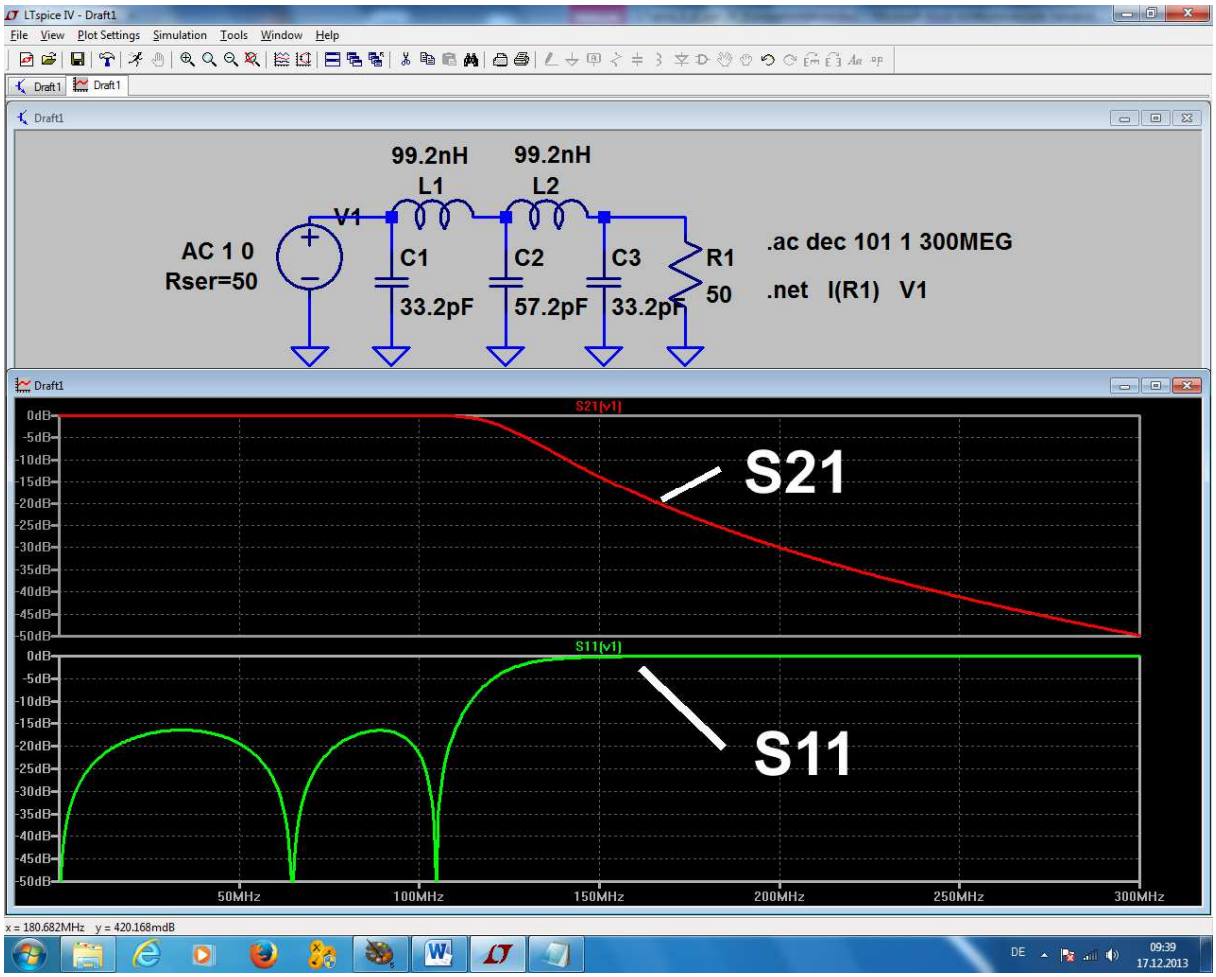
Damit ist die Phasendarstellung verschwunden.

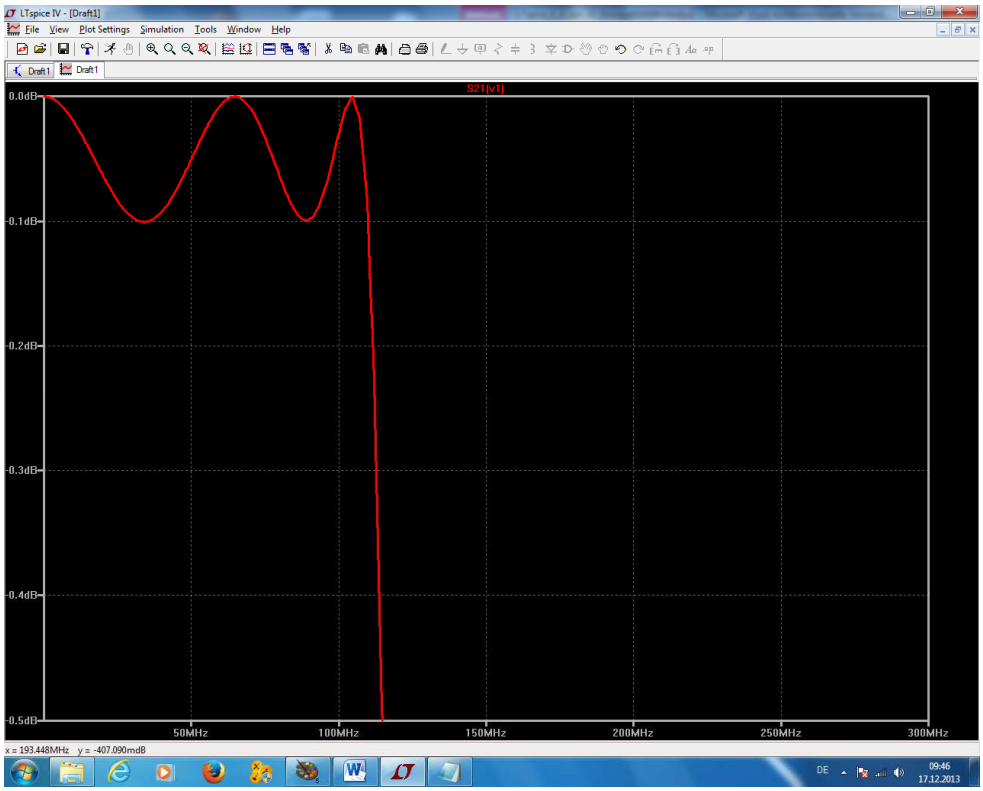
Jetzt wiederholen wir dieses Spiel mit der **linken senkrechten Achse** und sorgen für die Einstellung eines Bereiches von 0dB bis -50dB und einer 10dB-Teilung.
Die **Frequenz** an der **waagrechten Achse** stellen wir von **0…300 MHz in linearer Teilung mit einem „Tick“ von 50 MHz** dar-



Übrigens:
Wer gerne die **beiden Parameter in getrennten Diagrammen** darstellen möchte, der klickt wieder rechts auf das Diagramm und wählt **Add Plot Pane**

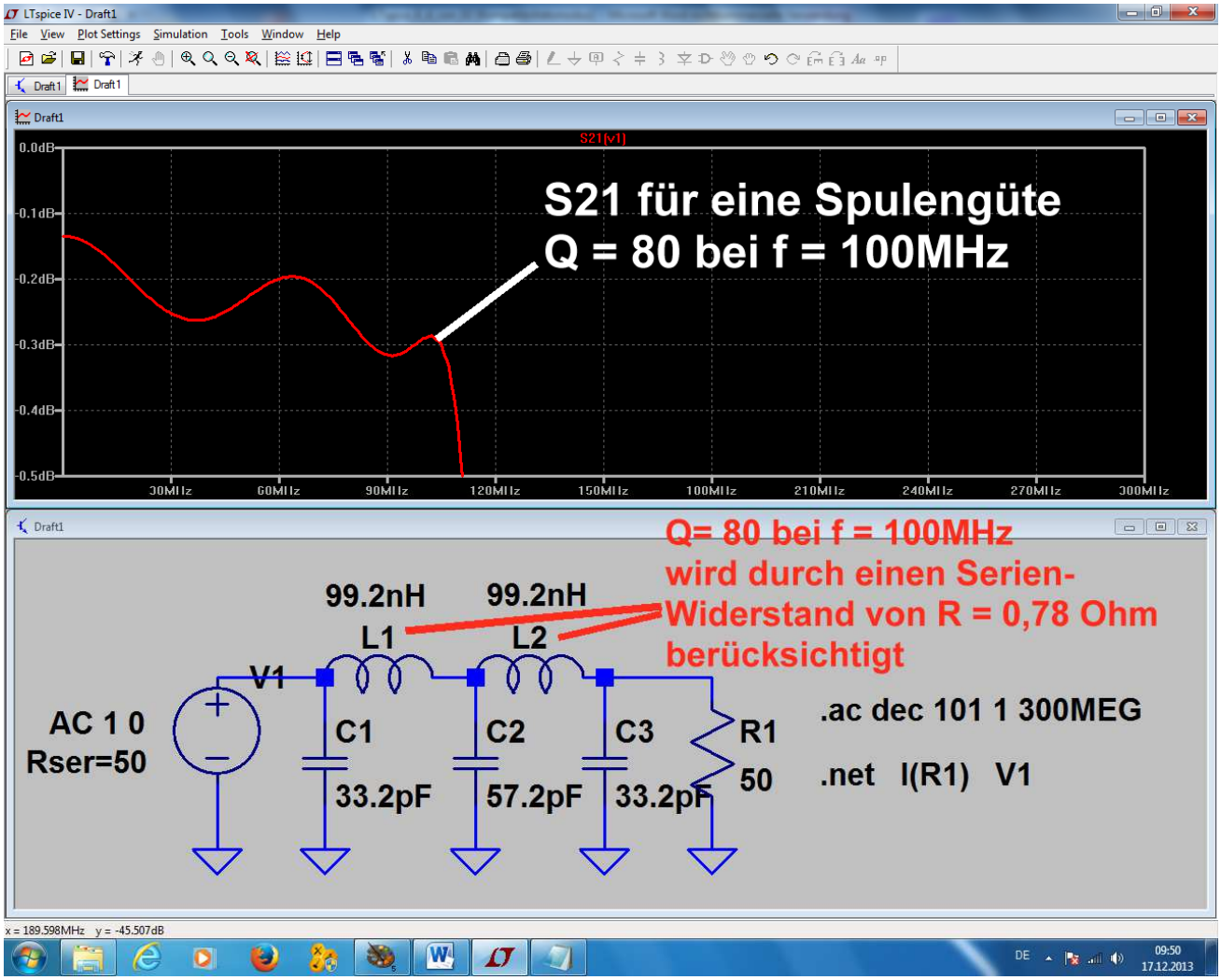
Da gibt es natürlich zuerst mal Arbeit, denn man muss im neuen Diagramm den Parameter S21 darstellen und ihn dann im alten Diagramm löschen….Wer sich da durchkämpft, sieht folgendes Bild:





Und wer sich für die Tschebyschef-Wellen im Verlauf von S21 interessiert, braucht nur die Skalenteilung der senkrechten Achse am betreffenden Diagramm zu ändern:

In der Praxis hat man es stets mit **nicht idealen Bauteilen** zu tun. **Das schwächste Glied sind immer die Spulen** und deshalb sehen wir uns an, wie sich eine **realistische Spulengüte von Q = 80 bei 100MHz** auswirkt:



Die typische Folge dieser Spulenverluste ist eine **„mit der Frequenz ansteigende Grunddämpfung“ -- die S11 Kurve „hängt also nach rechts unten“ und um diese Grunddämpfung herum schlängeln sich die Tschebyschef-Wellen.**

20. Vierzehntes Projekt: die Sache mit der Impulsantwort

In der modernen Kommunikationstechnik, Regelungstechnik und Systemtheorie wird sehr viel mit Fourier -, Fast – Fourier - und Laplace - Transformation gearbeitet.

Damit kann man z. B. die Reaktion eines Systems auf Eingangsspannungen mit beliebiger Kurvenform voraussagen, sofern man die „Übertragungsfunktion des Systems“ kennt.
Die Ermittlung dieser Übertragungsfunktion kann auf zwei verschiedene Arten -- entweder in der „time domain“ oder in der „frequency domain“ -- erfolgen.

(Die Übertragungsfunktion stellt nichts anderes als das **Verhältnis von Ausgangs- zu Eingangsspannung in Abhängigkeit von der Frequenz** dar, wobei „Betrag und Phase“ in komplexer Form dargestellt werden).

==========
Methode 1 / frequency domain:
Kennt man die Übertragungsfunktion des Systems, dann erhält man das Frequenzspektrum des Ausgangssignals durch Multiplikation von Übertragungsfunktion und Spektrum des Eingangssignals.
Zu diesem Frequenzspektrum liefert dann die inverse Fourier - bzw. Laplace - Transformation den zeitlichen Verlauf des Ausgangssignals.
==========

Hierbei macht allerdings bei komplizierten Systemen die genaue Messung oder Berechnung der Übertragungsfunktion bzw. die Bestimmung des Frequenzspektrums beim Eingangssignal oft Schwierigkeiten und man kann von einem weiteren Grundgesetz der Systemtheorie profitieren:

Führt man einem unbekannten System am Eingang einen „Dirac - Impuls“ zu, so lässt sich aus der dabei beobachteten Ausgangsspannung (= der „Impulsantwort g(t)“) unsere heiß ersehnte Übertragungsfunktion ermitteln!

Der Dirac-Impuls besitzt nämlich ein Frequenzspektrum, das von Null bis Unendlich reicht und alle Frequenzen sind darin mit gleicher Amplitude enthalten (Korrekte Bezeichnung: er hat eine „konstante spektrale Leistungdichte“ und das entspricht einem AC-Sweep mit der Startfrequenz „Null“ und der Stoppfrequenz „Unendlich“). Mit dieser so ermittelten Impulsantwort g(t) im Zeitbereich kann anschließend die Ausgangsspannung für beliebige andere Eingangssignale bestimmt werden, nämlich über

==========
Methode 2 / time domain:
Die Ausgangsspannung Ua(t) für einen beliebigen Verlauf der Eingangsspannung Ue(t) erhält man, indem man den zeitlichen Verlauf der Eingangsspannung Ue(t) mit der vorher ermittelten Impulsantwort g(t) „faltet“
(englisch: application of the „convolution integral“).

$$Ua(t) = Ue(t) * g(t)$$

Mit g(t) wird hierbei immer die Impulsantwort bezeichnet.
==========

Anmerkung:
Beim **Dirac - Impuls** handelt es sich um einen **Nadelimpuls**, der theoretisch **unendlich große Amplitude und Steigung** aufweist, aber dafür nur eine **unendlich kurze Zeit** dauert. Das ist natürlich in der Praxis nie möglich. Deshalb wird das Ganze erst sinnvoll, wenn man weiß, dass man ihn durch einen Rechteckimpuls mit hoher Amplitude annähern darf. Dessen Pulsdauer muss aber viel, viel kleiner (empfohlen: 100x bis 1000x) als die Zeitkonstante des Systems sein. Sofern sich wieder die „**Fläche 1**“ (beispielsweise 1 Voltsekunde) als Produkt aus Pulsamplitude und Pulsdauer ergibt, erhält man praktisch dieselben Ergebnisse wie beim echten Dirac - Impuls. **Und wenn es sich um ein LTI (= linear and time invariant = komplett lineares) – System handelt, darf man bei sehr kurzen Pulsen auch die Pulsamplitude und damit die Pulsfläche auf erträgliche kleine Werte reduzieren. Der Ergebnisverlauf bleibt prinzipiell gleich (…manche Simulationsprogramme mögen nämlich keine Amplituden in der Gigavolt-Gegend….)**

20.1. Erstes Beispiel: der Dirac-Impuls speist einen 160Hz-RC-Tiefpass

Wir stellen uns folgende Aufgabe:
Speisen Sie einen **Tiefpass (R = 100kΩ, C = 10nF)** mit einem Nadelimpuls, der eine **Fläche von 1 Voltsekunde** aufweist.

a) Bestimmen Sie zuerst die **Zeitkonstante** des Tiefpasses und wählen Sie dann die **Pulslänge um etwa den Faktor 1000 kürzer.**

b) Ermitteln Sie damit die erforderliche **Pulsamplitude** aus der Impulsfläche und legen Sie die Daten zur Programmierung des Puls-Spannungsverlaufs mit einer **PWL-Quelle** fest (Hinweis: **Anstiegs- und Abfallzeit jeweils 1000x kürzer als die Pulslänge**). Sehen Sie eine Startverzögerung von 1 Millisekunde vor.

c) Zeichnen Sie mit dem LTSpice-Editor die Schaltung, simulieren Sie im **Zeitbereich von 0….10ms** und stellen Sie **Eingangs- und Ausgangssignal** in getrennten Diagrammen dar („Tile Horizontally“). Der **„Maximum Time Step“ sei 100 Nanosekunden** -- das ergibt bei 10ms Simulationszeit 100 000 echte Samples.
Bauen Sie aber nach dem Start erst mal eine **Verzögerungszeit („= Delay Time) von 1 Millisekunde** ein und schalten Sie die Datenkompression ab.

d) Führen Sie für das **Eingangssignal eine FFT mit 65536 Punkten** (= Samples) durch. Analysieren Sie das Ergebnis und begründen Sie die Forderung nach a). Stellen Sie es anschließend auf lineare Darstellung um.

e) Führen Sie für das **Ausgangssignal eine FFT** durch.

Im Frequenzbereich ist eine **lineare Darstellung von 0….20kHz mit einem Tick von 1kHz** gefordert.

Lösung zu a): Ermittlung der nötigen Pulslänge

Es gilt:

$$f_{Grenz} = \frac{1}{2\pi \bullet RC} = \frac{1}{2\pi \bullet 100k\Omega \bullet 10nF} = 159Hz$$

Das ergibt:

$$RC = \frac{1}{2\pi \bullet f_{Grenz}} = \tau = 100k\Omega \bullet 10nF = 10^{-3}\,s = 1ms$$

Also wird die Pulslänge 1000x kürzer, nämlich **1 Mikrosekunde,** gewählt.

Lösung zu b): Simulation mit dem PWL-Signal

Eine Pulslänge von 1 Mikrosekunde und eine Fläche von 1Voltsekunde ergeben eine **Pulsamplitude von 1Megavolt. Die Anstiegs- und Abfallzeit muss jeweils 1 Nanosekunde (= 1µs / 1000) betragen.**

Wir brauchen eine PWL-Quelle und programmieren sie folgendermaßen:

Bei T = Null: Amplitude Null

Bei T = 1ms Amplitude Null

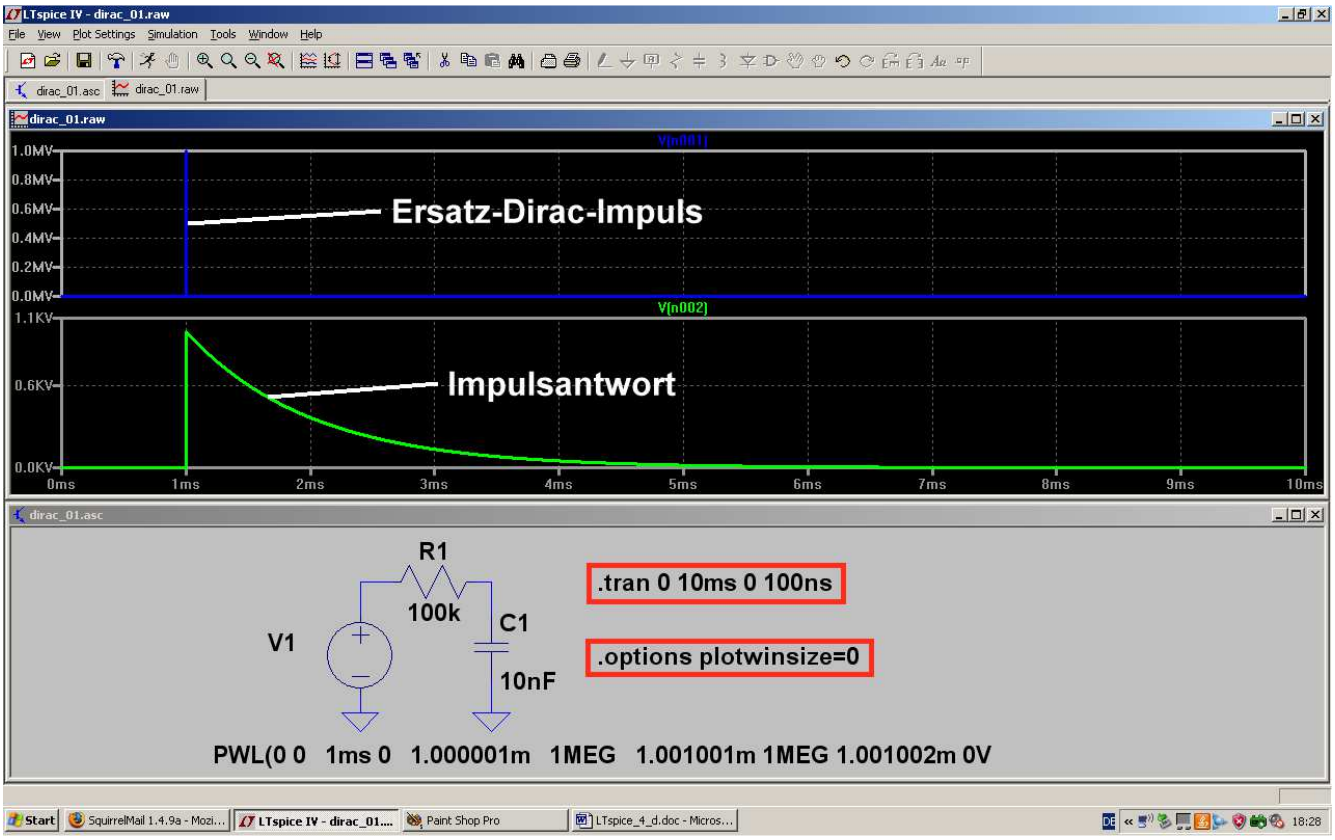
Bei T = 1.000001ms Amplitude 1MEG

Bei T = 1.001001ms Amplitude 1MEG

Bei T = 1.001002ms Amplitude Null

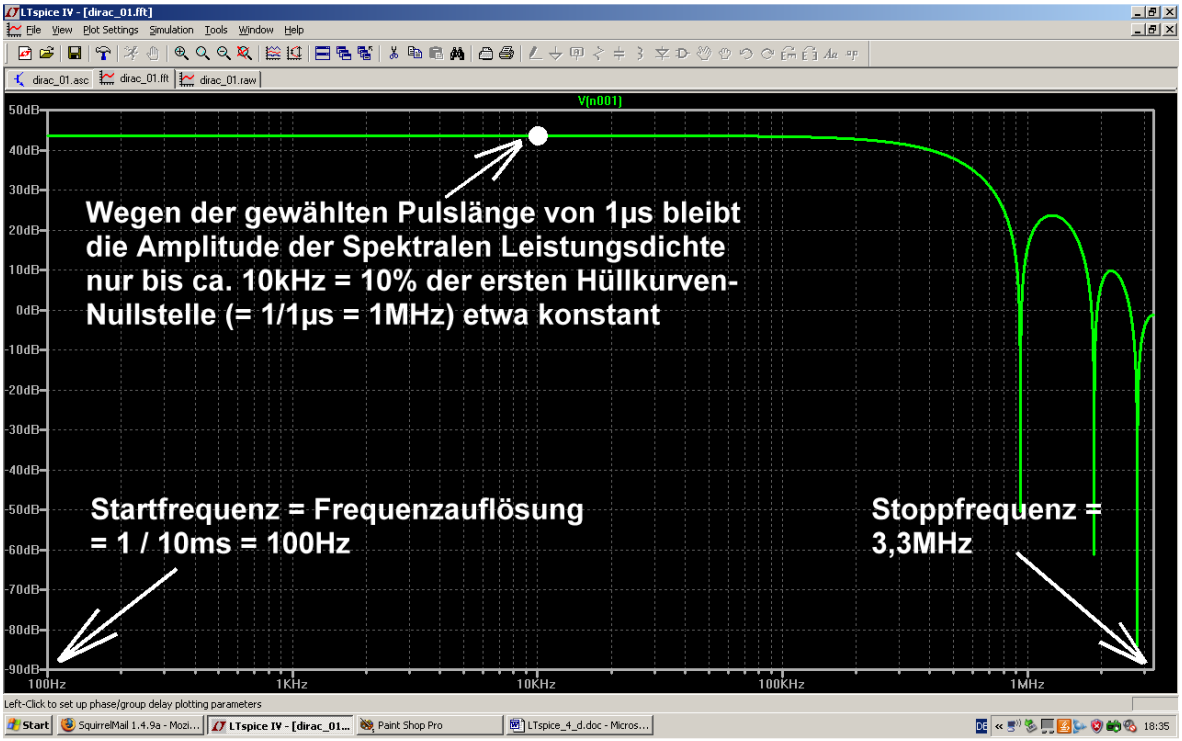
Lösung zu c) Schaltung und Simulation in der Time Domain

Die Programmierung der PWL-Quelle sowie das Simulationskommando finden sich in der unteren Schirmhälfte.
(…Bitte nie vergessen, die Datenkompression über **.option plotwinsize=0** abzuschalten. Nur dann fuktioniert übrigens auch die Interpolation…Siehe FFT-Menü).



Lösung zu d): FFT der Eingangsspannung

Wir rollen den Cursor auf das Diagramm mit dem Eingangspuls und klicken mit der rechten Maustaste darauf. Im auftauchenden Menü suchen wir erst „**View“, dann „FFT“** und stellen schließlich **65536 Samples** ein, bevor wir OK drücken. Das bekommen wir zu sehen:



Wichtig:

a) **Die Startfrequenz des Diagramms entspricht gleichzeitig der „besten Frequenzauflösung und der minimalen Linienbreite“, also dem kleinsten möglichen Frequenzschritt.** Das wird durch die **Simulationszeit im Zeitbereich** festgelegt und so erhält man bei t_{max} = 10ms

$$1 / t_{max} = 1 / 10ms = 100Hz$$

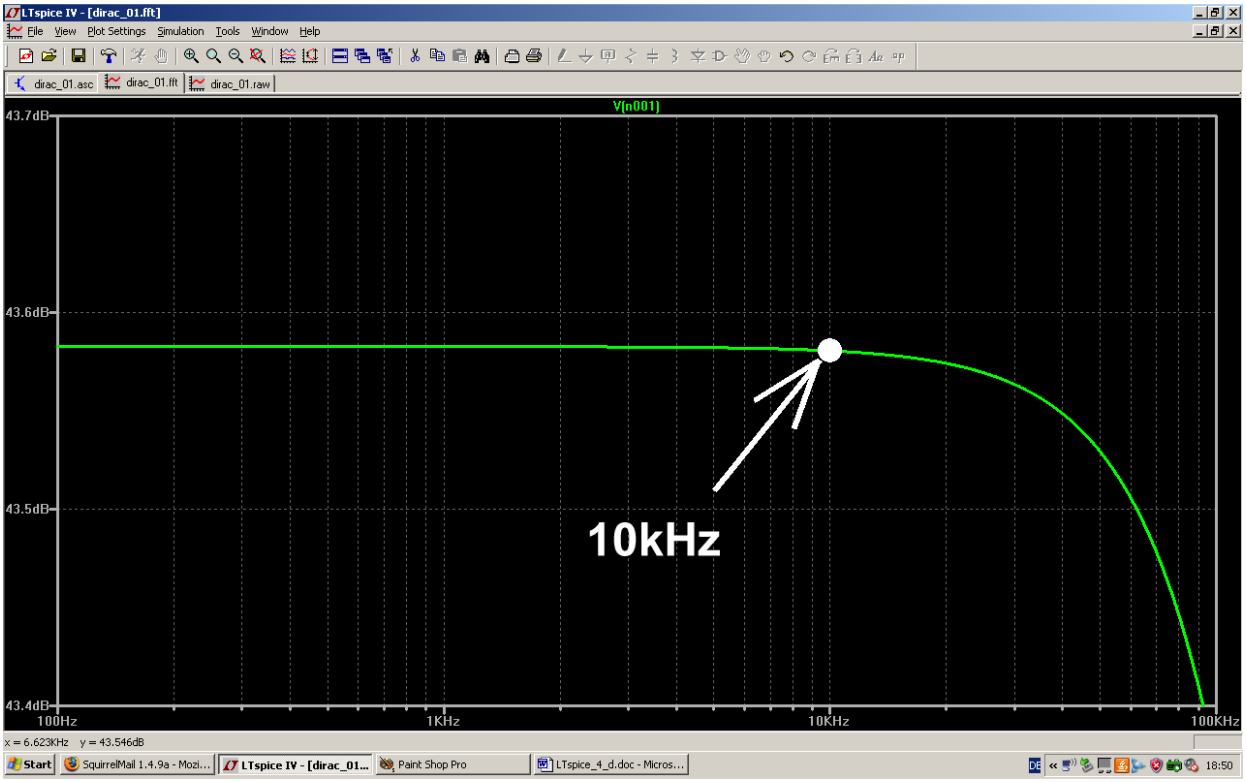
b) **Die höchste im Diagramm darstellbare Frequenz (= Stoppfrequenz beim AC-Sweep….) sowie der Dynamikbereich der FFT-Anzeige werden durch die Samplefrequenz bei der Time-Domain Simulation UND durch die Anzahl der für die FFT eingesetzten Samples bestimmt**. (Die Samplefrequenz bei der Time Domain Simulation erhält man als 1/ maximaler Timestep).

c) Wir haben eine Pulslänge von 1 Mikrosekunde gewählt und das führt zu einer ersten „**Nullstellenfrequenz bei der Hüllkurve“** von

$$1 / 1\mu s = 1MHz$$

Da es sich bei der Hüllkurve um einen „sin x / x“ – Verlauf handelt, bleibt dadurch die Spektrale Leistungsdichte nur bis etwa 1% der ersten Nullstellenfrequenz (…hier also bis 10kHz) konstant.

Das entspricht einem AC-Sweep (…der ja mit konstanter Amplitude beim Eingangssignals arbeitet….) mit einer Stoppfrequenz von ca. 10kHz. **Kontrollieren wir das doch mal, indem wir bei der Frequenzachse nur den Bereich von 100Hz bis 100kHz zulassen und außerdem den Amplitudenbereich stark aufdehnen**:



Na also…

Zusatzaufgaben:

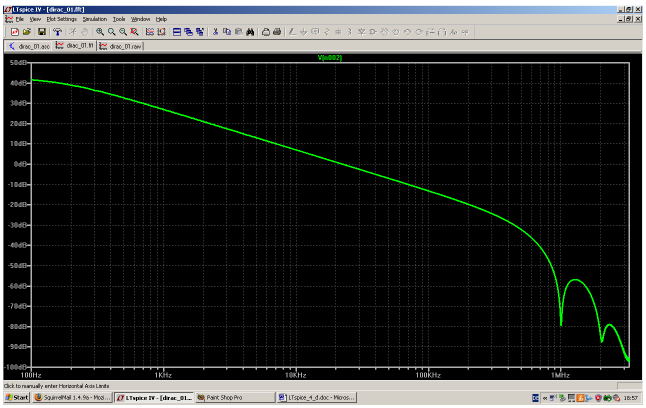
1) Testen Sie die Behauptung a) durch Simulationen mit unterschiedlichen **Simulationszeiten** (z. B. 20ms / 50ms / 100ms).

2) Testen Sie die Behauptung b) durch die FFT desselben Eingangssignals mit unterschiedlichen **Sample-Mengen** (z. B. 16384 / 32768 / 131072 / 262144)

Lösung zu e): FFT der Impulsantwort des Tiefpasses

Time Domain und Frequency Domain sind nur die zwei Seiten DERSELBEN MÜNZE und beschreiben vollständig alle Eigenschaften unseres untersuchten Objektes!

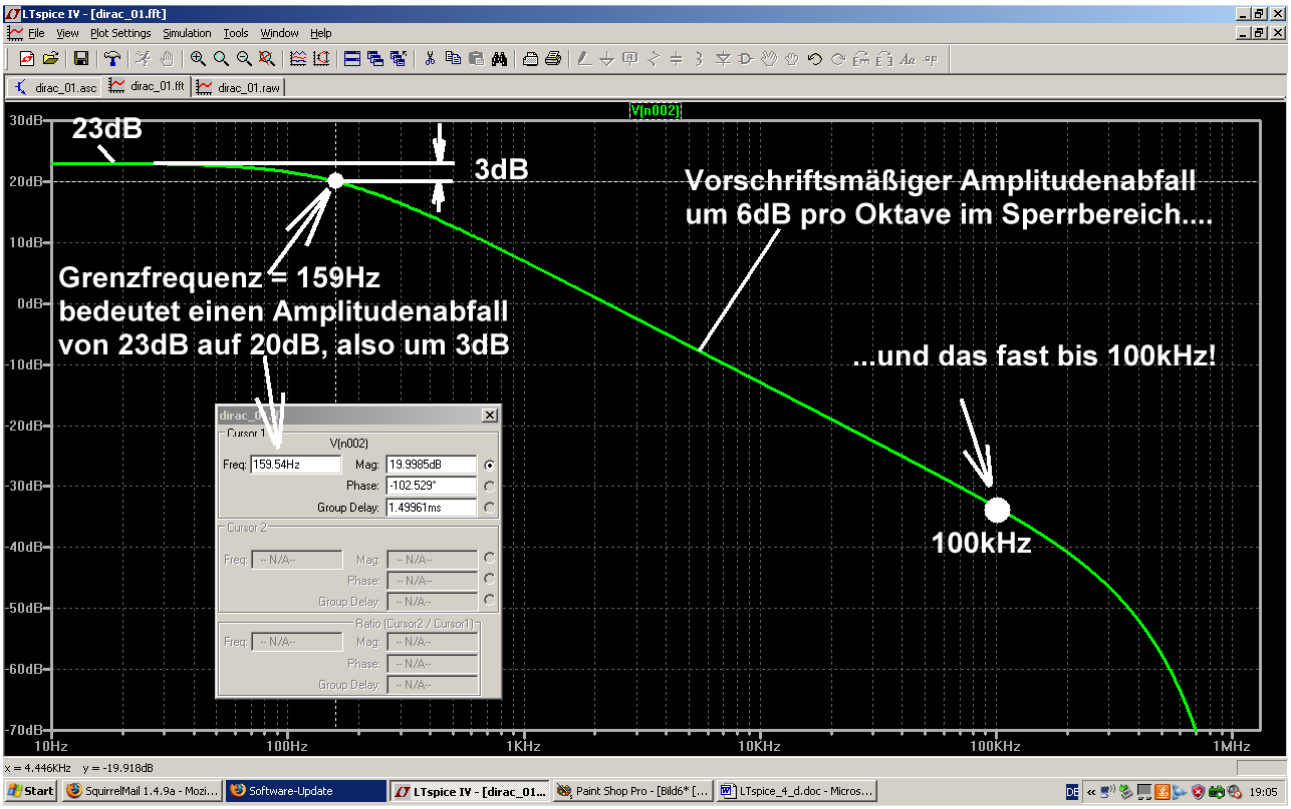
Folglich muss bei der „Rücktransformation der Impulsantwort in den Frequenzbereich“ (korrekt: durch den dadurch angewandten Prozess namens Convolution = Faltung) die Übertragungsfunktion des Bausteines sehen sein, zu dem diese Impulsantwort gehört. Und das ist die bekannte Kurve, die von den Regelungstechnikern im Bode-Diagramm auch mit „Amplitudengang“ bezeichnet wird.
Also werfen wir mal einen Blick auf die vom FFT-Programm ausgegebene Kurve.



Zunächst sehen wir nur -- wie es sich für einen solchen Tiefpass gehört -- den Amplitudenabfall im Sperrbereich mit -6dB pro Oktave.
Bei hohen Frequenzen tauchen natürlich die Schwächen unseres Ersatz-Dirac-Impulses mit den erwähnten Nullstellen in der Hüllkurve auf. Deshalb sind dort die Ergebnisse gelogen und unbrauchbar!
Schlimmer ist, dass die Startfrequenz des Diagramms mit 100Hz (entsprechend einer Simulationszeit von 10ms) für einen 160Hz-Tiefpass schlicht und einfach zu hoch ist.

Also wiederholen wir die Simulation mit einer Simulationszeit von 0…100ms und bekommen dadurch eine Auflösung bzw. einen Diagrammstart von 10Hz! Den minimalen Timestep dürfen wir auf 200ns erhöhen und erhalten trotzdem noch 500 000 echte Samples

Nun sieht man wirklich alles bei einer FFT mit 262 144 Samples:



Auswertung: Siehe nächste Seite!

Die Startfrequenz und damit auch die Frequenzauflösung beträgt wirklich 10Hz.

Mit steigender Frequenz kommen wir zur Grenzfrequenz von 159Hz und genau dort hat die Amplitude um 3dB abgenommen.

Im Sperrbereich sinkt die Amplitude vorschriftsmäßig um -6dB pro Oktave.

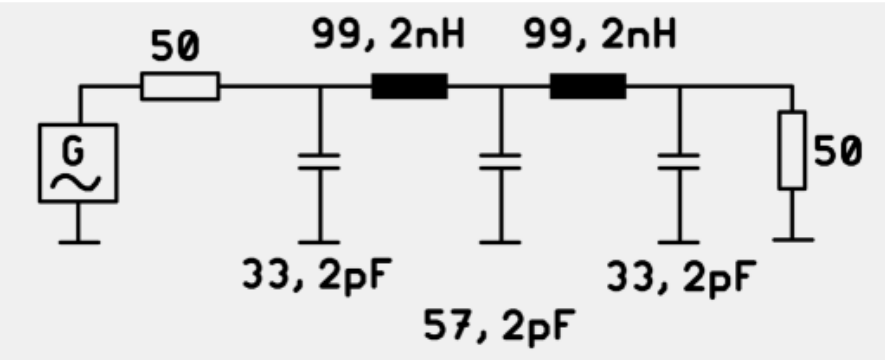
Oberhalb von 15kHz (= etwa der hundertfachen Grenzfrequenz des Tiefpasses) beginnt der Einfluss der Hüllkurve unseres Ersatz – Impuls-Spektrums und ab diesem Bereich darf man irgendwann nicht mehr alles glauben. Abhilfe bringt da nur eine deutliche Verminderung der Pulsdauer. Das schiebt die erste Hüllkurven-Nullstelle nach oben.

20.2. Zweites Beispiel: Dirac-Untersuchung des bekannten 110MHz-Tiefpasses

20.2.1. Bestimmung von S21 (= Forward Transmission)

Das waren die damaligen Vorgaben:

„Ripple“-Grenzfrequenz	**fg = 110 MHz**
Spulenarme Ausführung	
Filtergrad	**n = 5**
Systemwiderstand	**Z = 50Ω**
„Ripple“ (= Welligkeit der Dämpfung im Durchlassbereich) =	**0,1 dB.**
(	



…und das sind die von einem Filter-Calculator gelieferten Bauteilwerte für die Schaltung.

Nun stellen wir uns folgende Aufgabe:

a) Führen Sie einen „normalen“ AC-Sweep bei der Schaltung durch und simulieren Sie den Verlauf des **S-Parameters S21 bis 1000MHz** (…er entspricht der Übertragungsfunktion des Filters). Dehnen Sie anschließend die Ergebnis-Darstellung soweit, dass die „Tschebyschef-Wellen“ von 0,1dB im Durchlassbereich gut zu erkennen sind.

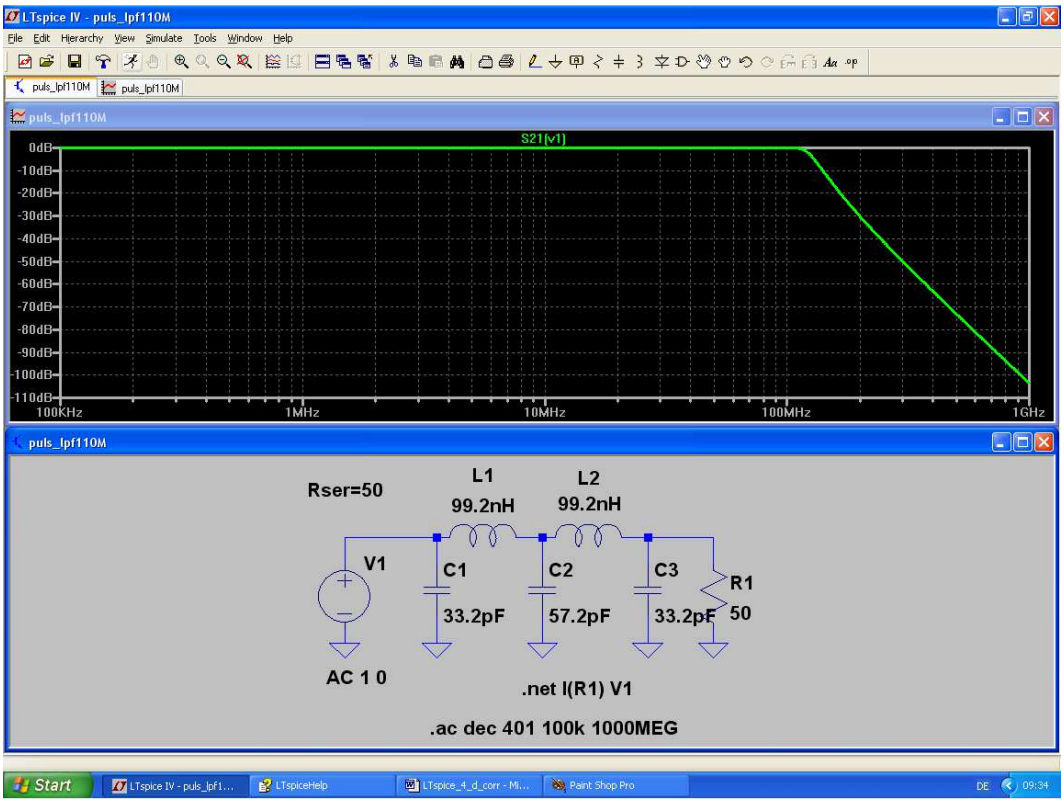
b) Ersetzen Sie dann die Spannungsquelle am Eingang durch einen **angenäherten Dirac-Impuls**, dessen Pulsdauer um den Faktor 100 kleiner ist als die Zeitkonstante des Filters. Anstiegs- und Abfallzeit sollen wieder um den Faktor 100 kleiner sein als die Pulslänge. Wählen Sie eine Pulsamplitude von maximal 1 KiloVolt und eine Simulationszeit, bei der Sie im Ergebnisdiagramm eine **Startfrequenz bzw. eine Frequenzauflösung von 100kHz** erhalten. Bauen Sie eine „Startverzögerung“ von einer Nanosekunde ein.

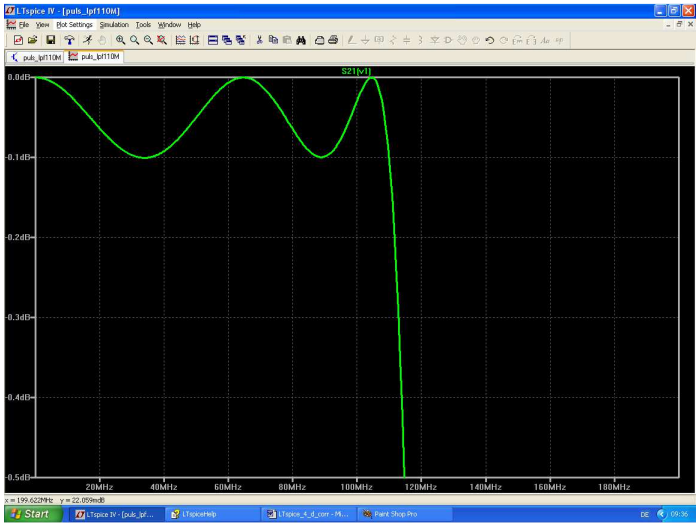
c) Wählen Sie den „maximalen Zeitschritt“ (= Maximum Time Step) nur so hoch, dass der Dirac-Impuls vom Programm garantiert nicht übersehen werden kann (z. B. mindestens 2…3 Samples pro Dirac-Impuls-Dauer). Simulieren Sie in der Time Domain. **Schalten Sie dabei die Datenkompression aus.**

d) Berechnen Sie, wie viel echte Samples zur Verfügung stehen. Führen Sie nun eine FFT der Impulsantwort mit **der maximal möglichen Anzahl an Samples** durch. Demonstrieren Sie die Leistungsfähigkeit der „Dirac-Impulsantwort-Methode“, indem Sie anschließend beim Durchlassbereich die „**Tschebyschef-Wellen“ und ihre Amplitude von 0,1dB sichtbar machen**. Vergleichen Sie das Ergebnis mit der S-Parameter-Simulation nach a).

e) **Verwenden Sie zur FFT nur 65 536 echte Samples. Z**eigen Sie am Ergebnisdiagram die Auswirkung dieser Maßnahme.

Lösung zu a):
Das ist der Verlauf von S21 bis 1GHz….





….und dies natürlich das „Ripple“ (= Tschebyschef-Wellen im Durchlassbereich) mit einer Amplitude von 0,1dB.

Das wollen wir als „Sollwert“ für die folgende Arbeit verwenden.

Lösung zu b) und c):

Die Grenzfrequenz des Tiefpasses beträgt 110MHz. Dazu gehört eine Zeitkonstante von etwa

$$\tau = \frac{1}{2\pi \bullet f_{Grenz}} = \frac{1}{2\pi \bullet 110MHz} = 1{,}44ns$$

Da wählen wir die **Pulslänge** unseres Dirac-Impulses 100x kürzer, also zu ca. **15 Picosekunden**.

Die **Anstiegs- und Abfallzeit** sind jeweils 100x kürzer, also je **0,15ps = 0,00015ns.**

Zusammen mit der Startverzögerung von 1ns und der maximalen Amplitude von 1 KiloVolt erhalten wir dann folgende Wertepaare für die Programmierung der PWL-Quelle:

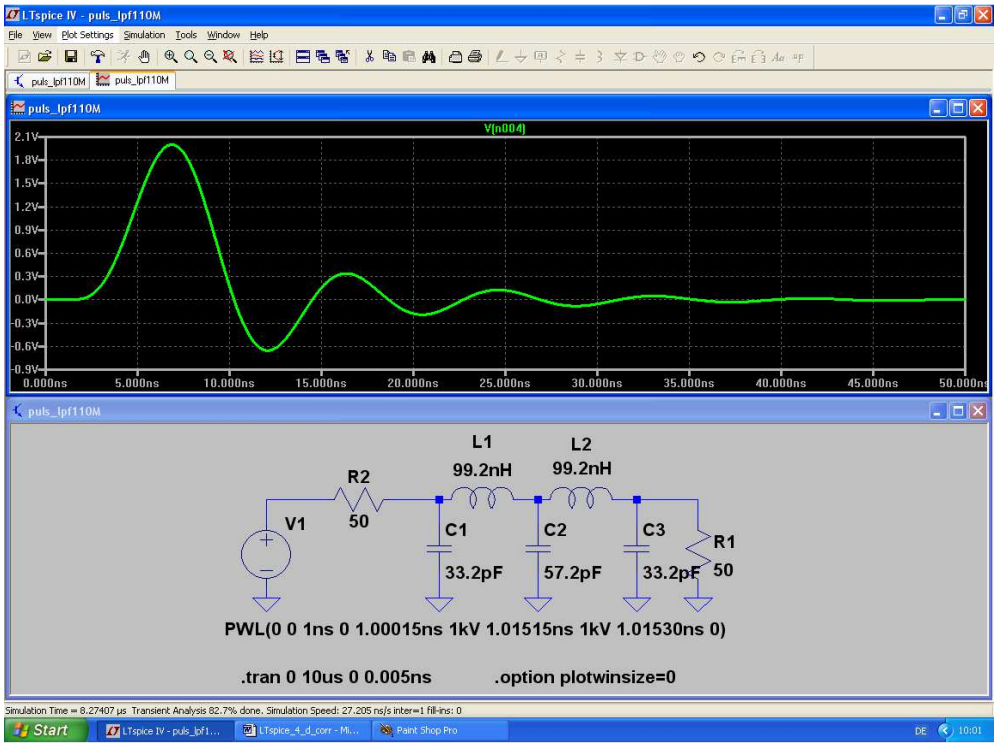
Zeitpunkt T = Null:	**U = Null**
Zeitpunkt T = 1ns:	**U = Null**
Zeitpunkt T = 1,00015ns:	**U = 1 KiloVolt**
Zeitpunkt T = 1,01515ns:	**U = 1 KiloVolt**
Zeitpunkt T = 1,01530ns:	**U = Null**

Gefordert ist eine Startfrequenz von 100kHz (…die gleichzeitig der „Frequenzauflösung beim Ergebnisdiagramm“ entspricht). Deshalb muss die Simulationszeit (in der ja die Informationen in Form der Samples gesammelt werden) folgenden Wert aufweisen:

$$t_{max} = 1 / 100kHz = 10\mu s$$

Damit LTspice den Dirac-Impuls nicht übersieht, wird der **Maximum Time Step zu 15ps / 3 = 5 Picosekunden** gewählt.

Die Datenkompression ist durch die Direktive **„.option plotwinsize=0“** abgeschaltet.

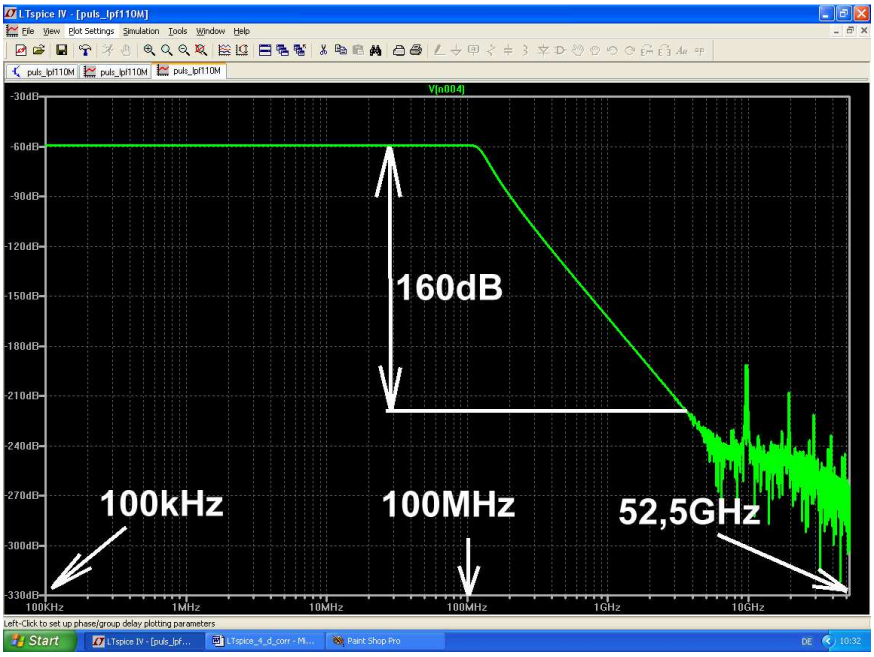


So sieht das aus, wenn wir uns nur die ersten 50 Nanosekunden anschauen:

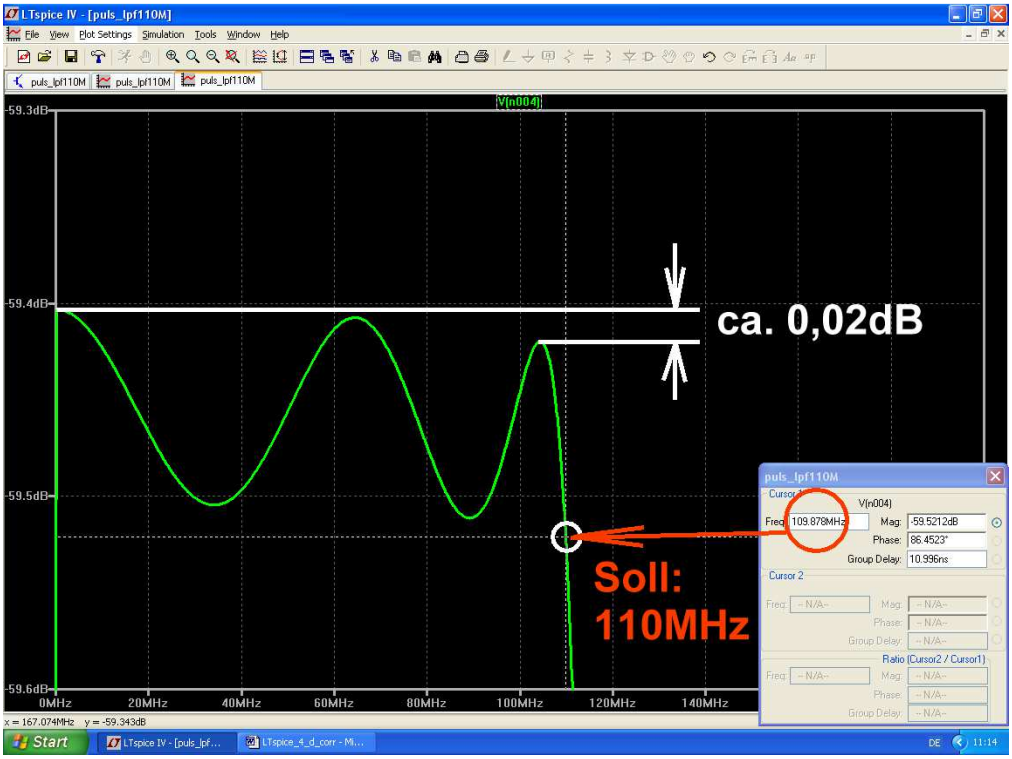
Lösung zu d): FFT
Wir simulieren 10µs lang mit einer maximalen Auflösung von 5Picosekunden. Das entspricht einer minimalen Anzahl von

10µs / 0,005ns = 2 000 000 Samples

Also können wir es wagen, bei der FFT mit **1048576 Werten** zu arbeiten.
Das Transformationsergebnis mit linearer Frequenzachse sieht zunächst so aus:



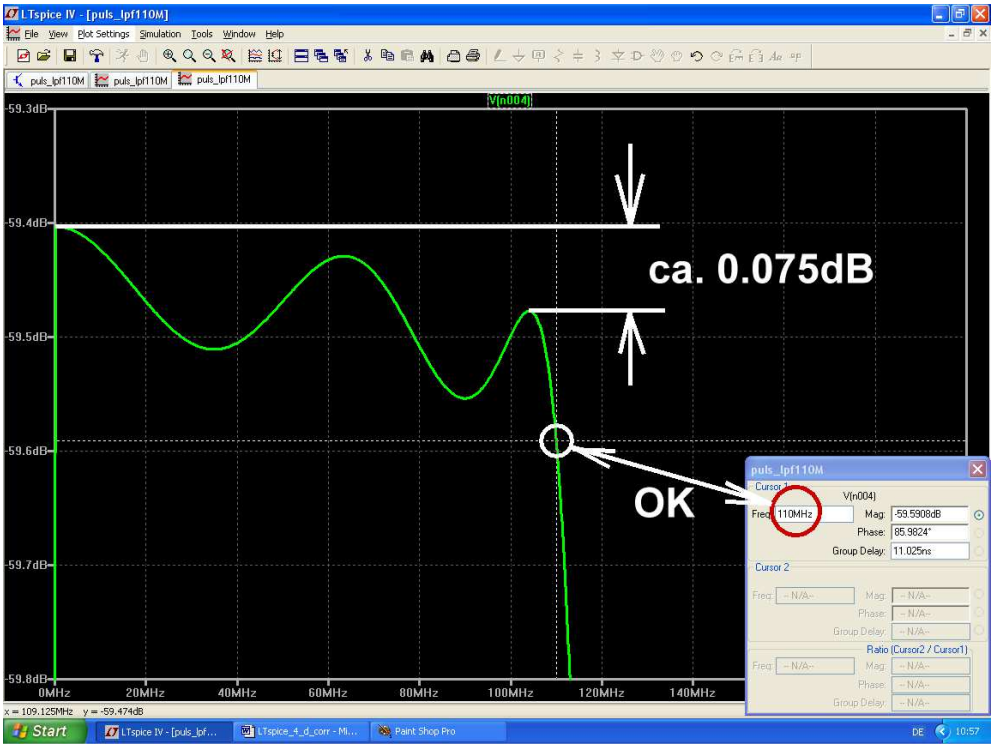
Werfen wir noch einen weiteren Blick auf die Tschebyschef-Wellen im Durchlassbereich und vergleichen sie mit dem am Anfang vorgenommenen AC-Sweep:



Die Abweichungen gegenüber der S-Parameter-Simulation sind fast unbedeutend.

Lösung zu e): Simulation mit 65536 Samples

Da muss natürlich manches etwas schlechter werden. Als erstes fällt beim Sweep über den vollen Bereich auf, dass die höchste angezeigte Frequenz im Diagramm entsprechend sinkt -- von 52,5GHz auf 3,2GHz. Und ab 500MHz…1GHz ist es mit der Genauigkeit der Simulation auch nicht mehr so weit her..



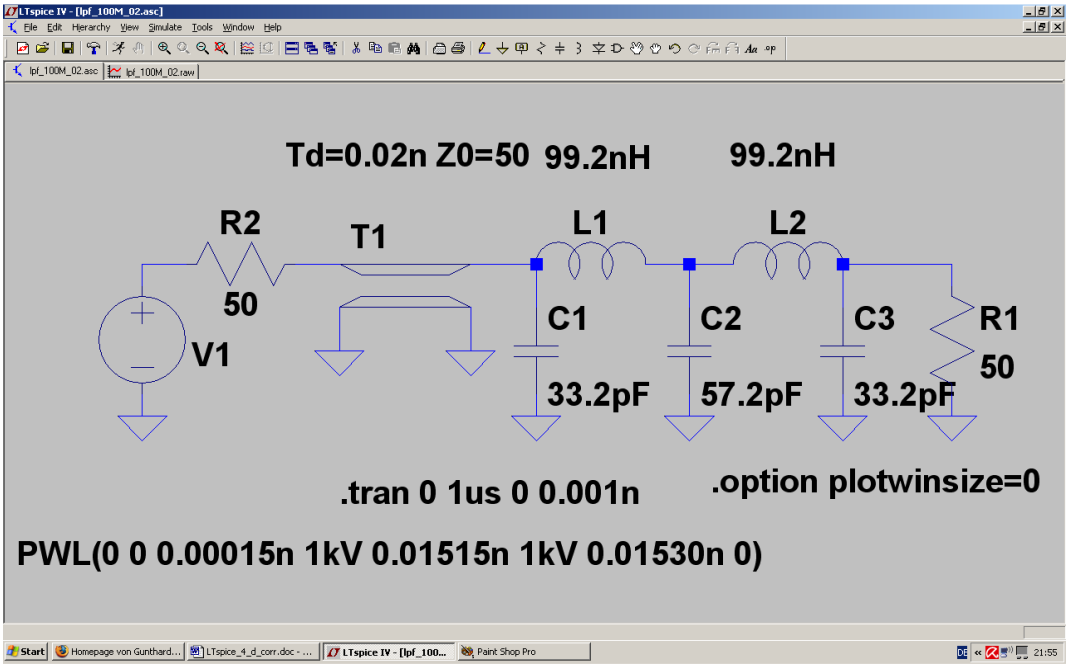
Und bei den Tschebyschef-Wellen im Passband steigt der Fehler leider entsprechend an. Die Ripple-Grenzfrequenz wird dagegen korrekt bestimmt.

20.2.2. Bestimmung von S11 (= Input Reflection) oder S22 (= Output Reflection)

Außer der „Forward Transmission S21“ (bzw. Reverse Transmission S12) interessieren auch die “Echos“ oder „Reflektionen“ S11 und S22 (= Reflections) am Eingang bzw. Ausgang eines Twoports. In der Praxis sind dazu Richtkoppler erforderlich, die das Echo (also die Reflektierte Welle = Reflected Wave) von der Hinlaufenden Welle (= Incident Wave) trennen.

In der Simulation mit dem Dirac-Impuls geht das wesentlich einfacher. Wir schalten zwischen Generator und Messobjekt einfach ein kurzes Leitungsstück mit Z = 50Ω. Die Kabellaufzeit brauchen wir nur etwas länger als die Pulsdauer zu wählen. Dann erscheint am Kabeleingang mit einer Verzögerung von 2 x (Kabellaufzeit) das Echo und kann zur FFT herangezogen werden. Das Ergebnis der FFT in dB ist dann direkt der S-Parameter S11 (oder S22).

Zuerst die verwendete Simulationsschaltung für S11:



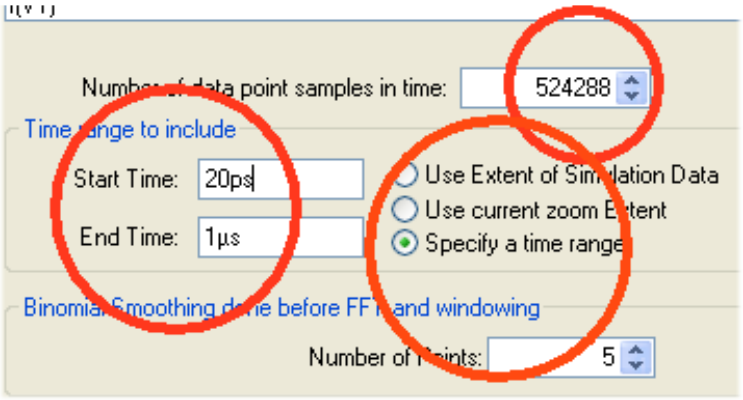
Die Signalquelle V1 ist über das erwähnte Kabelstück mit **Z = 50Ω** und einer Laufzeit von **20 Picosekunden** mit dem Eingang des Tiefpasses verbunden. Der Dirac-Impuls weist eine Länge von **15 Picosekunden** und eine Anstiegs- bzw. Abfallzeit von **je 0,15 Picosekunden** auf. Er startet sofort beim Zeitpunkt Null und wird als PWL-Signal mit folgender Wertefolge erzeugt:

Zeitpunkt T = Null:	**U = Null**
Zeitpunkt T = 0,00015ns:	**U = 1 KiloVolt**
Zeitpunkt T = 0,01515ns:	**U = 1 KiloVolt**
Zeitpunkt T = 0,01530ns:	**U = Null**

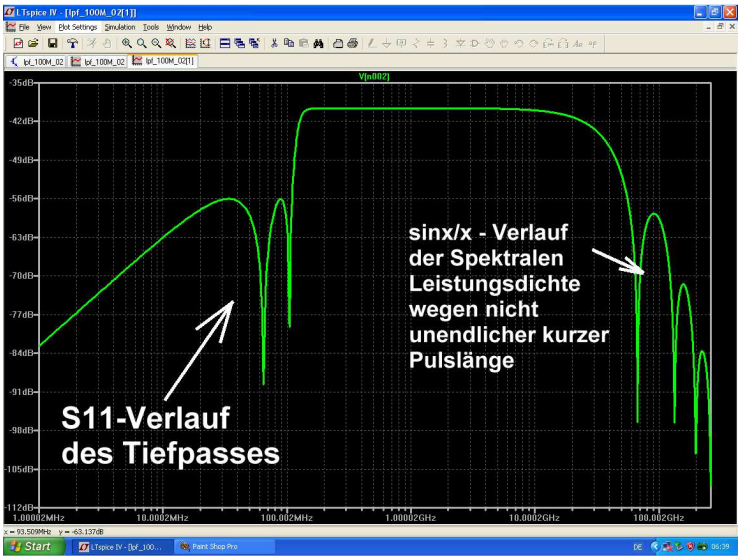
Damit erscheint das Echo nach **40 Picosekunden** am Kabeleingang und dort wird auch gemessen. **Für die FFT muss deshalb der Zeitraum von 0…20 Picosekunden (= Bereich des Eingangs-Pulses) unterdrückt werden.**

Die Datenkompression ist wie immer abgeschaltet (**.option plotwinsize=0**).

Um die Rechenzeit nicht zu lang werden zu lassen, wurde nur eine Simulationszeit von 1µs vorgesehen, was eine Frequenzauflösung von **1 / 1µs = 1MHz** ergibt (….reicht aber für den S11-Verlauf aus). **Dafür wurde die maximale Schrittweite auf 1 Picosekunde festgelegt, denn das Echosignal ist kaum länger als der Eingangs-Dirac-Impuls mit 15 Picosekunden und da sollte man einfach dessen Verlauf möglichst genau erfassen.**
Für die FFT stehen damit **1µs / 0.001ns = 1 000 000** berechnete Samples zur Verfügung.

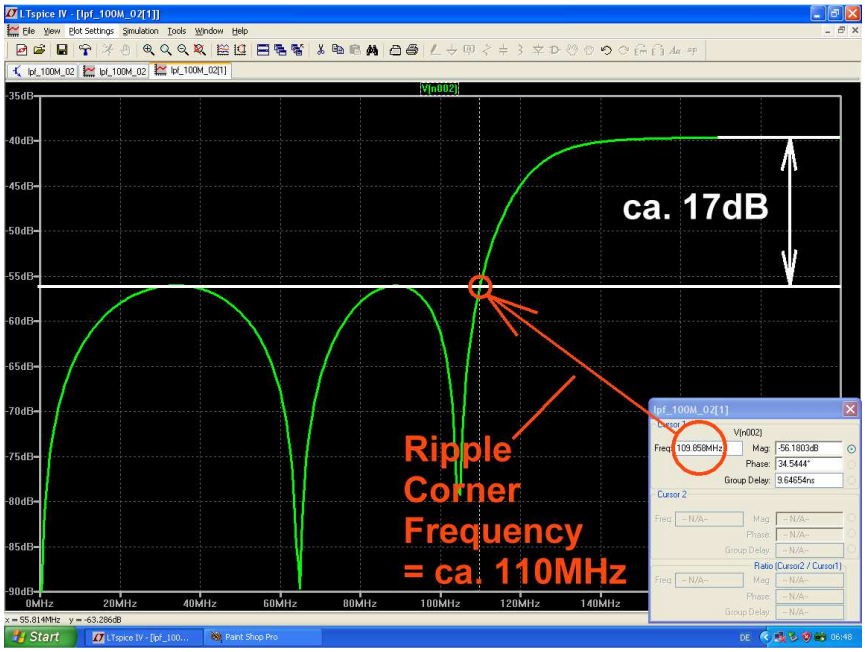


Dann geht es an die Einstellungen für die FFT. Verwendet wird der Zeitbereich ab 20 Picosekunden bis 1µs und die maximal mögliche Anzahl von 524288 Samples.



So sieht das Simulationsergebnis aus. Bei sehr hohen Frequenzen finden wir schon die Eigenheiten unseres nachgemachten Dirac-Impulses (….ist also uninteressant…), aber immerhin reicht das Diagramm bei der Zeitauflösung von 1 Picosekunde und der maximal möglichen Samplezahl bis 250GHz…

Zoomen wir deshalb nur den Bereich von 0 bis 200MHz und wählen wir eine lineare Frequenzachse:



Da brauchen wir uns vor der S-Parameter-Simulation in Kapitel 14.2. nicht zu verstecken.

21. Fünfzehntes Projekt: Rauschen -- Grundlagen, Anwendung, Simulation mit LTspice

21.1. Erste Bekanntschaft

Empfängt man eine Nachricht ohne jegliche Information, dann ist darin alles zufällig und ungeordnet und ergibt keinen Sinn oder ermöglicht keine Voraussagen. Jeder kennt das als „Rauschen“ und das klingt wirklich wie das Rauschen eines Wasserfalls oder einer Wasserleitung. Jede Antenne empfängt ein solches Signal aus der Atmosphäre oder aus dem Weltraum, jeder elektrische Widerstand oder Leiter erzeugt von sich aus eine solche Spannung durch die „Wärmebewegung der Leitungselektronen“, sofern wir nicht bis zum absoluten Nullpunkt abkühlen. Man drehe einfach mal sein UKW-Radio oder sein TV-Gerät auf einen Leerkanal zwischen zwei Stationen, dann weiß man, was gemeint ist. Aber Vorsicht: bei verschlüsselten Sendungen klingt das oft genau so wie Rauschen...und doch steckt Information drin. Das ist allerdings Absicht, um eventuelle Mithörer zu täuschen! Auch „künstlich erzeugtes Rauschen“ (= pseudo random noise) ist oft nie ganz perfekt.

Für „**ideales Rauschen“ = „Weißes Rauschen = Johnson Noise**“ gilt aber:
Seine Leistungsdichte für jedes Hertz an Bandbreite ist über der Frequenz konstant und das ist typisch für die in Ohm’schen Widerständen durch die Elektronen-Wärmebewegung entstehenden Rauschspannungen

(Sobald man hier diesen Leistungsdichte-Verlauf z. B. durch Filter verändert, spricht man vorn „Rosa Rauschen“). Doch nun zur Frage:

21.2. „Rauschen“ -- woher kommt das?

Das lässt sich schnell und präzise beantworten: in jedem elektrischen Widerstand werden Elektronen bewegt, wenn Strom fließt. Sobald aber Wärme mit im Spiel ist (…..und das ist automatisch oberhalb des absoluten Nullpunktes stets der Fall…), werden diese Teilchen immer unruhiger und nehmen nicht den geraden Weg von Minus nach Plus. Sie stoßen zusammen, prallen zurück, werden nach vorne oder zur Seite geschleudert…. der Strom schwankt also durch den Wärme-Einfluss dauernd und völlig unregelmäßig um kleinere und größere, aber winzige Beträge. Selbst wenn ***gar keine äußere Spannung angelegt ist***, merken wir dieses Gerangel unter den Ladungsträgern und messen an den Anschlüssen des Bauteils eine kleine „Rausch-Leerlaufspannung U_{NOISE}“. Sie lässt sich folgendermaßen berechnen:

$$U_{NOISE} = \sqrt{\frac{4hfBR}{e^{\frac{hf}{kT}} - 1}} \qquad (1)$$

mit
h = Planck’sches Wirkungsquantum
k = Boltzmann – Konstante = 1,38 x 10^{-23} J / Kelvin
T = absoluter Temperatur in Kelvin
B = betrachtete Bandbreite in Hz
f = Mittenfrequenz der betrachteten Bandbreite in Hz
R = Widerstandswert in Ω

Das sieht nun entsetzlich kompliziert und praxisfremd aus, aber man kann bis mindestens 100 GHz und Temperaturen bis hinunter zu 100 K problemlos die einfache (und bekannte) Näherungsformel verwenden:

$$\mathbf{U_{NOISE} = \sqrt{4kTBR}} \qquad (2)$$

Stellt man diese Formel etwas um, dann sieht man plötzlich sofort, was los ist:

$$\frac{\left(\frac{\mathbf{U_{NOISE}}}{\mathbf{2}}\right)^{\mathbf{2}}}{\mathbf{R}} = \mathbf{kT \bullet B} \qquad (3)$$

Das ist schlicht und einfach eine Leistungsangabe! Also wird in jedem Widerstand -- unabhängig von seinem Widerstandswert! -- durch den **Einfluss der Wärme die gleiche Rausch-LEISTUNG entstehen.** Sie ergibt die in Formel (2) angegebene Leerlaufspannung U_{NOISE}.

a) Sobald man also diesen Widerstand als Spannungsquelle mit Urspannung U_{NOISE} und (rauschfreiem) Innenwiderstand R auffasst, muss man an diese Quelle einen Lastwiderstand mit dem gleichen Wert R anschließen, um Leistungsanpassung zu erhalten. **Am Lastwiderstand liegt dann die halbe Urspannung aus Formel (2) und an den Lastwiderstand wird die verfügbare Rauschleistung „kTB“ abgegeben**.

b) Die Rauschleistung steigt linear mit der absoluten Temperatur des Bauteils und der zur Verfügung gestellten Bandbreite (…die Spannung folgt dann natürlich mit der Wurzel aus der Leistung). Sie ist normalerweise unabhängig von der Frequenz und trägt in einem solchen Fall den Namen „**Weißes Rauschen“.**

Ganz wichtig ist nun folgende Sache:
Fast immer arbeitet man bei Empfängern und Systemen mit „Pegeln“ anstelle von Spannungswerten. Das ist ein logarithmisches Maß und damit kann man das **Multiplizieren** (wenn z. B. Stufen in Reihe geschaltet sind und ihre Verstärkungen multipliziert werden müssen…) durch eine **Addition** ersetzen. Am bekanntesten dürfte hier die Einheit „dBm“ sein. Dabei wird nicht mit Spannung, sondern mit Leistung gearbeitet und jeder vorhandene Leistungswert wird ins Verhältnis zum Bezugswert

$$\mathbf{P_0 = 1\ Milliwatt\ am\ Systemwiderstand}$$

gesetzt. Das ergibt den

$$\textbf{Leistungs-Pegel} = \mathbf{10 \bullet \log\left(\frac{Leistungswert}{1mW}\right)} \quad \text{in dBm} \quad (4)$$

Betrachtet man nun die obige Sache mit der Rauschleistung „kTB“ etwas genauer, dann lässt sich eine interessante Vereinfachung einführen:

$$\mathbf{kTB = (kT) \bullet B = (Rauschleistungsdichte) \bullet (Bandbreite)}$$

Die Rauschleistungsdichte „kT“ stellt die Leistung in jedem „Hertz an Bandbreite“ dar und man muss sie mit der geltenden Bandbreite multiplizieren, um die gesamte produzierte Rauschleistung zu erhalten.

Geht man nun auf die Pegelrechnung über, dann sollte man folgendes wissen und gut abspeichern:
Jeder Widerstand produziert damit bei Raumtemperatur (T_0 = 300K) einen **internen Rauschpegel** und damit eine interne Rauschleistungsdichte von etwa

-168dBm pro Hz Bandbreite

Dadurch kann er an einen **rauschfreien** Lastwiderstand bei **Leistungsanpassung einen um 6dB niedrigeren Leistungspegel** von

-174dBm pro Hertz

abgeben.

Bei größeren Bandbreiten als 1 Hz ist der Rest dann ganz einfach:

Tatsächlicher maximal abgebbarer Rauschpegel in dBm = -174dBm + 10log(Bandbreite in Hz)

Aufgabe:

a) Wie groß ist **IM LEERLAUF** der Rauschpegel und die Rauschspannung an einem Widerstand von 50 Ohm und 300 Kelvin, wenn mit einer Bandbreite von 1Hz gearbeitet wird?“ **Bestätigen Sie diese Ergebnisse durch eine LTSpice-Simulation.**

Lösung:
Der Rauschpegel = -168dBm (Siehe oben) entspricht einer Leistung von

$$P = 1mW \bullet 10^{\frac{-168}{10}} = 1mW \bullet 10^{-16,8} = 1,6 \bullet 10^{-20}\,W$$

Daraus wird schließlich mit einem „Quellwiderstand von 50 Ohm“ eine Leerlaufspannung von

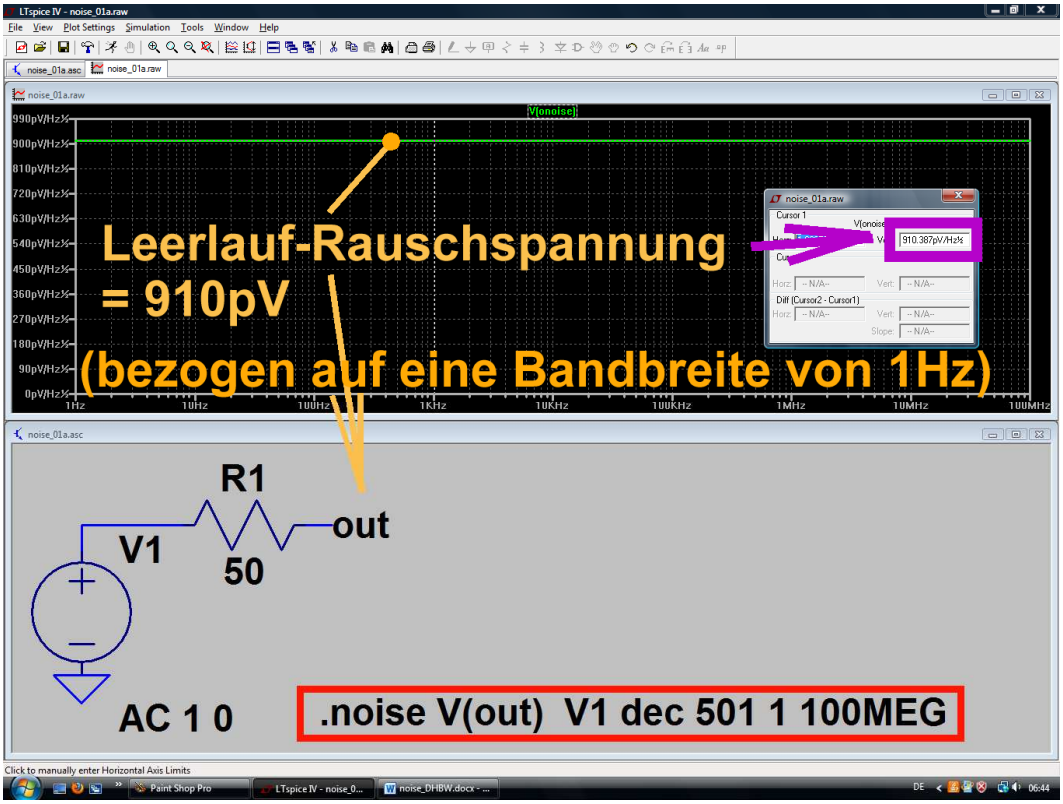
$$U_{NOISE} = \sqrt{P \bullet R} = \sqrt{50\Omega \bullet 1,6 \bullet 10^{-20}\,W} = 910pV$$

Und so sieht die LTspice-Simulation aus:
Man braucht dazu eine **normale Spannungsquelle im „AC-Sweep-Betrieb**“ (= Einstellung „**AC 1 0**“ bei der Small Signal AC Analysis) und einen 50Ω-Widerstand.

Mit dem SPICE Kommando

.noise V(out) V1 dec 501 1 100MEG

programmieren wir eine Rauschanalyse beim Label “out”. Bezugswert ist dabei eine Rauschspannung, die in Reihe zur Spannungsquelle V1 zu denken ist und vom Widerstand R1 erzeugt wird (…R1 selbst wird dabei als rauschfrei angenommen).
Der Rest ist einfach: erzeugt wird ein dekadischer Sweep mot 501 Werten pro Dekade im Frequenzbereich von 1 Hertz bis 100 MegaHertz.



Wie vorhergesagt, erhalten wir eine Leerlaufspannung von 0,91 NanoVolt pro 1 Hz.

Aufgabe:

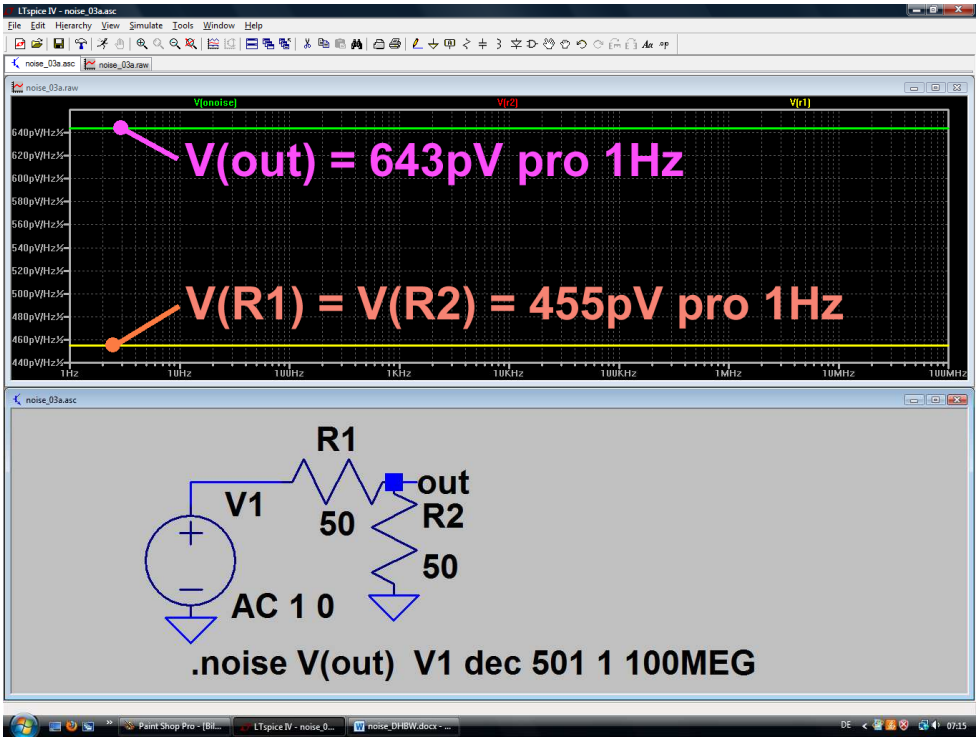
b) Wie groß werden der Rauschpegel und die Rauschspannung am Ausgang, **wenn diese Quelle mit einem rauschfreien Widerstand von 50 Ohm abgeschlossen wird?**

Das ist eine sehr einfache Übung, denn nun haben wir einen einfachen Spannungsteiler, bestehend aus zwei rauschfreien 50Ω-Widerständen, vor uns. Dieser Spannungsteiler wird von einer Urspannung mit 0,91 NanoVolt gespeist.
Also erhält man am Ausgang die halbe Urspannung (= 0,455 NanoVolt) und den Pegel von -174dBm pro 1Hz.
==========
Aufgabe:

c) Wie groß werden der Rauschpegel und die Rauschspannung am Ausgang, **wenn diese Quelle mit einem normalen, also rauschenden Widerstand von 50 Ohm abgeschlossen wird?**

Da sehen wir uns gleich mal die zugehörige Simulation an und klicken auf „out“, gefolgt von einem Klick auf das Schaltzeichen von R1 bzw. R2:

Das Programm geht dabei so vor:
Es betrachtet zunächst den rauschenden Widerstand R1 als eine Quelle mit der Rausch-Urspannung von 910pV und dem rauschfreien Innenwiderstand R1= 50Ω. Diese Quelle wird mit dem rauschfreien Widerstand R2 = 50Ω belastet und das ergibt 0,455nV = 455pV pro 1Hz beim Punkt „out“.



Nun wiederholt es diese Rechnung für den Widerstand R2 als Rauschquelle und das führt ebenfalls zu 455pV pro 1Hz bei „out“. Am Ende addiert man diese beiden Teilsignale nach dem „Überlagerungs-satz“.

Allerdings hat man es dabei mit „nicht korrelierten Signalen“ zu tun und darf deshalb nur die Effektivwerte addieren:

$$U_{gesamt} = \sqrt{(U_1)^2 + (U_2)^2} = \sqrt{(455pV)^2 + (455pV)^2} = 643pV$$

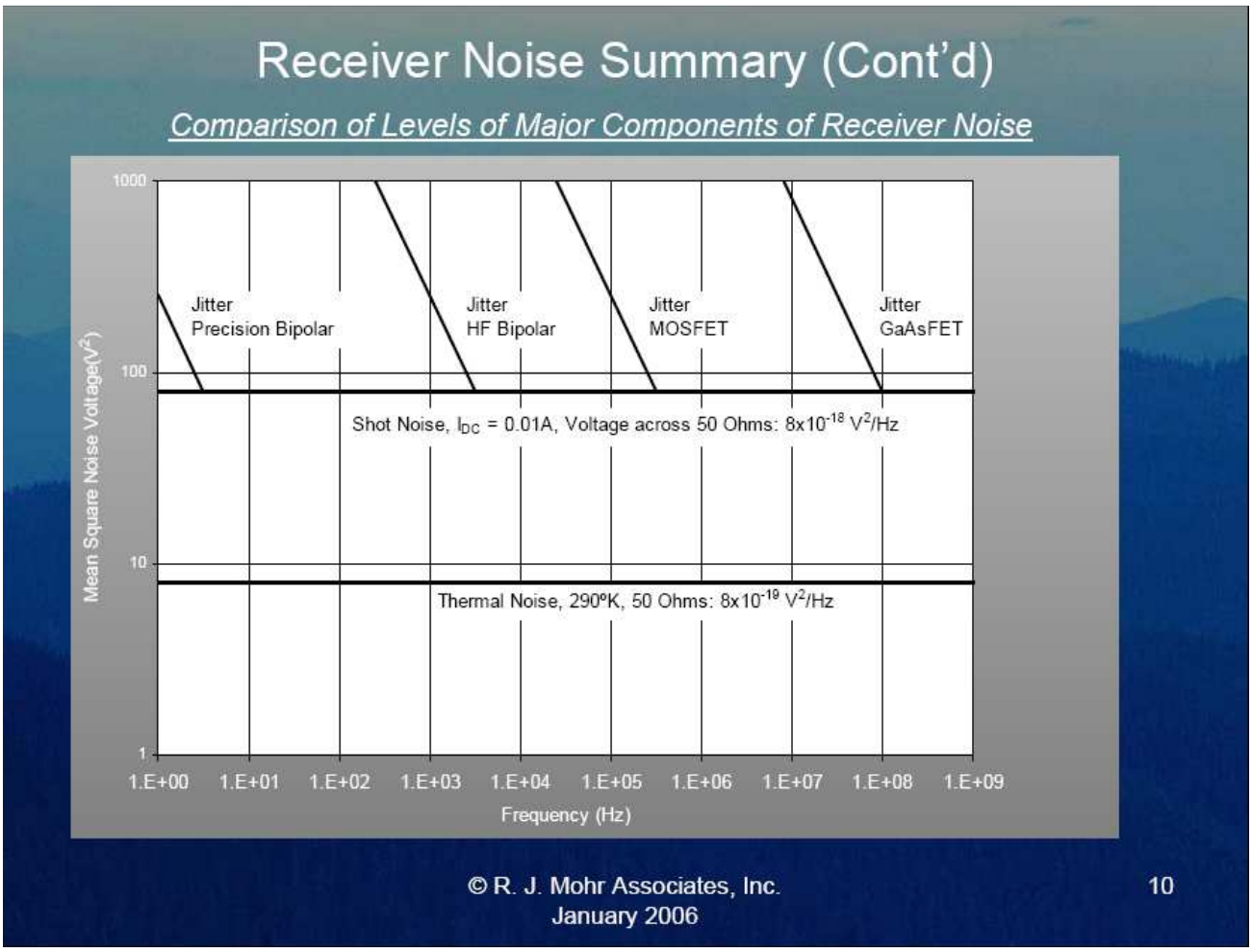
Genau das zeigt uns die Simulation…

21.3. Weitere Rauschquellen

In jedem aktiven Bauteil (Röhre, Bipolar-Transistor, Sperrschicht-FET, MOSFET, HEMT usw.) haben wir es außer mit dem thermischen Rauschen noch mit zwei zusätzlichen Rauscharten zu tun:

a) ***Shot-Noise*** (= Schrot-Rauschen) tritt bei Vakuumdioden und P-N-Übergängen durch Ungleichförmigkeiten des Stromflusses beim Durchqueren der Potential-Unterschiede auf. Es handelt sich hierbei um breitbandiges, weißes Rauschen.

b) ***Flicker-Noise*** oder „1 / f – Noise“ (= Funkel-Rauschen) entsteht durch Verunreinigungen oder Defekten im Kristallaufbau. Sie führen zu kurzen impulsförmigen Schwankungen beim Stromfluss -- und zu einem Impuls gehört eben ein Spektrum, dessen Leistungsdichte mit steigender Frequenz abnimmt. Man definiert hier wie bei einem Tiefpass eine „Corner Frequency“ (= Grenzfrequenz) und es ist interessant zu sehen, wie sich die aktiven Bauteile grundsätzlich voneinander unterscheiden. ***Das untenstehende Bild*** gibt hier einen guten Eindruck und die zugehörige Quelle (Mohr Associates) sollte man sich unbedingt aus dem Internet holen, denn sie stellt eine sehr präzise, aber kompakte und gleichzeitig gut verständliche Einführung dar.



Zusätzlich arbeitete man viele Jahre lang mit Gasentladungsröhren als Rauschquellen, die ein sehr breitbandiges „Plasma-Noise“ erzeugten.

21.4. Weisses Rauschen zur Messung einer Übertragungs-Funktion

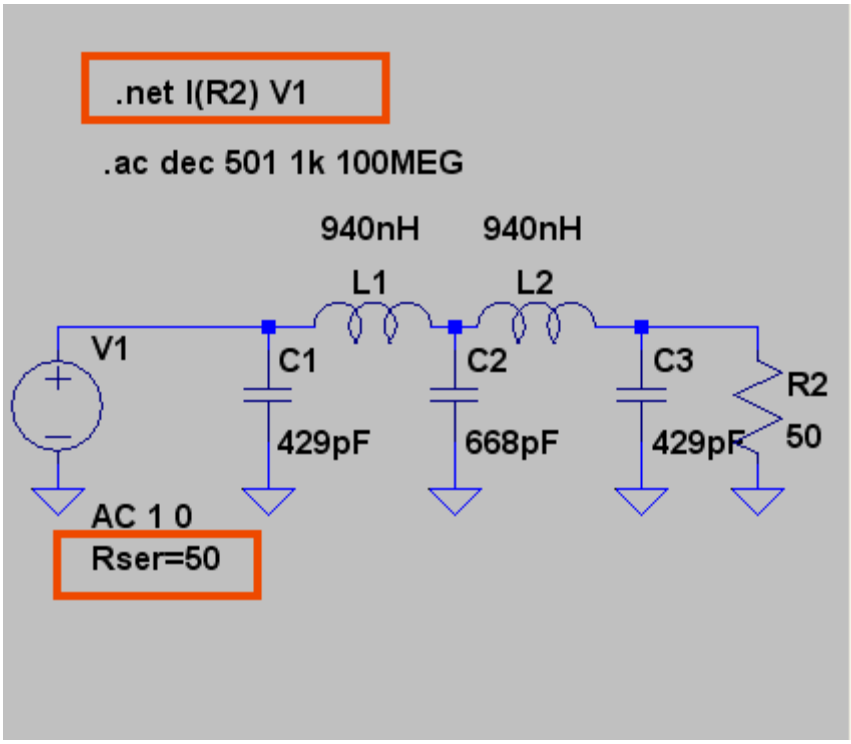
Dieses Problem stellt sich pausenlos in der Übertragungstechnik: es werden Filter, Verstärker, Weichen, Koppler usw. benötigt, entwickelt und produziert. Da möchte man doch stets wissen, was hinten herauskommt, wenn man vorne etwas hineinschickt.

Nehmen wir uns **als Beispiel einen Tiefpass mit der Grenzfrequenz von 11 MHz vor** (…denn den benötigt man, um Signale oberhalb der oft bei Radio-Empfängern benützten „Zwischenfrequenz“ von 10,7 MHz zu unterdrücken).
Bei allen diesen Bausteinen der Übertragungstechnik arbeitet man am Eingang und Ausgang mit demselben Systemwiderstand (meist: 50Ω) und beschreibt die Eigenschaften durch die „S-Parameter“ (= Streuparameter = scattering parameters). Man versucht also stets, den Idealfall der Leistungsanpassung zu erreichen und die S-Parameter zeigen, wie gut das glückt.
Das Prinzip ist ganz einfach:

Man schickt auf einem 50Ω-Kabel das Ausgangssignal eines Generators mit 50Ω Innenwiderstand zum Prüfling. Es hat die Amplitude „Halbe Urspannung“ (wegen Ri = Ra) und löst bei der Ankunft am Prüfling zwei Reaktionen aus:

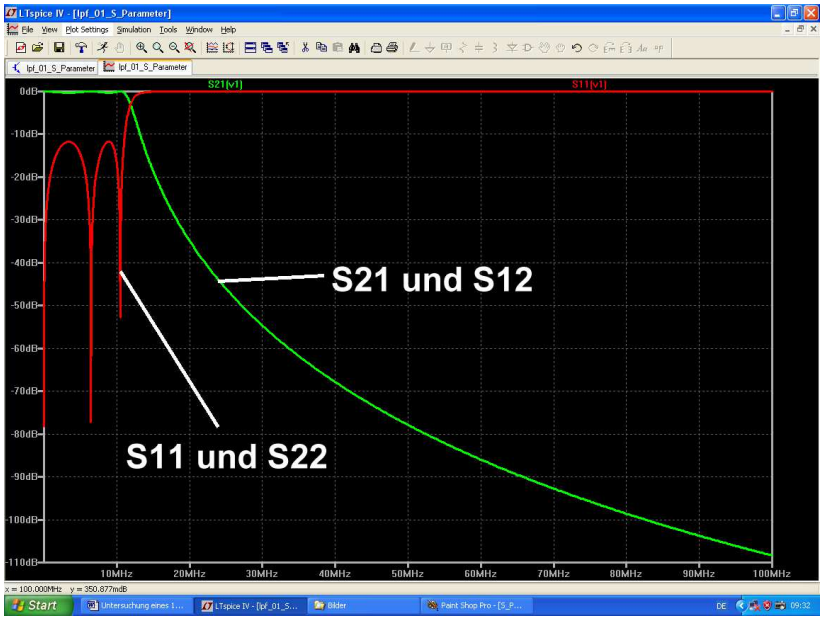
> Weicht der Eingangswiderstand von 50Ω ab, dann gibt es ein Echo und diese „nicht abgebbare Energie“ läuft auf dem Kabel zum Generator zurück (= **Input Reflection**. Sie wird durch **S11** ausgedrückt).
> Der Rest der ankommenden Energie tritt in den Baustein ein und erzeugt dort ein Ausgangssignal, das größer oder kleiner sein kann. **Diese Reaktion beschreibt die „Forward Transmission S21“ und sie ist die übliche Methode zur Angabe einer Übertragungsfunktion in der Nachrichtentechnik.**

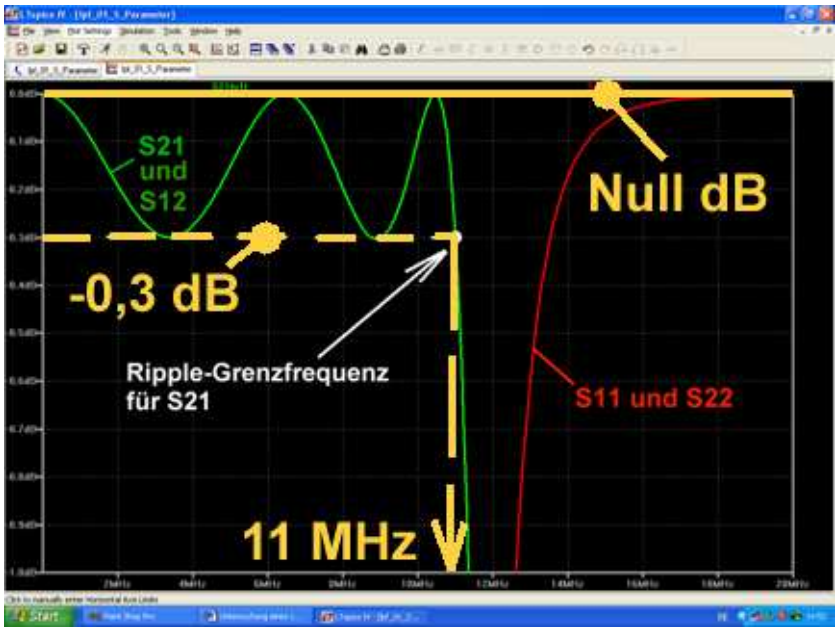
Das Ganze wird dann auf der Ausgangsseite wiederholt und liefert die beiden restlichen Parameter S22 und S12.



So sieht die Tiefpass-Schaltung in LTspice aus, wie sie für die S-Parameter-Simulation vorbereitet werden muss.

Das ist das Simulationsergebnis:
Alle S-Parameter in „Gesamt-Darstellung“….





…und hier nochmals speziell die „Forward Transmission S21“, aber zur Demonstration der Tschebyschef-Wellen entsprechend gedehnt.

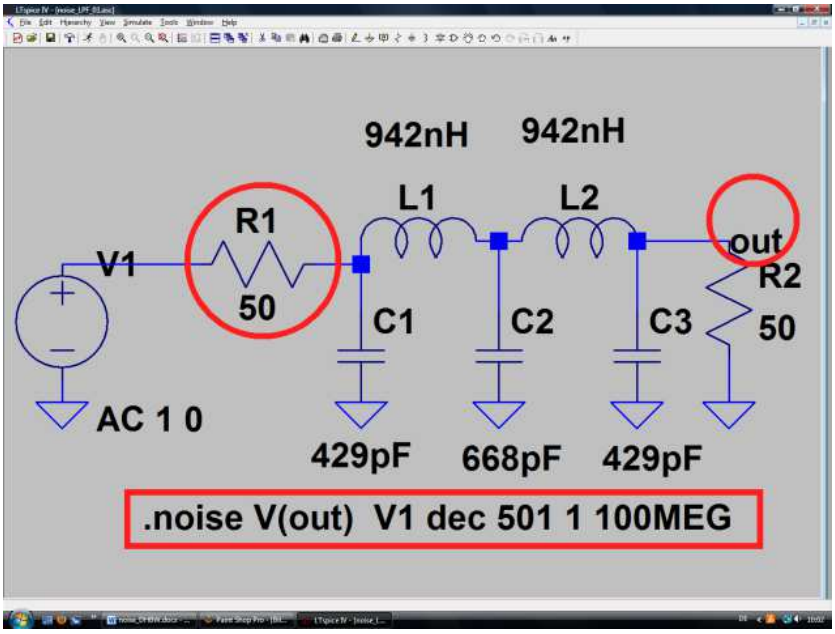
Die Bestimmung der Forward Transmission kann (sowohl in Simulation wie auch in der Praxis) auf drei unterschiedliche Weisen erfolgen:

a) Mit der **S-Parameter-Simulation bzw. mit dem Vektoriellen Network Analyzer**, bei dem die Frequenz des speisenden Generators von der tiefsten bis zur höchsten Frequenz langsam **„gesweept“** wird. Das entspricht dem Beispiel auf der vorigen Seite.

b) Mit einem „**Dirac-Impuls**“ am Eingang, der bekanntlich kein Linienspektrum besitzt, sondern den vollständigen Frequenzbereich lückenlos mit Energie belegt (…dabei ist die Spektrale Leistungsdichte über der Frequenz konstant). Die praktische Umsetzung ist allerdings sehr aufwendig. Nicht nur wegen der hohen Anforderungen an die nötigen speziellen Messgeneratoren und den Messaufbau, sondern auch wegen der Messung und Auswertung der „Impulsantwort“ in der Time Domain. Aus ihr muss nämlich anschließend über die FFT der Frequenzgang der Übertragungsfunktion berechnet werden.

c) Da **Weißes Rauschen** (ähnlich wie der Dirac-Impuls) ebenfalls ein Spektrum mit konstanter Leistungsdichte aufweist, bietet sich diese Methode an, wenn der Eingang einer Schaltung durch eine zu hohe Dirac-Impuls-Amplitude gefährdet ist. Am Ausgang des Messobjektes muss dann mit einem Spektrum-Analyzer der Verlauf der Spektralen Leistungsdichte des Rauschsignal-Outputs im vorgesehenen Frequenzbereich gemessen werden.

Wir wollen uns nun mit der Simulation von Methode „c)“ etwas ausführlicher beschäftigen. Wir verwenden dazu wieder dieselbe Simulationsschaltung und wandeln sie nur etwas ab:



Am Ausgang bringen wir einen Label „out“ an. Der Innenwiderstand der Spannungsquelle mit 50Ω wird aus dem Schaltzeichen herausgenommen und als diskretes Bauteil in die Schaltung eingebaut.

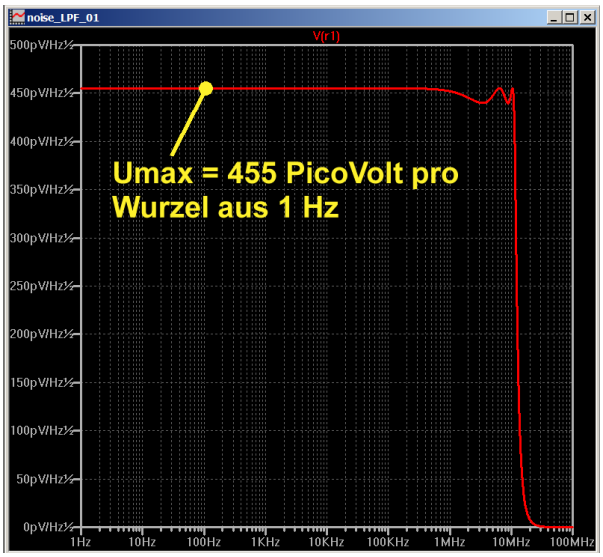
Für die Rauschsimulation brauchen wir folgendes SPICE-Kommando:

.noise V(out) V1 dec 501 1 100MEG

Es ist nicht schwer zu verstehen und liest sich so:

Simuliere die Rauschspannung am Punkt „out“ und benutze dabei die Quelle V1 mit ihren Innenwiderstand R1 als Ausgangspunkt des Rauschens. Führe einen dekadischen Sweep mit 501 Punkten pro Dekade von 1 Hertz bis 100 MHz durch.

Bei der Ergebnisausgabe müssen wir allerdings sehr vorsichtig sein, denn SPICE betrachtet ja den Widerstand R2 ebenfalls als Rauschquelle, berechnet seinen Rauschanteil beim Ausgang und ermittelt schließlich die Ausgangsspannung als „Summe aller Effektivwerte“.



Also dürfen wir NUR auf R1 klicken und damit seinen Rauschanteil beim Ausgang darstellen! Und wenn der die Filterschaltung durchlaufen hat, können wir bei der Ausgangsspannung die „Forward Transmission“ erkennen.

Dieser Ausgangs-Spannungsverlauf mit dem Namen **V(r1)** ist natürlich nichts anderes als die Quadratwurzel aus dem Leistungsdichte-Verlauf pro 1Hz (….daher die „Wurzel pro 1Hz“…). Aber er zeigt sehr schön das Übertragungsverhalten des Filters.

Zur Darstellung von S21 müssen wir uns jedoch etwas Anderes ausdenken.
Es gibt dazu die Möglichkeit, in LTspice selbst geschriebene Funktionen berechnen und anschließend ihr Ergebnis ausplotten zu lassen.

Diese Zusatzfunktionen müssen vorher mit einem Texteditor extra geschrieben und in einer Datei „plot.defs“ als ANSI-Text gespeichert werden. Diese Datei wird vor Beginn im LTspice-Programmordner (mit Administrator-Rechten) gespeichert -- aber bei jeder Änderung von „plot.defs“ muss LTspice erst wieder neu gestartet werden, damit die Änderung wirksam wird.

Die Funktionen haben stets folgende Form:

.func myfunc(Übergabe-Parameter){ Formel zur Verknüpfung der übergebenen Parameter }

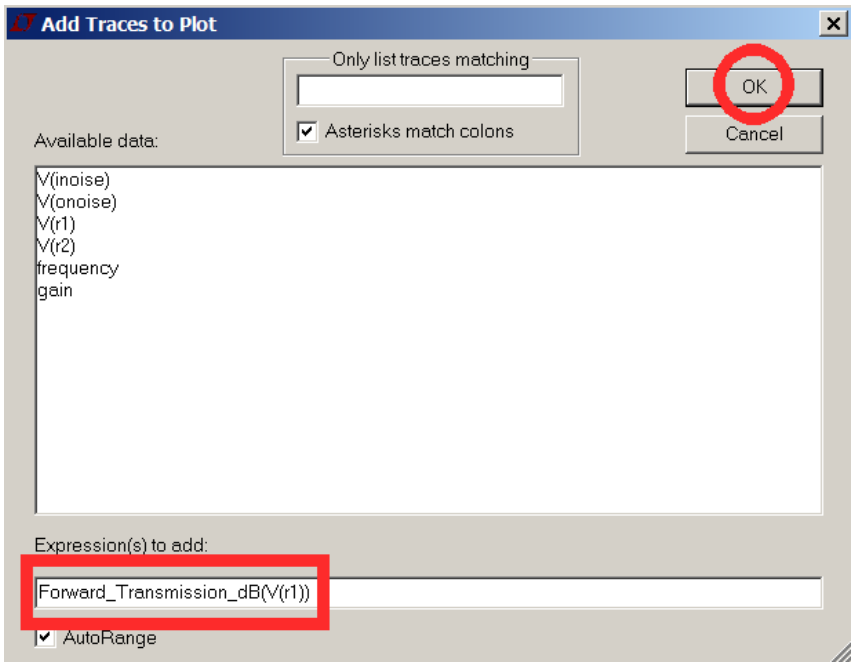
Um das auf unser S21 anzuwenden, brauchen wir zuerst den physikalischen Hintergrund. Da gilt:

$$S21 = 20 * \log_{10}\left(\frac{V(r1)}{V_{incident}}\right) = 20 * \log_{10}\left(\frac{V(r1)}{\sqrt{k * T * R}}\right) = 20 * \log_{10}\left(\frac{V(r1)}{\sqrt{k * 300 * 50}}\right)$$

S21 ist also das Verhältnis von tatsächlich am Ausgangswiderstand (mit 50Ω) ankommender Leistung (= $V(r1)^2$ / 50Ω) zur maximal abgebbaren Rauschleistung = „k*T*B“ „incident power“. Die Bandbreite „B“ wählen wir zu 1Hz und brauchen sie dann nicht mehr in der Formel zu berücksichtigen. Die absolute Temperatur sei 300 Kelvin. Um auf ein Spannungsverhältnis zu kommen, wird die Wurzel gezogen und mit der Zusatzoperation „20 x $\log_{10}$(….)“ erhält man die gesuchten „dB’s“.

Damit haben wir den erforderlichen Inhalt von „plot.defs“, den wir schreiben und vor dem Programmstart speichern müssen:

.func Forward_Transmission_dB(V(r1)) {20*log10(V(r1)/sqrt(k*300*50))}



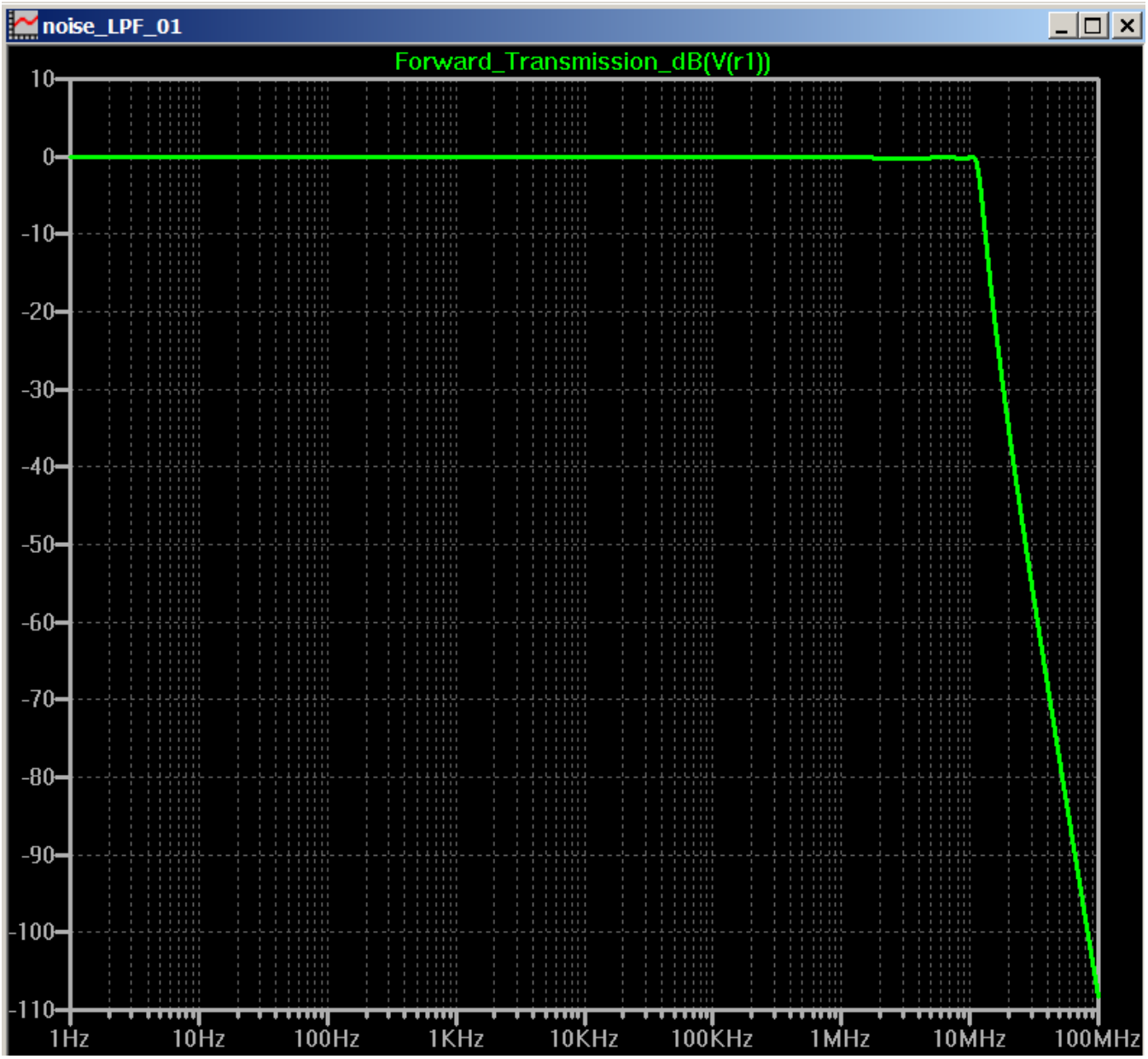
Wird LTspice gestartet und die Schaltung simuliert, dann öffnen wir anschließend eine neue „**Plot Pane**“ und wählen „**Add Trace**“.

Unter „**Expression(s) to add**“ geben wir unsere selbst geschriebene Funktion

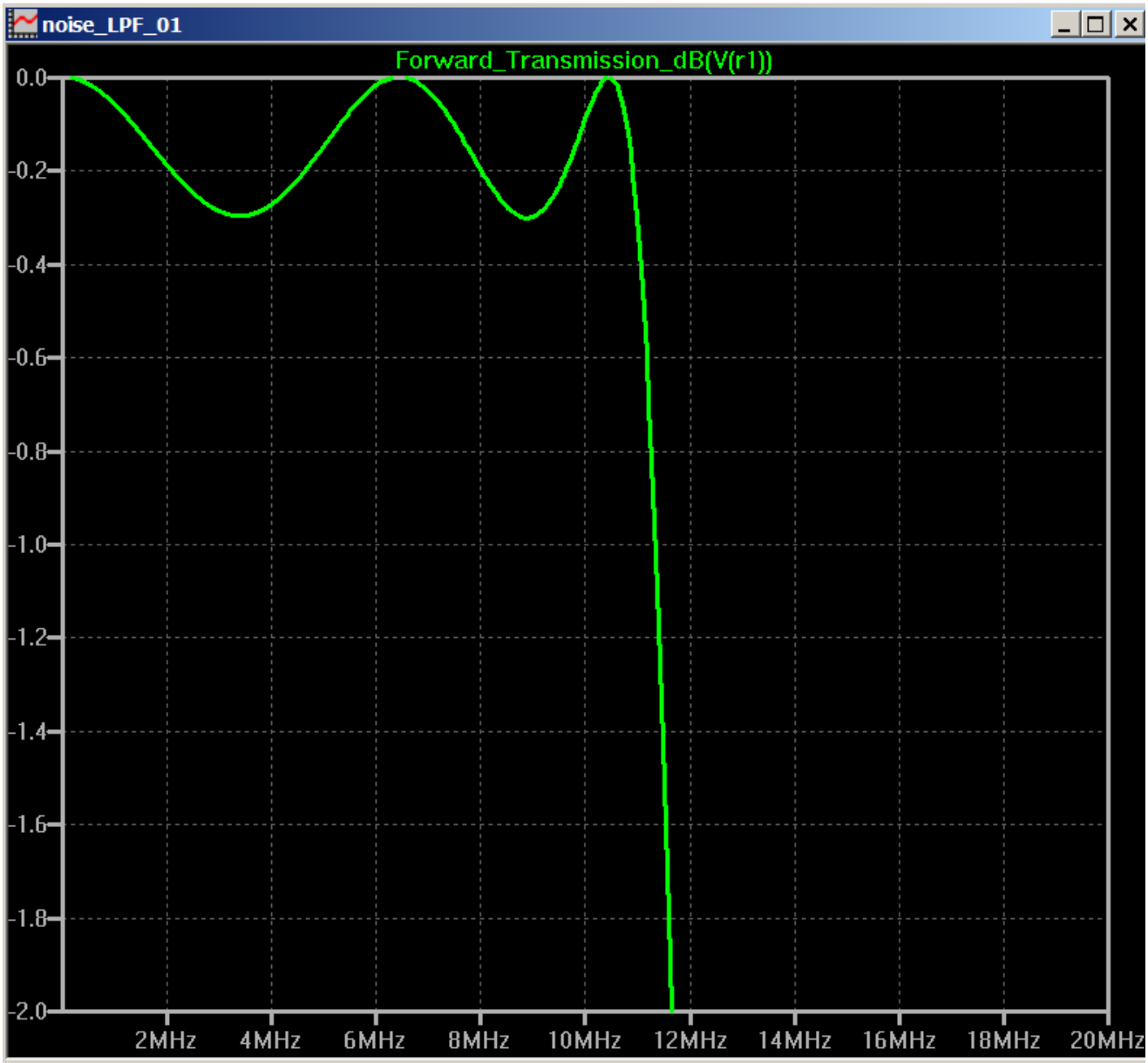
Forward_Transmission_dB(V(r1))

an und drücken OK.

Das Ergebnis ist überzeugend.



Wer sich für die Tschebschef-Wellen im Passband interessiert, braucht nur die Achsenkalibrierung zu ändern:

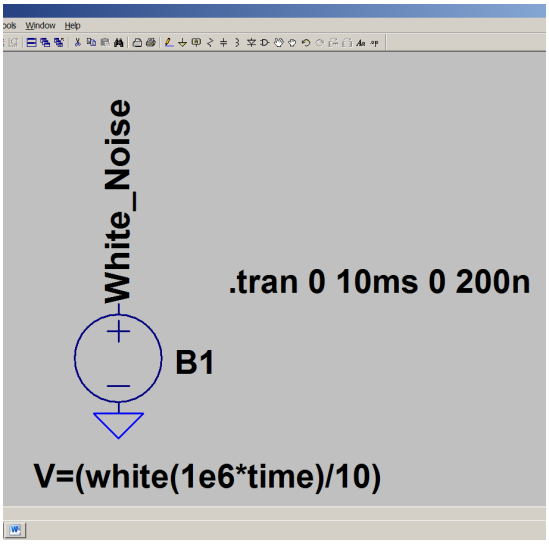


Hinweis:
Bei der praktischen Anwendung muss man den vom **Widerstand R1 erzeugten Rauschpegel sehr stark erhöhen, damit am Ausgang der dazu kommende Anteil des Rauschens von Abschlusswiderstand R2 vernachlässigt werden kann.**

Deshalb wird als Spannungsquelle V1 stets ein Rauschgenerator mit nachfolgender kräftiger Verstärkung eingesetzt. Den Verlauf der Rauschleistungsdichte an R2 (und damit S21) bestimmt man dann mit einem durchstimmbaren Spektrum Analyzer.

21.5. Erzeugung eines Rauschsignals im Zeitbereich mit LTspice

Das interessiert direkt den Schaltungsentwickler, denn so etwas ist oft sehr praktisch. Man kann dadurch z. B. unregelmäßige Schwankungen einer Versorgungsspannung nachbilden, bei AD-Wandlern die Linearität verbessern, die „Bit-Error-Rate“ eines Systems untersuchen usw.



Hierzu stellt LTspice eine Funktion „white(x)“ bereit, die eine Folge von Zufallszahlen im Amplitudenbereich von

-0,5 bis +0,5

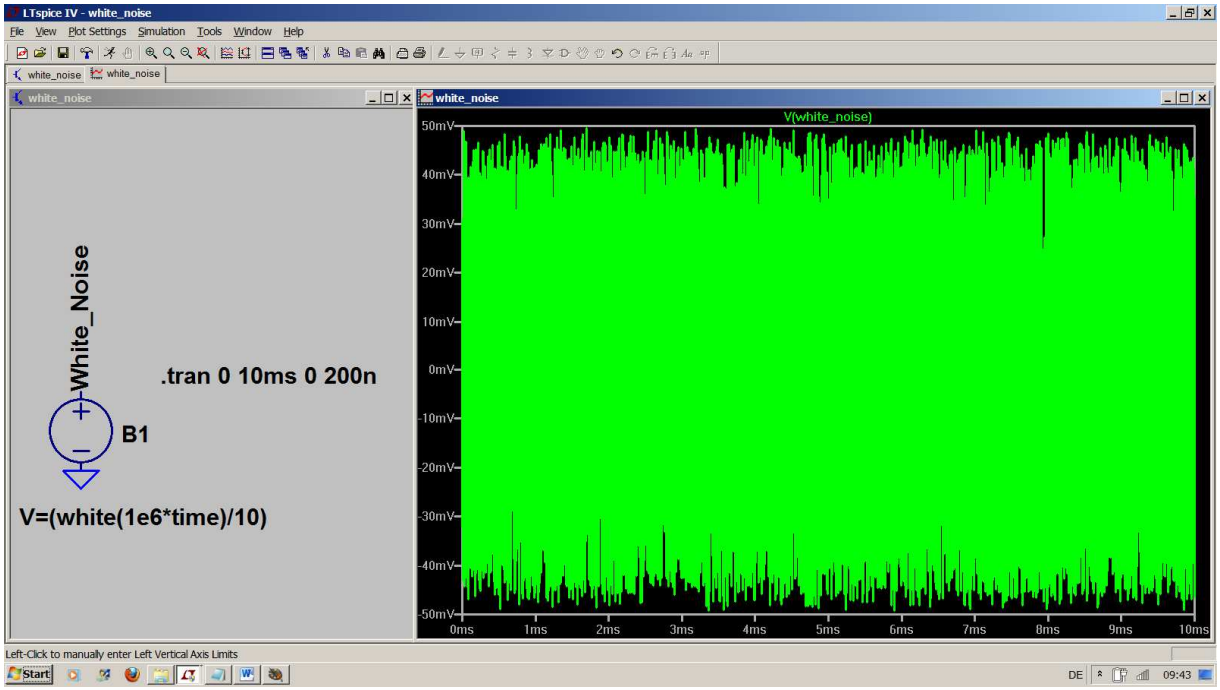
erzeugt und diese Werte durch eine „geglättete Linie“ verbindet (…um extrem scharfe Übergänge = einen Zick-Zack-Verlauf) zwischen den Werten zu vermeiden).
Damit und einer „bv“-Spannungsquelle sind wir schon am Ziel. Allerdings ist noch eine Erläuterung zur Eingabe bei der bv-Quelle nötig.

V=(white(1e6*time)/10)

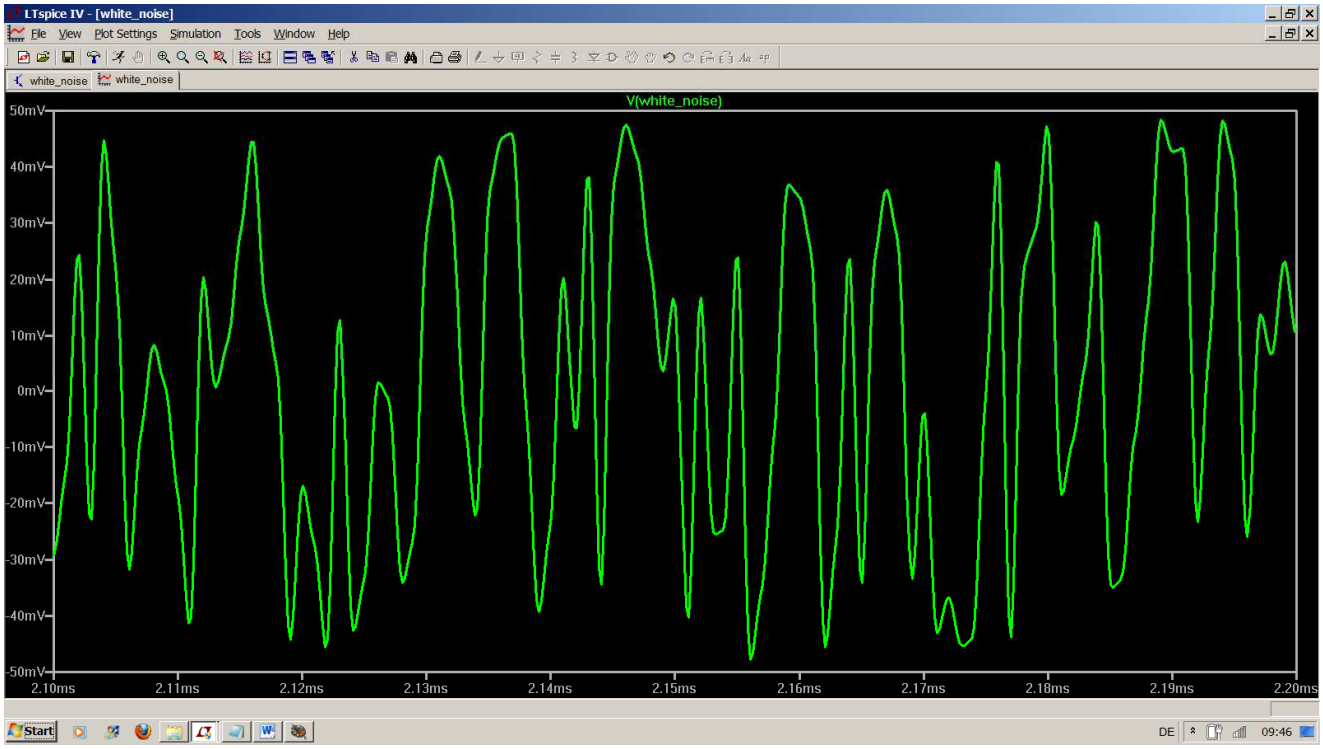
bedeutet:

Wähle eine „Zufalls-Zahlendichte“ von 1e6 = 1 Million Werten pro Sekunde.

Multipliziere diese Zahlendichte mit der vorgesehenen maximalen Simulationszeit von 10 Millisekunden und Du bekommst in diesen 10 Millisekunden eine Folge von
(1 000 000 Werten pro Sekunde) x (10 Millisekunden) = 10 000 Werte
geliefert.
Die Division durch die Zahl 10 vermindert die Maximalamplitude von +-0,5V auf einen praxisgerechteren Wert von +-50 Millivolt und liefert folgendes für die gewählte Sweep-Einstellung folgendes Ergebnis:



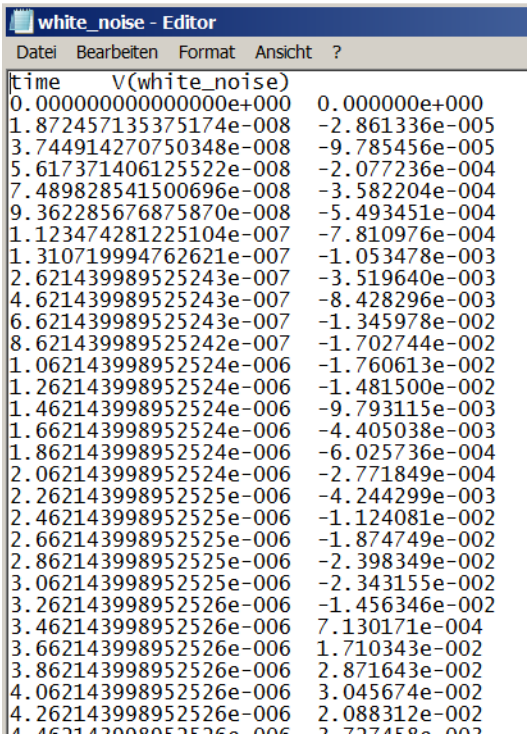
Bitte nachprüfen und dann nur einen Ausschnitt mit 100 Mikrosekunden Breite darstellen!



Sehr gut sind die „weichgespülten“ Übergänge zwischen den von der „white“-Funktion angelieferten Einzelwerten zu erkennen.

Ein Tipp:
Wer für andere Zwecke die **komplette Folge der simulierten Werte als Textfile** benötigt, der gehe so vor:

a) Zuerst klickt man im Menü „File“ auf „Export“
b) Dann wählt man im auftauchenden Menü „V(white_noise)“ aus und klickt auf OK
c) Jetzt findet man das Textfile mit den Daten genau in dem Ordner, in dem auch die simulierte Schaltung gespeichert ist.



white_noise - Editor

Datei Bearbeiten Format Ansicht ?

time	V(white_noise)
0.000000000000000e+000	0.000000e+000
1.872457135375174e-008	-2.861336e-005
3.744914270750348e-008	-9.785456e-005
5.617371406125522e-008	-2.077236e-004
7.489828541500696e-008	-3.582204e-004
9.362285676875870e-008	-5.493451e-004
1.123474281225104e-007	-7.810976e-004
1.310719994762621e-007	-1.053478e-003
2.621439989525243e-007	-3.519640e-003
4.621439989525243e-007	-8.428296e-003
6.621439989525243e-007	-1.345978e-002
8.621439989525242e-007	-1.702744e-002
1.062143998952524e-006	-1.760613e-002
1.262143998952524e-006	-1.481500e-002
1.462143998952524e-006	-9.793115e-003
1.662143998952524e-006	-4.405038e-003
1.862143998952524e-006	-6.025736e-004
2.062143998952524e-006	-2.771849e-004
2.262143998952525e-006	-4.244299e-003
2.462143998952525e-006	-1.124081e-002
2.662143998952525e-006	-1.874749e-002
2.862143998952525e-006	-2.398349e-002
3.062143998952525e-006	-2.343155e-002
3.262143998952526e-006	-1.456346e-002
3.462143998952526e-006	7.130171e-004
3.662143998952526e-006	1.710343e-002
3.862143998952526e-006	2.871643e-002
4.062143998952526e-006	3.045674e-002
4.262143998952526e-006	2.088312e-002

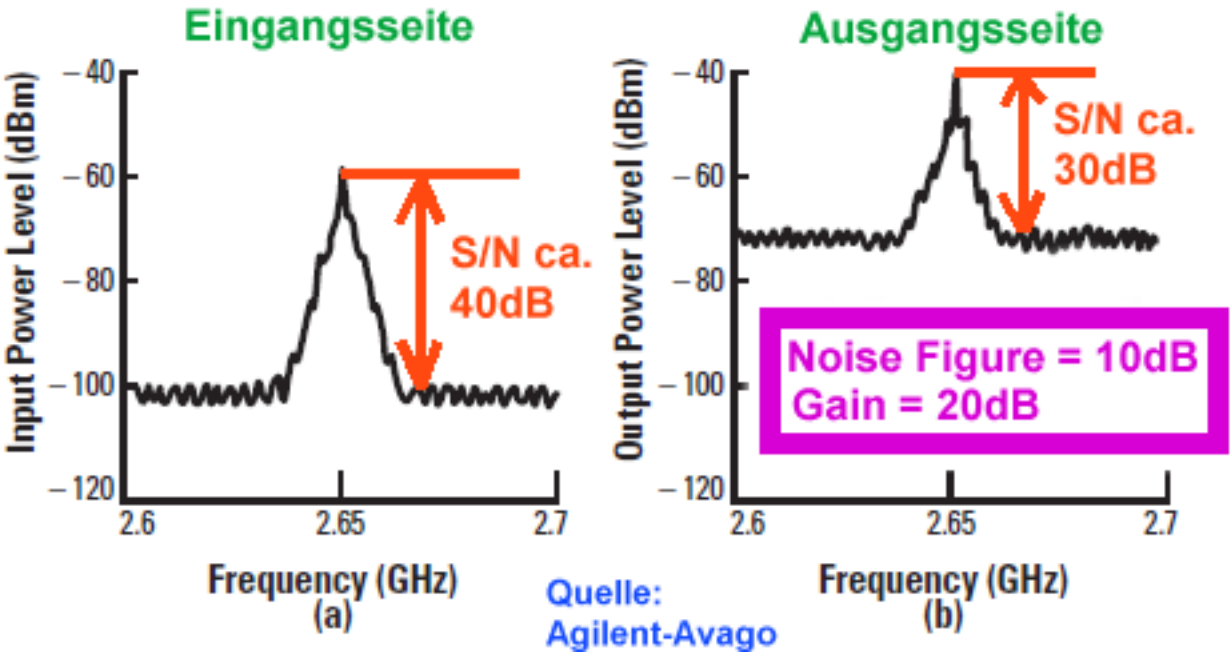
Beispiel für obige Simulation:

Übrigens:
Bei der praktischen Anwendung in einer Schaltung kann man nun diese bv-Quelle z. B. einer Betriebsspannung oder einem ansteuernden Eingangssignal in Reihe schalten. Natürlich darf man dann nicht vergessen, auch einen Innenwiderstand bei dieser Quelle vorzusehen…
==========

21.6. Die „Noise Figure NF in dB“ bei Verstärkern

Da ist zuerst eine kleine Erläuterung erforderlich:
Nahezu alle Bausteine der Kommunikationstechnik verschlechtern den Signal-Rauschabstand von Signalen, die ihnen zur Verarbeitung zugeführt werden, durch ihr Eigenrauschen. **Dieser Unterschied zwischen den Signal-Rauschabständen beim Eingang und Ausgang in dB (= also die Verschlechterung…)** **wird als „Noise Figure“ bezeichnet und ebenfalls in dB angegeben.**

Beispiel für einen Verstärker mit NF = 10dB und einer Verstärkung von 20dB:



Wie bestimmt nun LTspice diese „Noise Figure“?
Nun, dazu werden in bekannter Weise nacheinander alle Widerstände oder Halbleiterbauteile der Schaltung als Rauschquellen betrachtet und ihr Anteil am Ausgangsrauschen bestimmt. Teilt man am Ende diese Ausgangsrauschspannung „V(onoise)“ durch die Verstärkung des Bausteines, dann erhält man die

„in Reihe zur ansteuernden Signalquelle zu denkende Eingangs-Rauschspannung V(inoise)“.

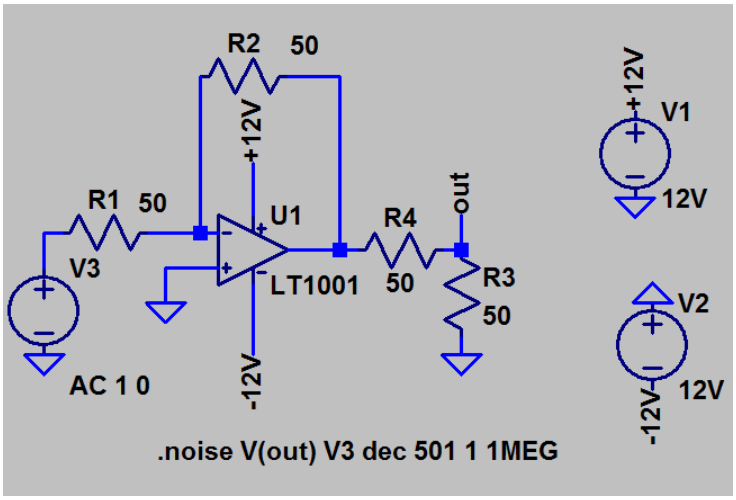
Die zu ihr gehörende theoretische Rauschleistung wird anschließend mit derjenigen verglichen, die im **Innenwiderstand allein entsteht (= 4 * k * T * B)** und das Ergebnis in dB ausgedrückt. Das muss dann die „**Noise Figure in dB**“ sein. Also müssen wir dazu wieder mal eine neue Funktion schreiben. Die nötige Beziehung lautet:

$$\text{NF in dB} = 10 * \log_{10}(V(inoise) * V(inoise) / R_{source} * 4 * k * T * B)$$

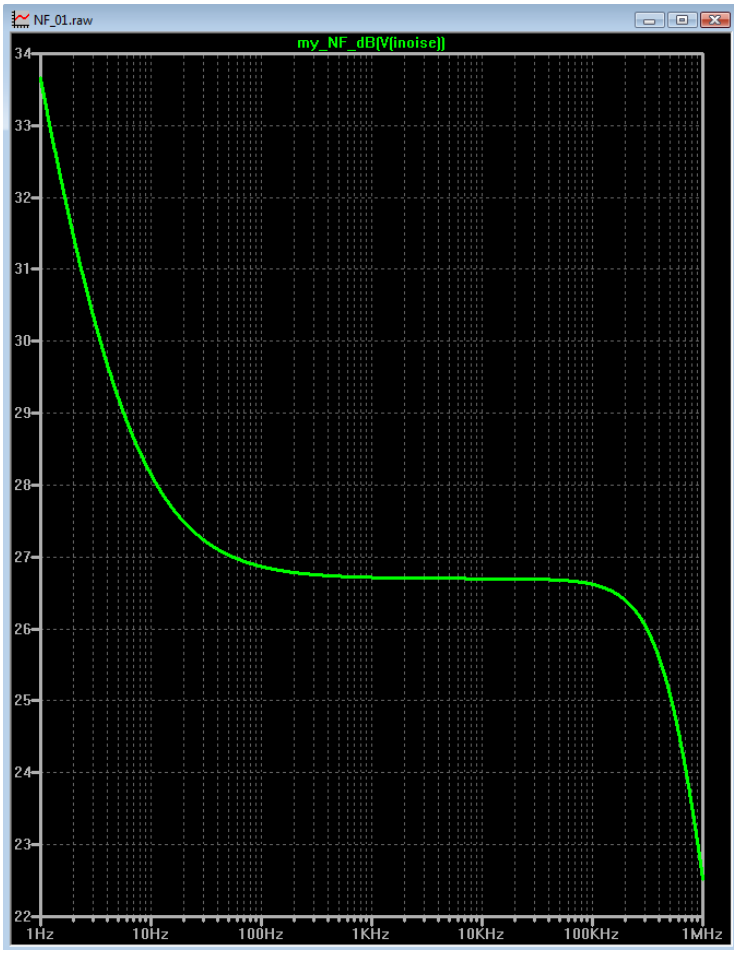
Für LTspice müssen wir das allerdings so formulieren (und eine Temperatur von 300k sowie einen Innenwiderstand von 50Ω und eine Bandbreite von 1Hz verwenden):

```
.func   my_NF_dB(V(inoise))
{ 10*log10(V(inoise)*V(inoise)/(50*4*k*300)) }
```

Diese Anweisung wird im LTspice-Programmordner als “plot.defs” gespeichert und erst hinterher das Programm sowie die Simulation gestartet.
Als Anwendungsbeispiel nehmen wir den einfachen „Wald- und Wiesen – OPV“ vom Typ „LT1001“ aus dem Bauteilvorrat und erstellen damit eine umkehrende Verstärkerschaltung mit einem Eingangswiderstand von 50Ω, einem Innenwiderstand und Lastwiderstand von je 50Ω und +-12V als Versorgung.



Das Label „out“ am Lastwiderstand dürfen wir natürlich nicht vergessen!



Nach der Simulation öffnen wir eine neue „**Plot Pane**“, gefolgt von „**Add Trace**“. Wenn wir darin

my_NF_dB

angeben, sollte dieses Bild erscheinen.

Das wär’s…

21.7. Die Bestimmung der „Noise Figure NF in dB“ in der Praxis

Dieses Kapitel beginnt mit der Empfehlung, sich folgende zwei Application Notes der Firma Agilent Avago aus dem Internet zu holen. Darin finden sich nicht nur die erforderlichen Rausch-Grundlagen in gut verständlicher Form, sondern auch eine ausführliche Beschreibung der „**Y-Methode**“ -- denn sie ist das häufigste Messverfahren bei Rausch-Messplätzen.

Application Note 57-1:

Fundamentals of RF and Microwave Noise Figure Measurements

Application Note 57-2:

Noise Figure Measurement Accuracy – The Y-Factor Method

Die **Y- Methode** geht so vor:

A) Man verwendet meist eine **Rauschdiode im Avalanche-Betrieb**, die durch ein **Rechteck-Signal als Versorgungsspannung periodisch ein- und ausgeschaltet wird.** Sie speist den Eingang des Prüflings.

B) Im **ausgeschalteten** Zustand gibt sie an den Eingang nur die zu ihrem **50Ω-Innenwiderstand gehörende Rauschleistung ab**. Die wird am Ausgang des Prüflings gemessen.

C) Im **eingeschalteten** Zustand produziert sie dagegen eine starke Rauschleistung, die durch die Angabe des „**Excess Noise Ratio ENR“** charakterisiert wird. Letztlich ist das der Wert, um den nun die abgegebene Rauschleistung höher ist als das Eigenrauschen des Innenwiderstandes im „kalten“ Zustand (…in der Praxis liegt die möglichen und erhältlichen ENR-Werte zwischen 4 dB und 16 dB). Wieder wird die am Ausgang des Prüflings auftretende Rauschleistung gemessen.

D) **Das Verhältnis der beiden gemessenen Rauschleistungen wird als „Y – Faktor“ berechnet und weiterverarbeitet.** Das geht so:

$$\textbf{Rauschfaktor} = \frac{\textbf{ENR}}{\textbf{Y}-\textbf{1}}$$

Und die **Noise Figure in dB** ist hinterher ein Kinderspiel

NF in dB = 10 x log(Rauschfaktor)

Natürlich wartet da manche Tücke auf den Anwender und man muss Etliches beachten. Aber das kann man speziell in der zweiten Application Note entspannt nachlesen.

Noch zur Information:
So sieht die professionelle Rauschquelle mit der Avalanche-Diode in der Praxis aus.

22. Sechzehntes Projekt: Double Balanced Mixer, Teil 1= Ringmodulator

22.1. Etwas Grundlagen

Mit dieser bekannten Schaltung aus der Kommunikationstechnik kann man die Frequenzen von Signalen verändern, ohne dass die 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, die beide zugeführten Eingangssignale miteinander multipliziert.
Und dafür hat die Mathematik eine Formel bereit, denn es gilt:

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

Wenn man nun 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 die Summenfrequenz und die Differenzfrequenz der beiden Eingangsspannungen!

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

Das Rechtecksignal besteht bekanntlich selbst wieder aus einer Summe von sinusförmigen Signalen. Neben der „Grundwelle“ findet man die „Oberwellen“, und die haben 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 multiplizert 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.

22.2. Standardschaltung des Ringmodulators

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.

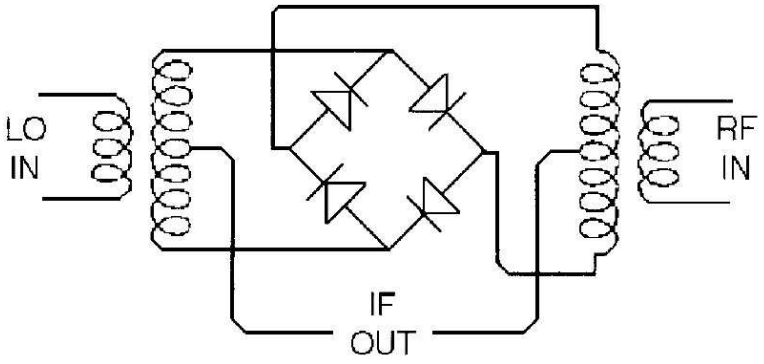


Figure 2. A popular diode-ring mixer topology.

Sie besteht aus einem Ring von 4 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.

Ein Tipp: für hochfrequente Anwendungen dürfen nur extrem schnelle Schottky-Dioden eingesetzt werden.

„**LO-IN**“ bedeutet „Local Oscillator“-Input

„**RF-IN**“ heißt „Radio Frequency“-Input und entspricht dem **Informationseingang.**

„**IF-OUT**“ liefert das Modulationsprodukt und steht für „Intermediate Frequency Output“ (in Deutsch: Zwischenfrequenz-Ausgang)

Arbeitsweise:**Durch das (große) LO-Signal werden bei dessen positiver Halbwelle die beiden rechten, in Reihe liegenden Dioden, bei der negativen Halbwelle dagegen die beiden linken, in Reihe liegenden Dioden als elektronische Schalter betrieben und durchgeschaltet. Dadurch wird das von rechts 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 „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 oder mehr) 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 mehr viel.

Ist im Einsatz völlig laiensicher und kann tatsächlich wie ein Bauklötzchen benützt werden, da er meist schon mit drei passenden HF-Buchsen (BNC- oder N- oder SMA-Steckverbinder) 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 und gründlich ein- und auszuschalten. Es ist keine Verstärkung möglich, deshalb besitzt der Baustein grundsätzlich eine **Dämpfung** (typisch: zwischen 5 und 8dB) und verschlechtert deshalb auch das Rauschverhalten eines Systems (Konsequenz: mehr „Vorverstärkung“ nötig).

22.3. Ein Problem: die erforderlichen Übertrager

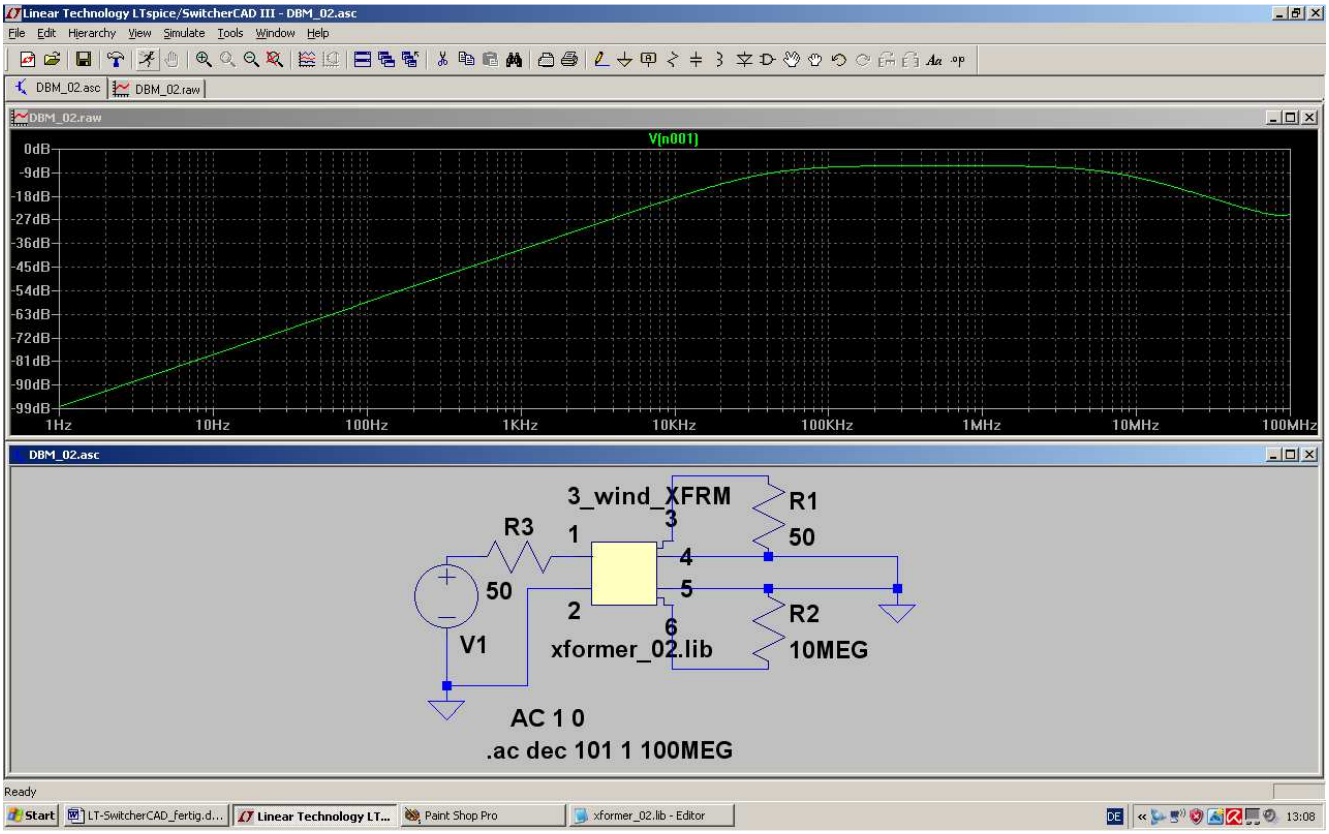
Wir wollen die Schaltung mit idealen Übertragern simulieren, aber ein Blick auf die Übertrager-Probleme bei einer echten DBM-Schaltung ist wichtig. Jeder Trafo weist nämlich prinzipiell Kapazitäten und Induktivitäten auf, die im Ersatzschaltbild berücksichtigt werden müssen. Das schränkt die nutzbare Bandbreite deutlich ein und deshalb muss man vorher den gewünschten Frequenzbereich vorgeben, in dem der DBM arbeiten soll.

Beispiel:
Würden wir mit einem LO-Signal von 1MHz und mit einem RF-Signal von 100kHz arbeiten, dann sollte die **obere Grenzfrequenz der Übertrager mindestens bei 10….20 MHz und die untere Grenzfrequenz deutlich unterhalb von 100kHz** liegen.

Also nehmen wir unseren altbekannten „xformer_02“ mit 3 identischen Wicklungen und verändern solange seine Daten, bis diese Bedingung erfüllt ist. Allerdings muss man beachten, dass immer **nur EINE Sekundärwicklung** in Betrieb ist (und die andere leer läuft…deshalb muss diese zweite Wicklung in der Simulation mit einem Widerstand abgeschlossen werden. Hier sind es 10MΩ.).

Das führt zu folgender Simulation der Spannung an R1, bei der

jede einzelne Wicklung eine Induktivität von 100µH aufweist (bei einem Wicklungswiderstand von 1Ω und einer Wicklungskapazität von 1pF).



Sehr schön ist nun der eingeschränkte Übertragungsbereich zu erkennen.

22.4. Simulation des DBM-Verhaltens mit idealen Übertragern

Links wird das LO-Signal als Schaltspannung für die Schottky-Dioden angelegt (Spitzenwert = 2V, Frequenz = 1MHz). Von rechts kommt die umzusetzende Information in Form des RF-Signals mit dem Spitzenwert von 20mV und der Frequenz f = 100kHz.

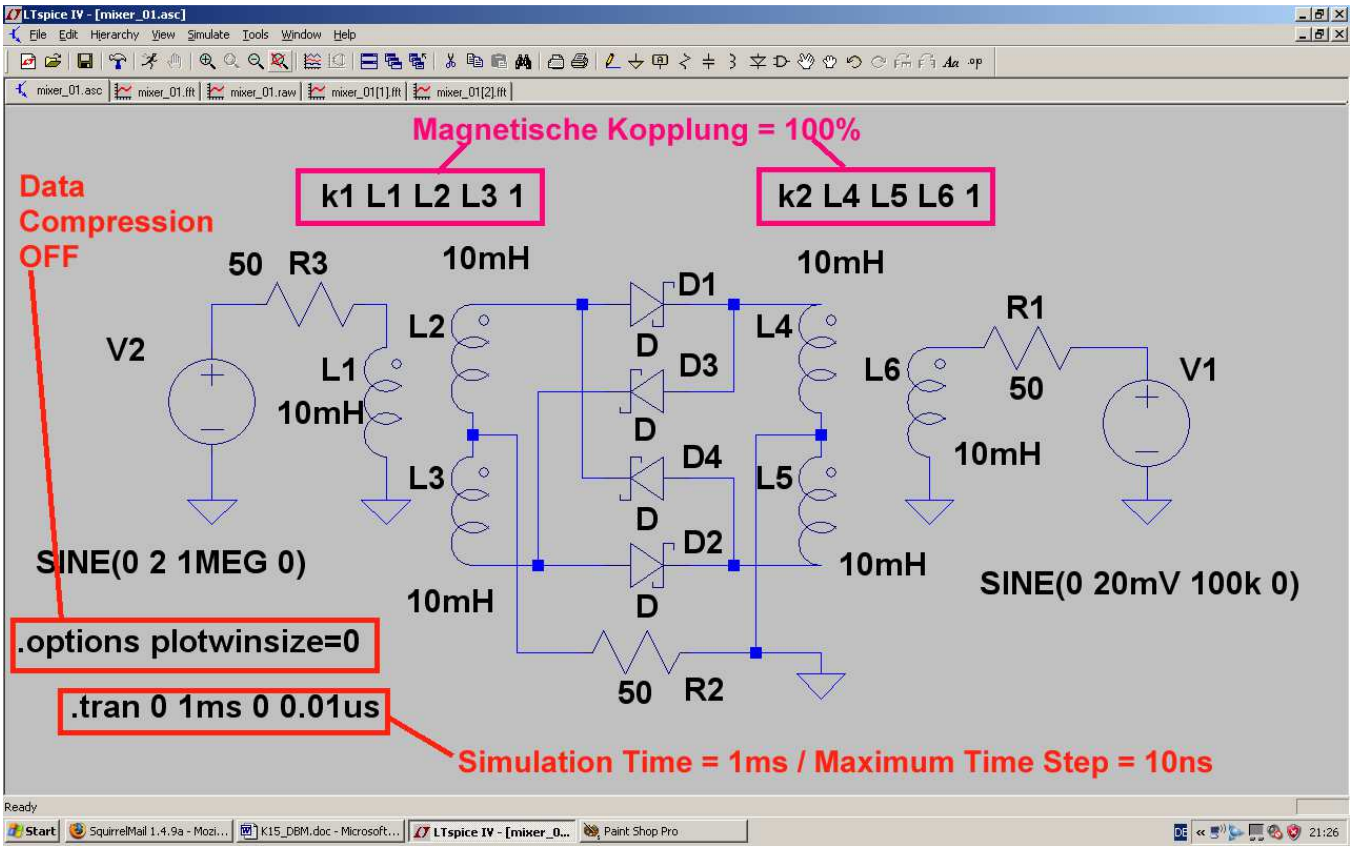
Beide idealen Übertrager bestehen (Siehe Kapitel 6.2) aus jeweils drei Einzelinduktivitäten vom Typ „ind2“, die zu 100% magnetisch gekoppelt sind. Jede Wicklung weist eine Induktivität von 10mH auf.

Die Schottky-Dioden tragen die Bezeichnung „schottky“ und finden sich in der LTSpice-Bauteilbibliothek.

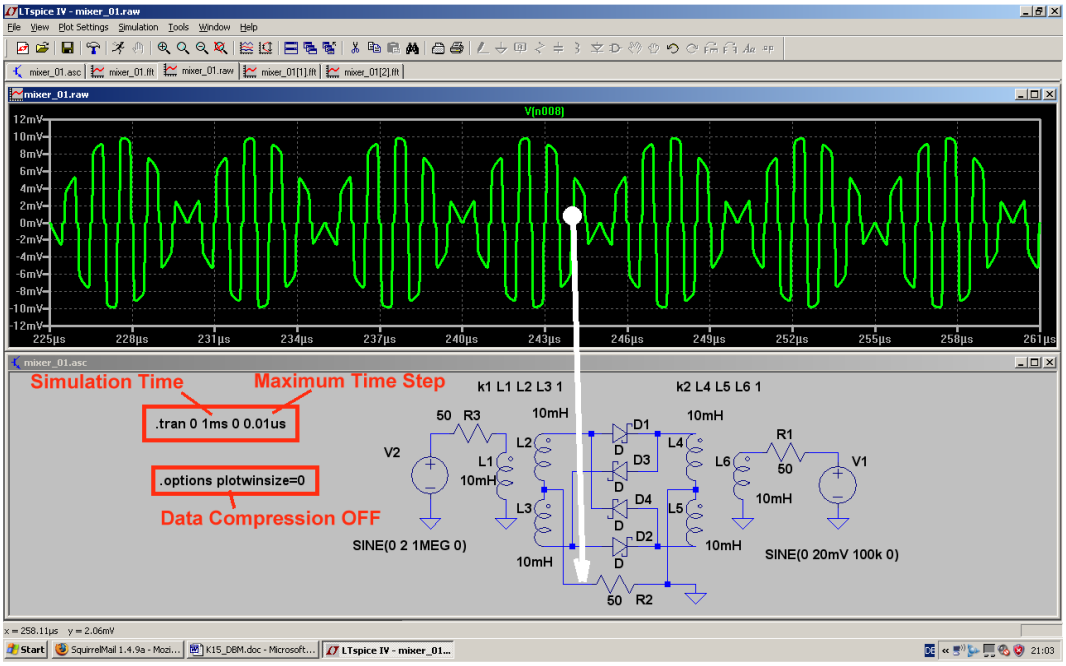
Eine Simulationszeit von 1ms ergibt eine Frequenzauflösung von 1/1ms = 1000Hz. Fordert man mindestens
100 000 echte Samples, dann muss ein Maximum Time Step von 1ms/100000 = 10ns gewählt werden. Dies gehört zu einer Samplefrequenz von 1/10ns = 100MHz und so braucht man bis zu einer Signalfrequenz von 50MHz keine Aliasing-Effekte zu befürchten.

Die Datenkompression wird durch die Direktive **„.option plotwinsize=0“** außer Betrieb gesetzt.

Damit sollte vor dem Simulationsstart folgender Editor-Bildschirm zu sehen sein:

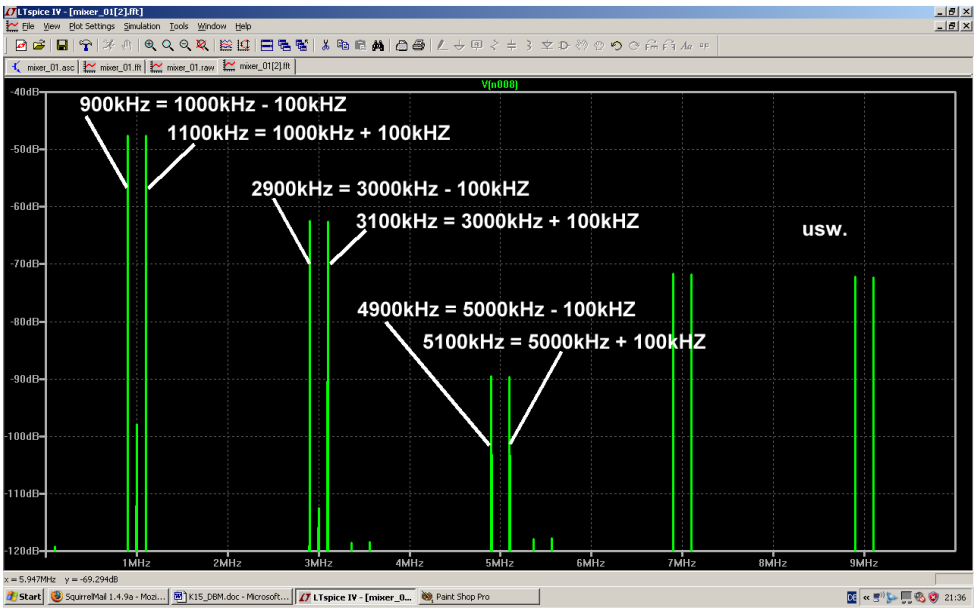


Und so sieht das aus, wenn man simuliert und anschließend die Spannung am Ausgangswiderstand R2 etwas zoomt:



Man erkennt sehr gut die dauernde Umpolung des RF-Signals durch das LO-Signal. Ebenso sieht man, dass wegen der Schwellspannung der Schottky-Dioden (= ca. 0,4v) bei jedem Nulldurchgang ein kurzer „Aussetzer“ zu beobachten ist (= unterhalb von einer LO-Signalamplitude von etwa 2 x 0,25V sperren die Dioden bereits und dann fehlt kurz ein Stück vom RF-Signal am Ausgang).
Interessant ist nun natürlich das Frequenzspektrum mit seinen unterschiedlichen Frequenzanteilen.
Also klicken wir „rechts“ auf das Diagramm, wählen „FFT“ und darin **65 536 Samples** für die Spannung an R1.
Nach OK stellen wir das Ergebnisdiagramm auf folgende Skalierungen um:

Frequenzachse mit linearer Teilung / Frequenzbereich von 0Hz bis 10 MHz / Tick = 1MHz
Senkrechte Achse mit dB-Teilung. Dargestellter Bereich = -40dB……-120dB mit einem Tick von 10dB.



Und da haben wir genau das, was die Theorie vorhergesagt hat: Bei jeder ungeraden Oberwelle (= „Vielfache“ oder „Harmonische“ des LO-Signals, also 1 / 3 / 5 / 7 / 9…..MHz) taucht ein Signalpärchen auf, das selbst wieder als Summen- und Differenzfrequenz von RF-Signal und betreffender Oberwelle entstanden ist.

Sowohl das LO-Signal wie auch das RF-Signal sollten (theoretisch…) nicht mehr mit ihren „Originalfrequenzwerten“ vorhanden sein…

23. Siebzehntes Projekt: Double Balanced Mixer, Teil 2 = Active Mixer = Gilbert Cell

23.1. Einführung

Die **passiven Double Balanced Mixer** (Siehe Teil 1) weisen als geschaltete Mischer hervorragende Eigenschaften bezüglich Linearität und Breitbandigkeit auf. Allerdings haben sie folgende Eigenheiten:

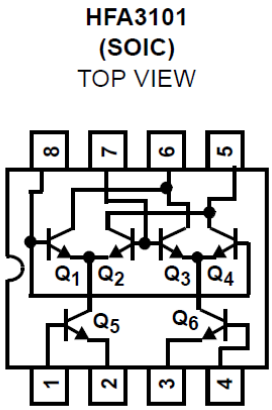
a) Sie verlangen sehr **korrekte Abschlüsse mit dem Systemwiderstand „Z“** (meist 50 Ω) an allen drei Ports, wenn die erwähnten guten Eigenschaften erhalten bleiben sollen.

b) Sie weisen prinzipiell eine „**Umsetz-Dämpfung**“ auf, die in der Praxis **zwischen 5 und 8 dB** liegt (deshalb hat auch die SSB-Rauschzahl NF denselben Wert).

c) Die Linearität steigt direkt mit dem angelegten LO-Pegel, denn dieses Signal schaltet die verwendeten Schottky-Dioden. Dieser Pegel muss von der zugehörigen LO-Signalquelle aufgebracht werden. **Hochlineare Mischer benötigen bis zu 1 Watt Schaltleistung, um die geforderten Daten zu erreichen**.

Deshalb hat man, als die ersten HF-Transistoren verfügbar waren (= ca. 1960), damit eine **aktive Schaltung mit dem Namen „Gilbert Cell“** entwickelt. Sie ist heute das Herz von vielen integrierten Up- oder Downconvertern oder Frequenzumsetzern oder Demodulatoren . Natürlich werden jetzt auch FETs, MOSFETs oder (für sehr hohe Frequenzen) HEMTs eingesetzt, aber das Prinzip (Siehe dieses Bildchen) ist unverändert geblieben.

Nachteil: Betriebsspannung nötig.



Vorteile:

a) Bereits in der Grundschaltung ist ein **wesentlich kleinerer LO-Pegel** als beim passiven DBM ausreichend. Oft wird zusätzlich ein LO-Verstärker auf dem Chip angeordnet, dann reichen Pegel kleiner als 100mV für den erfolgreichen linearen Betrieb aus.

b) Durch die Verwendung aktiver Bauteile lässt sich bei der Schaltung sogar eine **Verstärkung** (statt einer Dämpfung wie beim passiven DBM) erzielen.

c) Transistoren oder FETS weisen **höhere Eingangswiderstände** auf. Das ist für viele Anwendungen von Vorteil, aber auch ein vom Entwickler auf 50Ω gebrachter Eingang ist dann wesentlich reflexionsärmer und breitbandiger. Damit ist der Ruf des passiven DBMs nach breitbandiger korrekter Anpassung an allen Ports nicht mehr zwingend.

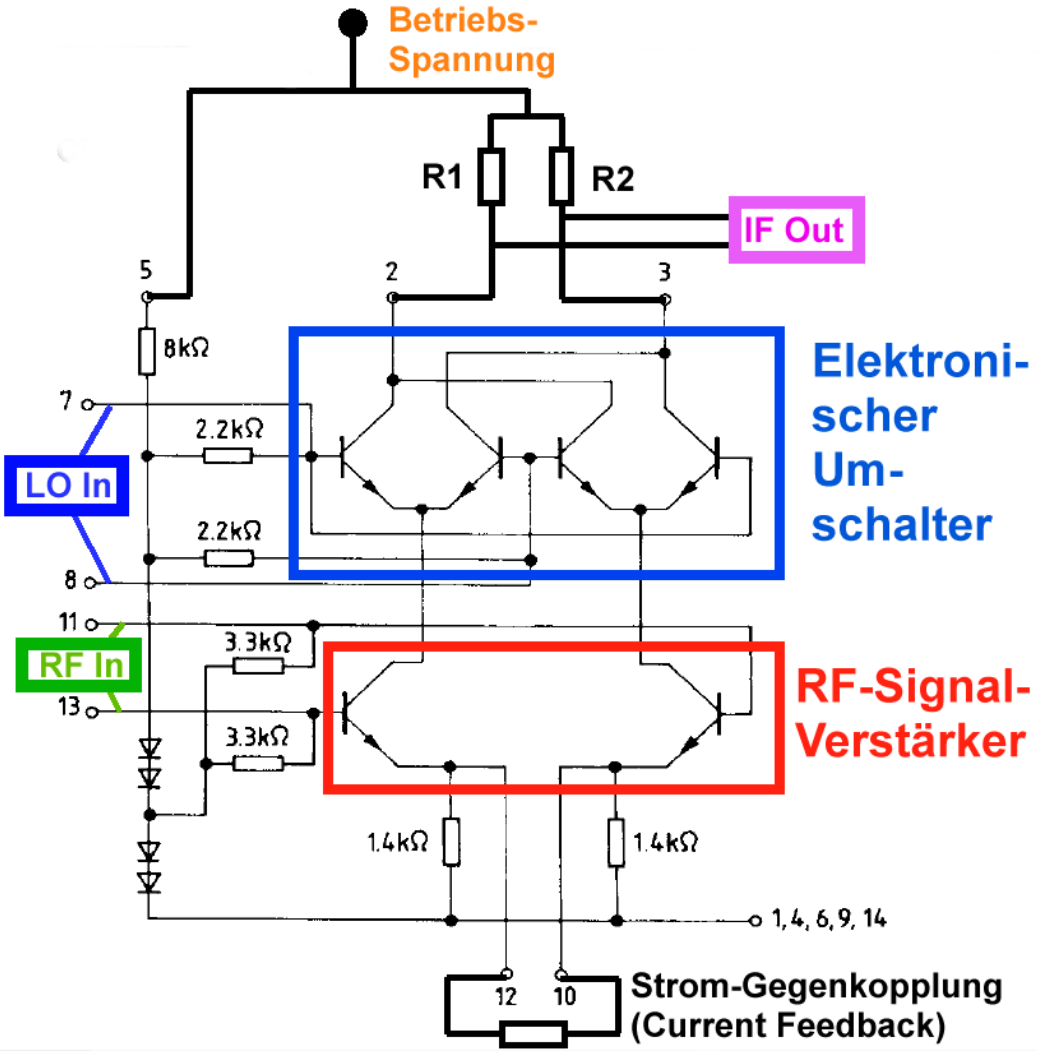
Allerdings scheinen bei sehr hohen Frequenzen (oberhalb von 20GHz) die passiven Mischer (mit Dioden oder FETs oder HEMTs als Schalter) noch die Nase vorn zu haben, wenn es um sehr kleines Rauschen oder extrem hohe Linearitätsforderungen bei hohen Pegeln geht. Da muss man einfach die Veröffentlichungen im Internet oder in den Fachzeitschriften verfolgen und sehen, wie das weitergeht.

23.2. Prinzip

Dieses Bild soll uns weiterhelfen. Es zeigt das Innenleben des ersten käuflichen Gilbert Cell Mixers (S042P), der viele Jahre bis heute überlebt hat und noch käuflich ist. Wir nehmen uns aus einem ganz einfachen Grund dieses historische Bauteil vor:

a) Die Grundschaltung wird prinzipiell auch im modernsten Empfänger- oder Konverter-IC weiter verwendet, aber

b) nur hier finden wir noch genaue Werte für die auf dem Chip eingesetzten Bauteile und können damit das Verhalten exakt simulieren.

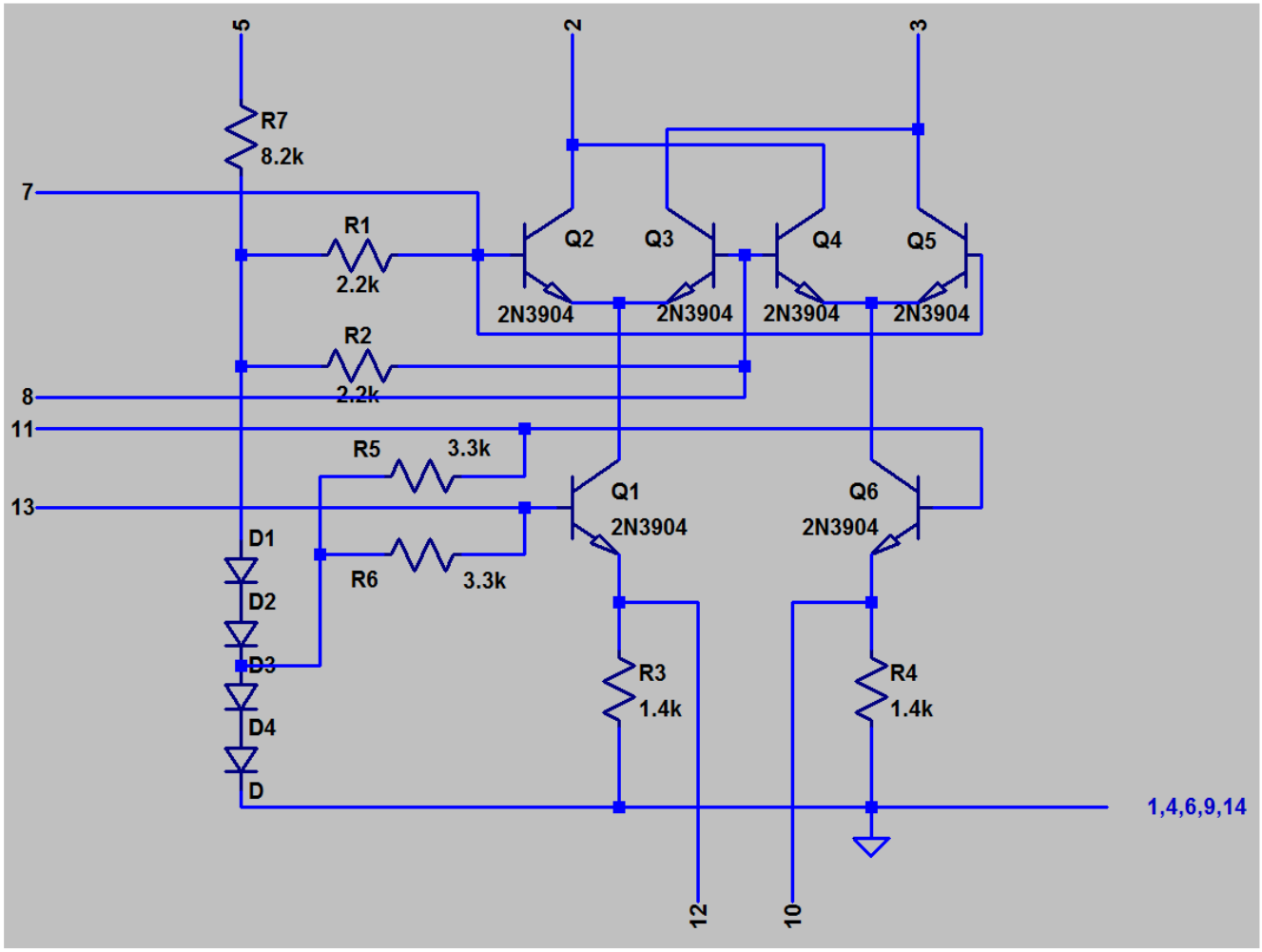


Das symmetrische RF-Signal wird dem unteren Verstärker zugeführt, der als „**Differenzverstärker**“ oder „**Emittergekoppelter Verstärker**“ arbeitet. Dadurch entstehen zwei gegenphasige Wechselströme in den beiden Transistoren, die ihren Weg über einen elektronischen Umschalter zu den beiden Lastwiderstände R1 und R2 nehmen. So entstehen an ihnen zwei gegenphasige Spannungen, deren **DIFFERENZ** das Ausgangssignal „IF Out“ ergibt.
Mit dem LO-Signal kann nun der elektronische Umschalter betätigt und dadurch die Stromrichtung in R1 und R2 im Rhythmus des LO-Signals umgepolt werden.

Das entspricht jedoch eine Multiplikation des RF-Signals mit einem rechteckförmigen LO-Signal und genau das kennen wir schon vom passiven Double Balanced Mixer!

Aufgabe:
Erstellen Sie für die Innenschaltung der Integrierten Schaltung S042P ein LTspice-Simulationsschaltbild. Lassen Sie dabei die externen Zusatzwiderstände weg, bringen Sie jedoch die **korrekten Pin-Nummern als Label** an. Speichern Sie das fertige Schaltbild als „**S042P_01**“ in Ihrem Home-Verzeichnis.

Muster:



Einige Tipps:

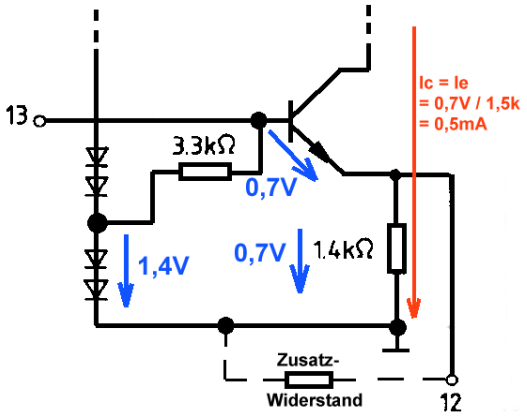
a) Die Transistoren werden zunächst als Bauteil „npn“ platziert. Anschließend fährt man mit der Maus auf das Schaltzeichen, klickt mit der rechten Maustaste und ändert über „**Pick new transistor**“ die Bezeichnung in „**2N3904**“ ab (…dieser ältere Typ entspricht ziemlich genau den auf dem Chip integrierten Transistoren).

b) Drei Transistoren müssen gespiegelt werden. Also drückt man **F7** und fährt mit der Maus auf das Schaltzeichen, um es zu markieren. Mit **<CTRL + e>** lässt sich dann das Symbol spiegeln.

c) Die Pins „1, 4, 6, 9, 14“ sind mit Masse verbunden. Deshalb setzen wir nur ein Massesymbol an diese Leitung und tragen die Pin-Nummern **nur als Text ein (also NICHT als Label!).**

23.3. Untersuchung des RF-Verstärkers

23.3.1. DC-Analyse



Da holen wir uns aus dem Gesamt-Schaltbild **nur den unteren linken Transistor** heraus.

Der in ihm fließende Ruhestrom ist leicht zu bestimmen, denn von den unteren beiden Dioden wird eine Gleichspannung von 2 x 0,7V = 1,4V erzeugt, die über 3,3k an die Basis des Transistors gelegt wird. Davon nimmt sich die Basis-Emitter-Strecke 0,7V weg und es bleiben für den Emitter -Widerstand mit 1,4k ebenfalls noch 0,7V übrig.

Also fließt im Emitter - Widerstand ein Strom von

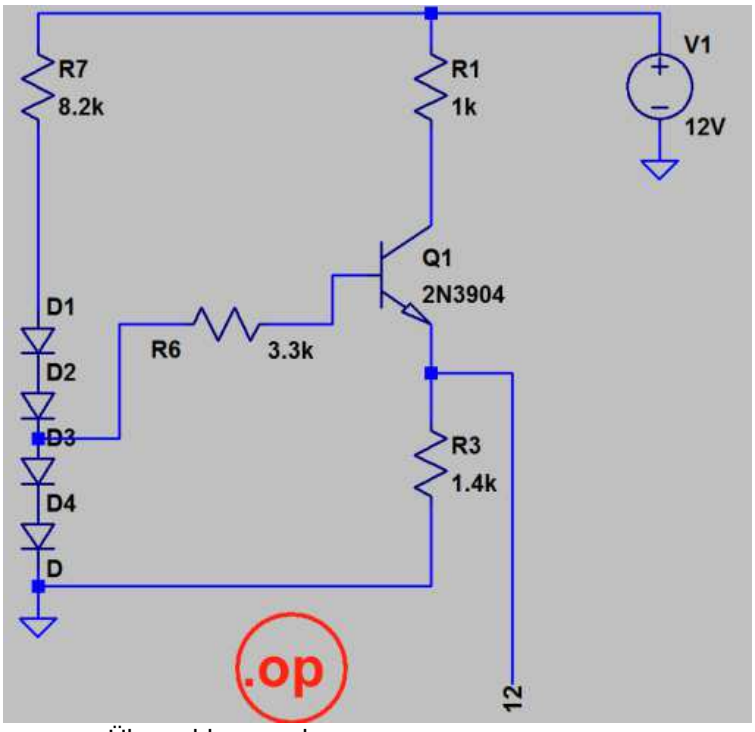
$I_E = I_C = 0{,}7V / 1{,}4k = 0{,}5mA$

Dieser Ruhestrom ist aber sehr klein und bereits Signalamplituden von 10mV führen zur Übersteuerung des Verstärkers.
Deshalb darf an Pin 12 ein Zusatzwiderstand zur Erhöhung des Ruhestromes angeschlossen werden, aber dessen Wert muss größer als 200Ω sein.

Aufgabe:
a) Ermitteln Sie den kleinsten zulässigen Widerstand aus der Normreihe E12, der an Pin 12 angeschlossen werden darf.

Hinweis: die Normreihe E12 verwendet stets ein Vielfaches folgender Werte:
10 / 12 / 15 / 18 / 22 / 27 / 33 / 39 / 47 / 56 / 68 / 82 / 100

b) Welcher Strom fließt nun im Transistor?



(Lösung: 270Ω / 3.1mA)
c) Laden Sie in LTspice das Innenleben des ICs (= Schaltung S042P_01) und **speichern Sie es sofort als S042P_02** ab.
Löschen Sie darin alles bis auf die DC-Schaltung des linken Transistors heraus und speichern Sie diese neue Schaltung unter eigenem Namen. Verpassen Sie dem Transistor Q1 einen Kollektorwiderstand mit 1 kΩ und speisen Sie die Schaltung mit einer Versorgungsspannung von +12V. So sollte das aussehen.

Dann ergänzen Sie bitte die Schaltung um das **SPICE – Kommando „.op“,** denn damit werden alle Spannungen und Ströme für den Arbeitspunkt (= Operating Point) errechnet.

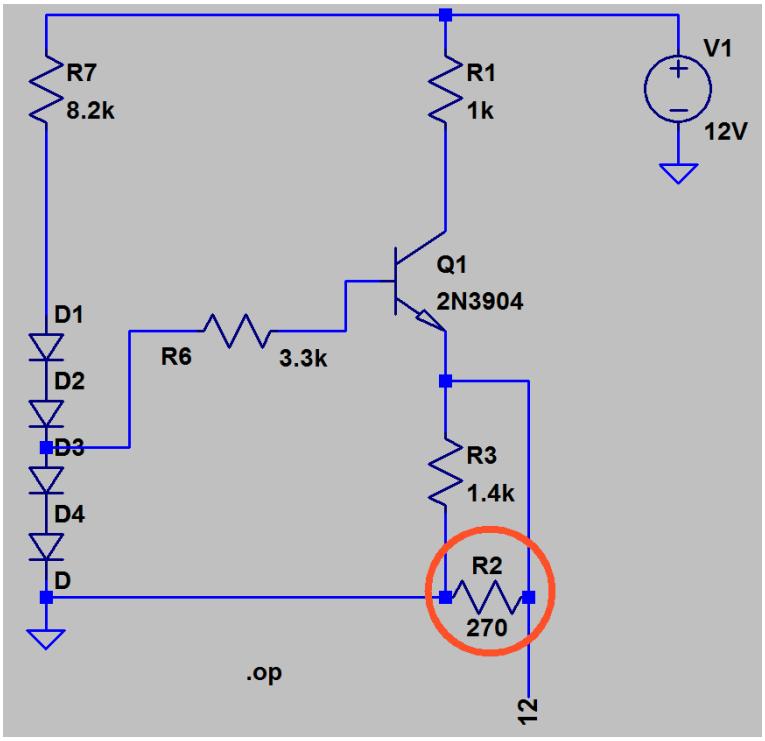
Starten Sie nun die Simulation und vergleichen Sie das Ergebnis mit unserer Überschlags-rechnung.

So wird das aussehen und der uns interessierende Emitter- bzw. Kollektorstrom ist farbig markiert. Er ist etwas kleiner, als unsere Überschlagsrechnung vorausgesagt hat.

```
       --- Operating Point ---

V(n002):        11.5169        voltage
V(n003):        1.31222        voltage
V(12):          0.678329       voltage
V(n005):        1.31705        voltage
V(n004):        2.63416        voltage
V(p001):        1.97561        voltage
V(p002):        0.658524       voltage
V(n001):        12             voltage
Ic(Q1):         0.000483058    device_current
Ib(Q1):         1.46269e-006   device_current
Ie(Q1):         -0.000484521   device_current
I(D4):          0.00114071     device_current
I(D3):          0.00114071     device_current
I(D2):          0.00114218     device_current
I(D1):          0.00114218     device_current
I(R1):          0.000483058    device_current
I(R7):          0.00114218     device_current
I(R6):          -1.46269e-006  device_current
I(R3):          0.000484521    device_current
I(V1):          -0.00162523    device_current
```

---

d) Ergänzen Sie nun die Schaltung um den externen **Zusatzwiderstand von 270Ω**, um den Ruhestrom zu erhöhen. Wiederholen Sie die DC-Simulation. Welcher Ruhestrom fließt nun im Transistor?



(Lösung: knapp 2,7mA)

23.3.2. Die Verstärker-Schaltung

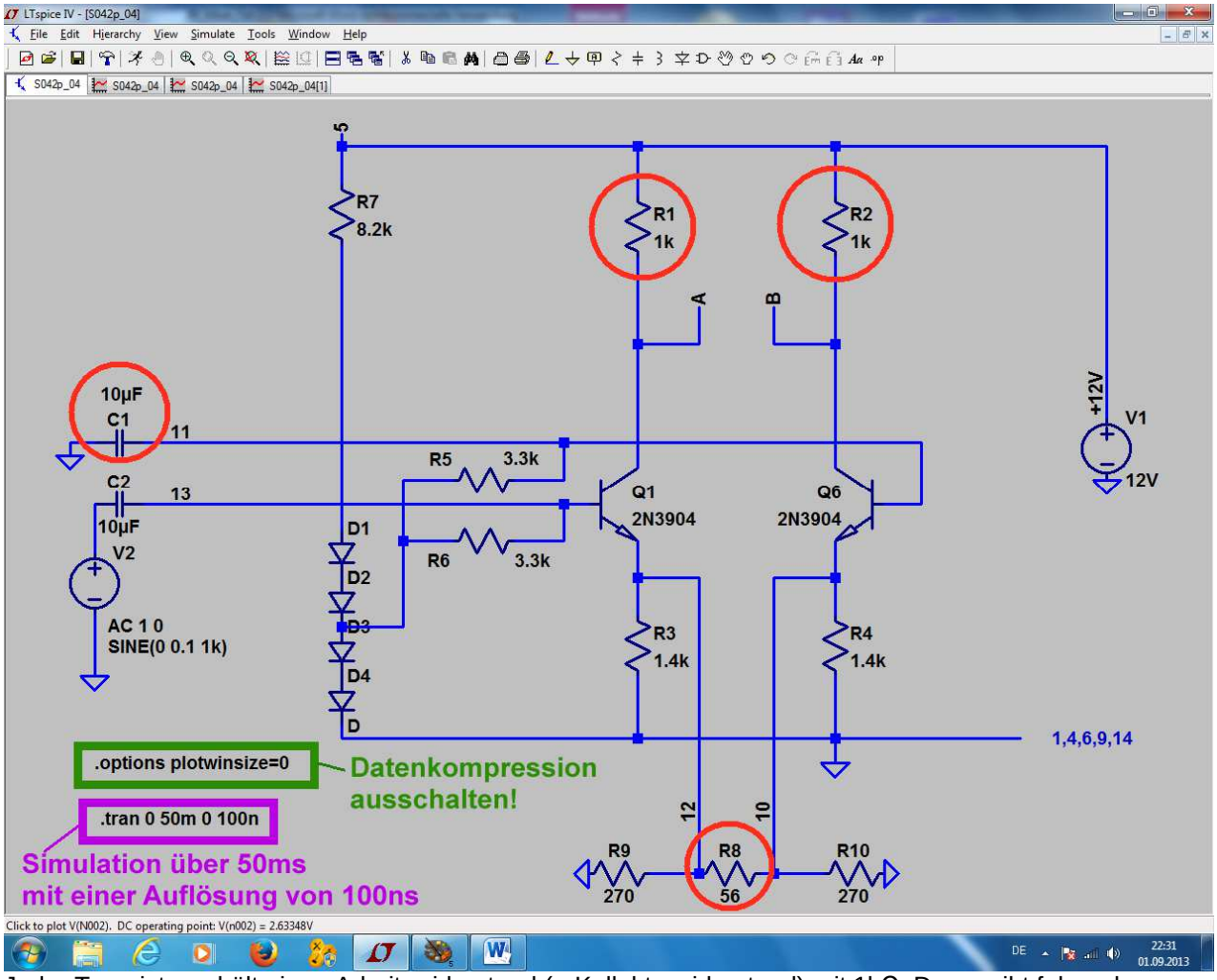
Dazu müssen wir die bisherige Schaltung um die rechte Verstärkerstufe erweitern und zwischen zwei Betriebsarten unterscheiden, nämlich

a) dem Betrieb als **„Emittergekoppelter Verstärker“** und

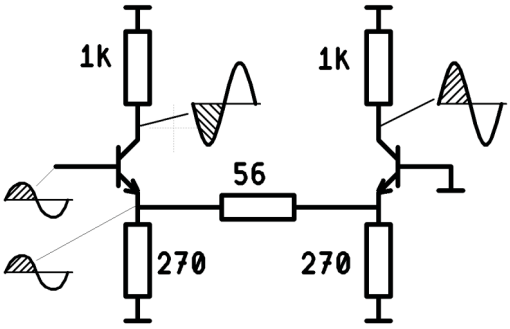
b) dem Betrieb als „**Differenzverstärker“.**

23.3.2.1. Der Emittergekoppelte Verstärker / Unsymmetrischer Betrieb

Dazu erden wir die Basis des rechten Transistors über den Kondensator C1 und steuern die Basis des linken Transistors über den Kondensator C2 mit einem Sinussignal (Spitzenwert = 100mV, Frequenz = 1kHz) an. Die beiden Emitter werden über den Gegenkopplungswiderstand R8 mit 56Ω verbunden.



Jeder Transistor erhält einen Arbeitswiderstand (= Kollektorwiderstand) mit 1kΩ. Das ergibt folgende Gesamtschaltung:



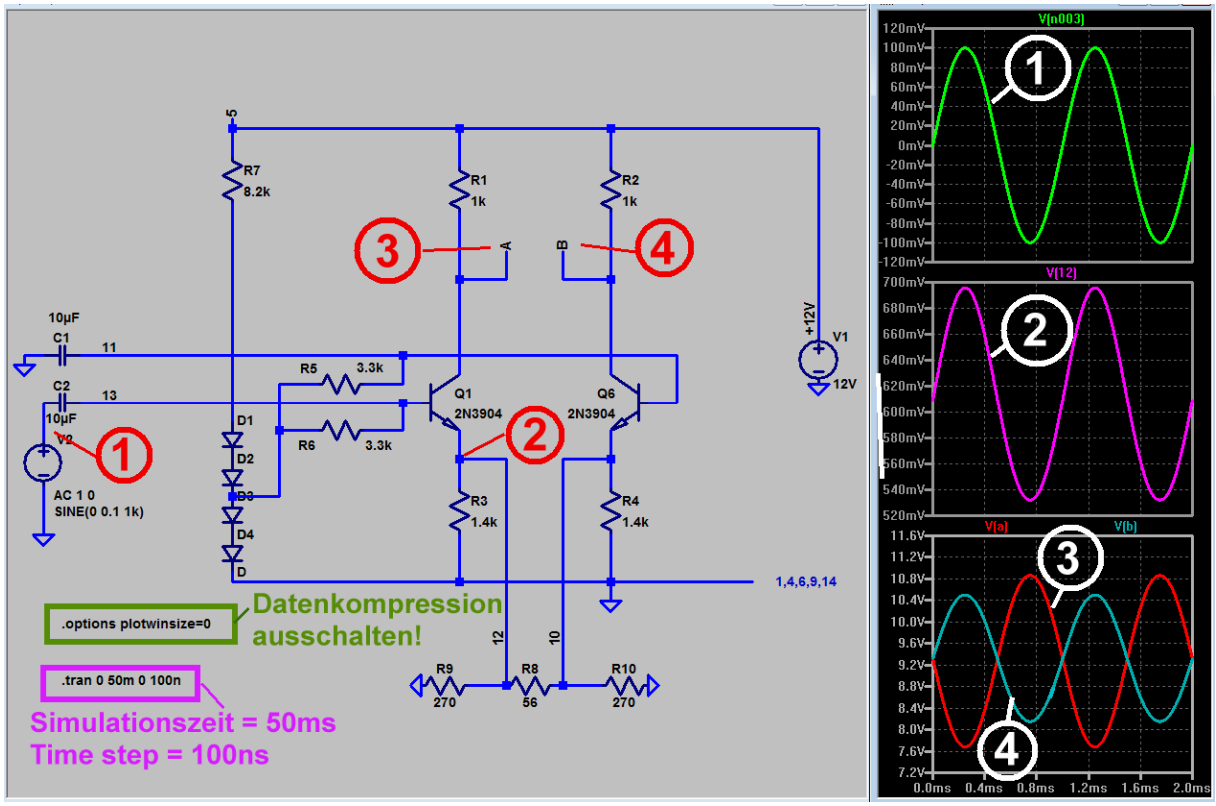
Bevor wir diese Schaltung (**für 50ms mit einem Time step von 100ns**) simulieren, folgen noch einige Informationen.

Wenn man das Wechselstrom-Verhalten der Schaltung betrachtet, erkennt man:

a) der **linke Transistor arbeitet in Emitterschaltung.** Am Kollektor erhält man deshalb ein verstärktes, aber gegenphasiges Ausgangssignal.

b) Die **Emitter-Wechselspannung des linken Transistors ist praktisch gleich groß wie die ansteuernde Spannung an der Basis.** Sie versorgt den rechten Transistor, der in **Basisschaltung** arbeitet. Man erhält nun an dessen Kollektor ebenfalls **ein (fast gleich großes) verstärktes Signal. Aber es ist jetzt in Phase mit der Eingangsspannung!**

Bitte prüfen Sie diese Überlegungen durch eine LTspice-Simulation im Zeitbereich nach und vergleichen Sie diese vier Signale (bezüglich Amplitude und Phasenlage)!
Arbeiten Sie mit **50ms Simulationszeit und einem maximalen Time step von 100ns**.



Erkenntnisse:
a) Die Überlegungen zu den Phasenlagen der Signale stimmten exakt.

b) Die beiden Ausgangssignale An den Punkten „A“ und „B“ sind zwar, wie vorausgesagt, gegenphasig, aber **das „In-Phase“ – Signal am rechten Kollektor ist etwas kleiner.** Hier fließt also offensichtlich ein kleinerer Strom als im linken Transistor. (Gründe: erstens fehlt der nach Masse abfließende Strom durch den rechten 270Ω – Widerstand. Zweitens weist die Basisschaltung einen sehr kleinen Eingangswiderstand auf und da macht sich sofort der Innenwiderstand des linken, ansteuernden Transistors stärker bemerkbar).

Hinweis:
Die Signale an den Punkten „A“ und „B“ sind fast gleich groß, aber gegenphasig. Misst man folglich zwischen A und B mit einem „Differential Voltmeter“, dann ergibt sich **etwas weniger als die doppelte Verstärkung (= Anstieg nur um 5dB statt 6dB)**. **Bitte prüfen Sie das nach!** (Zur Differenzmessung klicken Sie im Schaltbild-Menü auf „Simulate“ und wählen „Set Probe Reference“ auf Punkt „B“ der Schaltung).

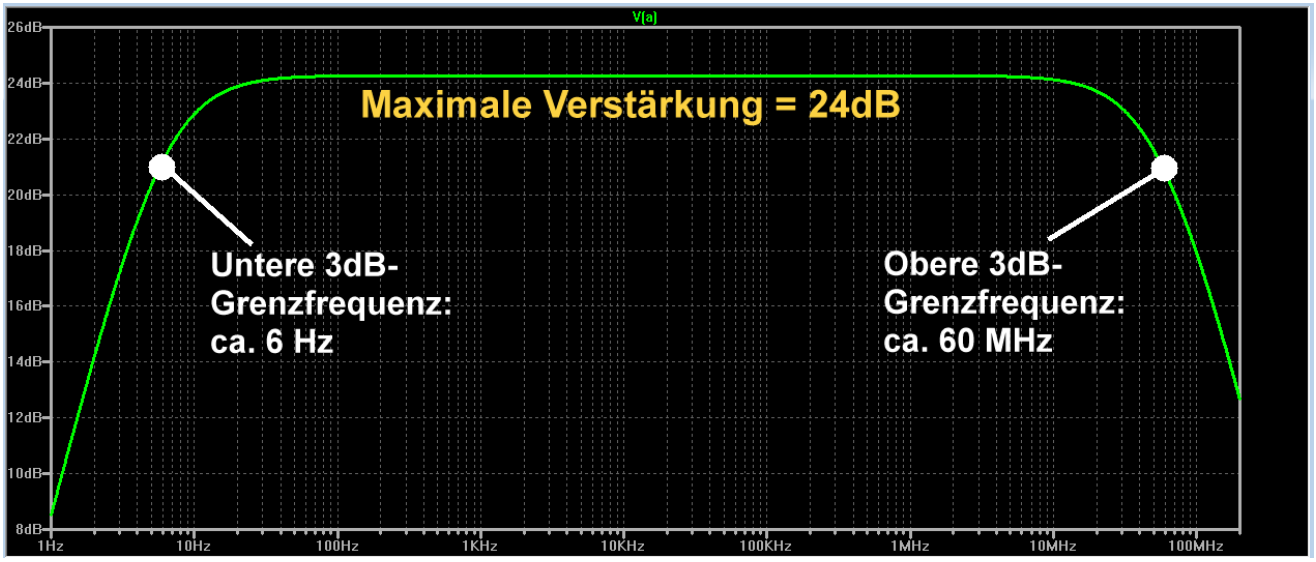
Zusatzaufgabe:
Bestimmen Sie mit einem AC-Sweep den Verstärkungsverlauf an Punkt „A“ und ermitteln Sie die untere und die obere Grenzfrequenz (= „3 dB“ –Grenzen) mit einem **dekadischen Sweep von 1Hz bis 200MHz.**

Simulationskommando: **.ac dec 501 1 200MEG**

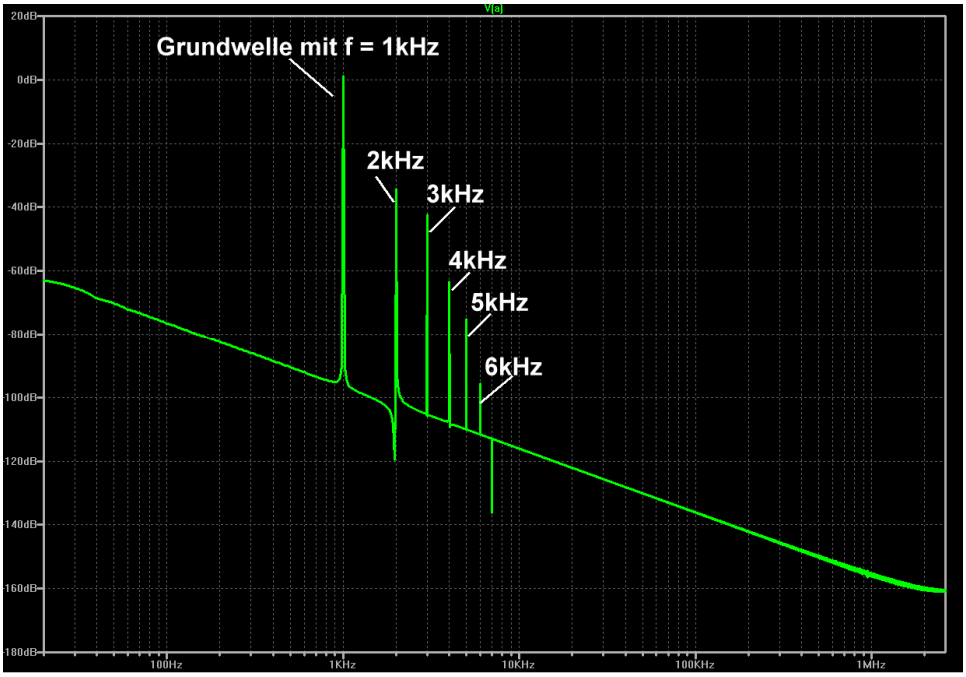
Vergessen Sie bei den Property-Einstellungen der Spannungsquelle nicht die zusätzlichen Einträge

AC amplitude =1 AC phase = 0

für die „small signal AC analysis”!



Zusatzaufgabe:
Simulieren Sie das Frequenzspektrum des Ausgangssignals „A“ beim linken Transistor mit einer FFT. Arbeiten Sie mit 262 144 Samples.



Vergessen Sie nicht, die Datenkompression durch die Direktive **.option plotwinsize=0** auszuschalten.

Wählen Sie einen Amplitudenbereich von **+20dB….-180dB**

und einem Tick von 20dB.

Dieses Ergebnis kann noch verbessert werden, wenn man **vor** dem Start der FFT

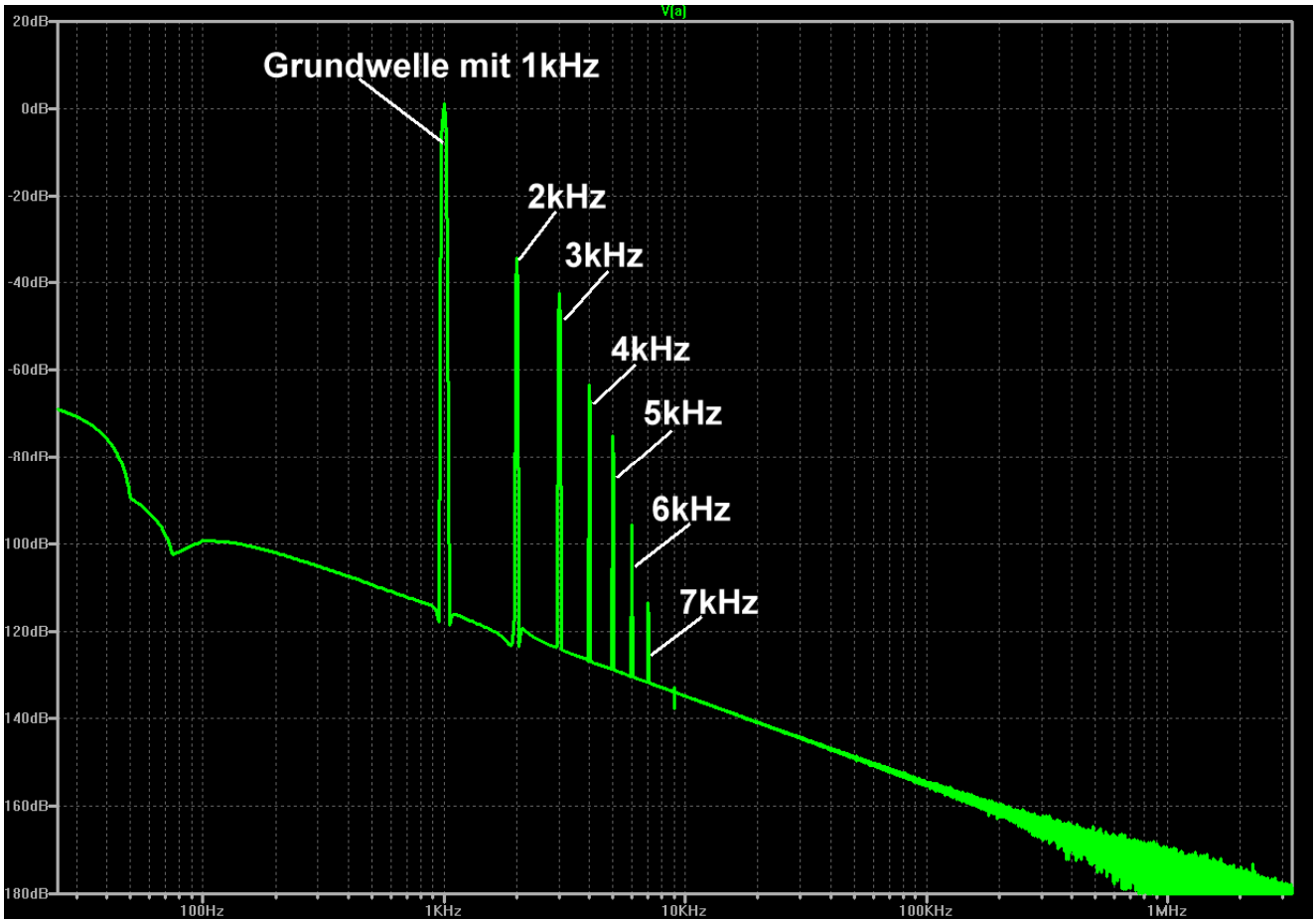
a) eine **Fensterfunktion** über das ausgewählte Sample-Paket legt (Grund: am Anfang und am Ende des Pakets haben wir „Unstetigkeiten“ -- also ein plötzliches Ein- und Ausschalten, das ja in Wirklichkeit nicht vorkommt. Das führt zu zusätzlichen Störspektren, die sich unserem gewünschten Verlauf überlagern).

b) Am Anfang des Signalverlaufs wird die Stufe eingeschaltet und da kann es zusätzliche Störungen geben (…die Ruheströme in den Transistoren fahren hoch und pendeln sich ein, falls Übertrager mit im Spiel sind, kann es je nach Einschalt-Phasenlage zusätzliche Effekte geben usw.). Also **wählt man den Startpunkt der FFT etwas später,** damit diese Schmutzeffekte vorbei sind…

c) Die Mathematik fordert, **dass das zu transformierende Signal ohne „Phasenanschnitt“ genau in die gewählte Simulationszeit passt** (….sonst gibt es Fehler durch die abgeschnittenen Teile der Signalperiode. Das trägt den Namen „**Leakage**“). Auch das korrigiert die Fensterfunktion….

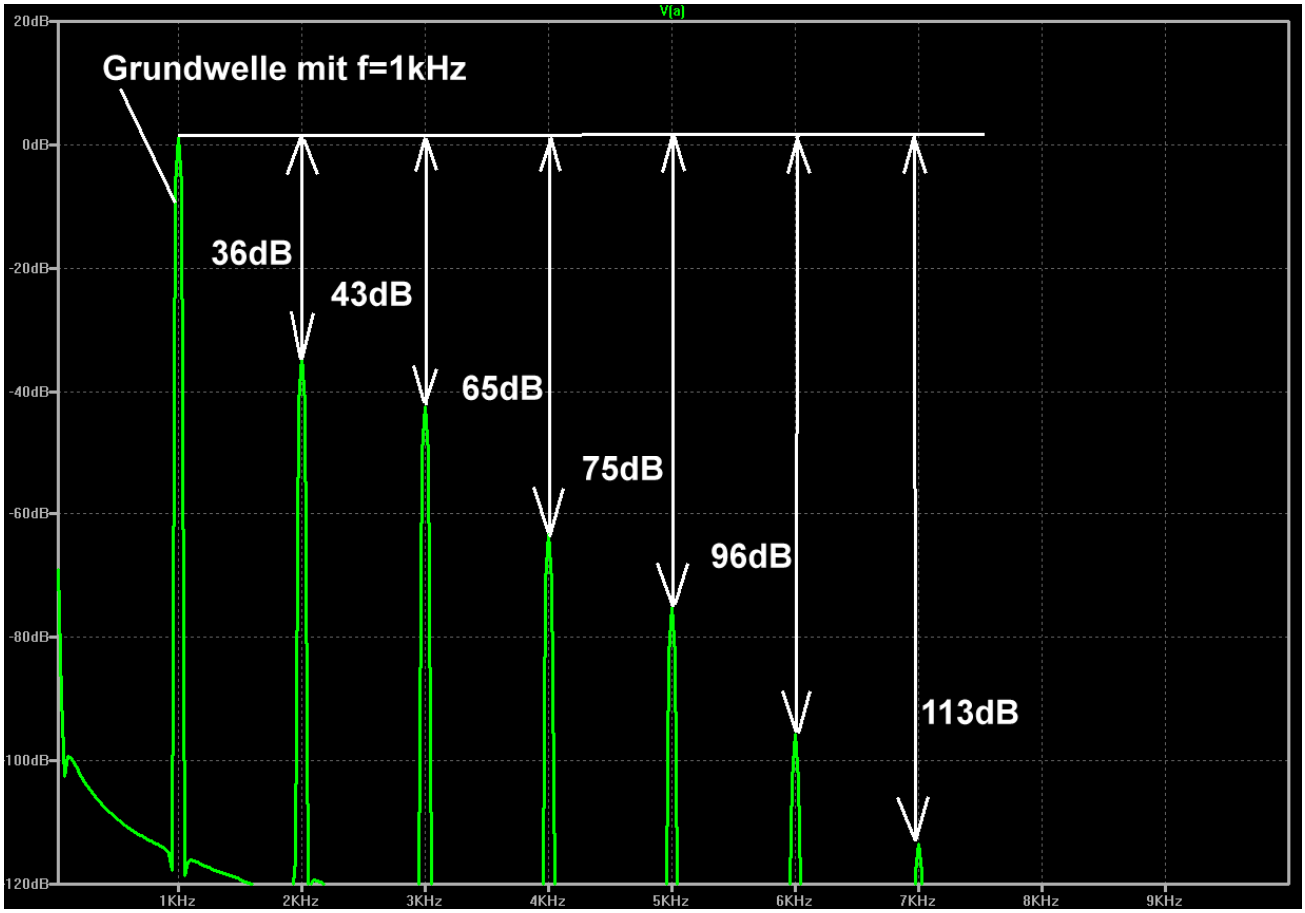
Aufgabe:
a) Wiederholen Sie die FFT und legen Sie ein **Hamming-Fenster** über den Signalverlauf. Wählen Sie zusätzlich einen **Zeitbereich von 5ms…..50ms** für die FFT. Vergleichen Sie das Ergebnis mit dem obigen Bild (Achtung: **bitte denselben Bereich von +20…-180dB mit einem Tick von 20dB bei der senkrechten Achse einstellen!)**

Lösung:



Die Auswirkung ist überzeugend. Es sinkt das „Grundrauschen“ und da wird nun sogar die **Linie für f = 7kHz sichtbar.**

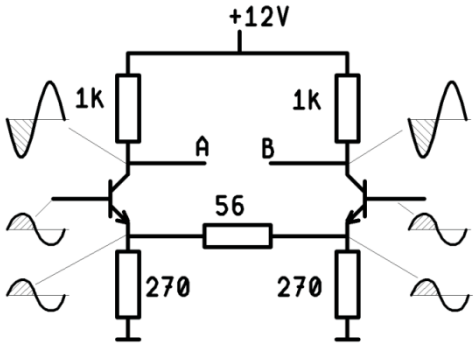
b) Ermitteln Sie für das gegebene Eingangssignal (Spitzenwert = 100 mV / Frequenz = 1 kHz) die genauen **Dämpfungen der harmonischen Verzerrungen in dB gegenüber der Grundwelle im Bereich von 0…10kHz** Wählen Sie dazu eine **lineare Darstellung bei der Frequenzachse und einen Anzeigebereich von +20….-120dB bei der senkrechten Achse (Tick: 20dB)**.



Erkenntnis:
Da auch geradzahlige Harmonische (= 2 / 4 / 6kHz…) vorhanden sind, muss das Ausgangssignal **unsymmetrisch verzerrt** sein!
(Bei perfekter Symmetrie treten nur ungeradzahlige Oberwellen auf)

23.3.2.2. Der Differenzverstärker

Dazu steuern wir zuerst den linken UND den rechten Transistor an der Basis mit demselben Signal an.



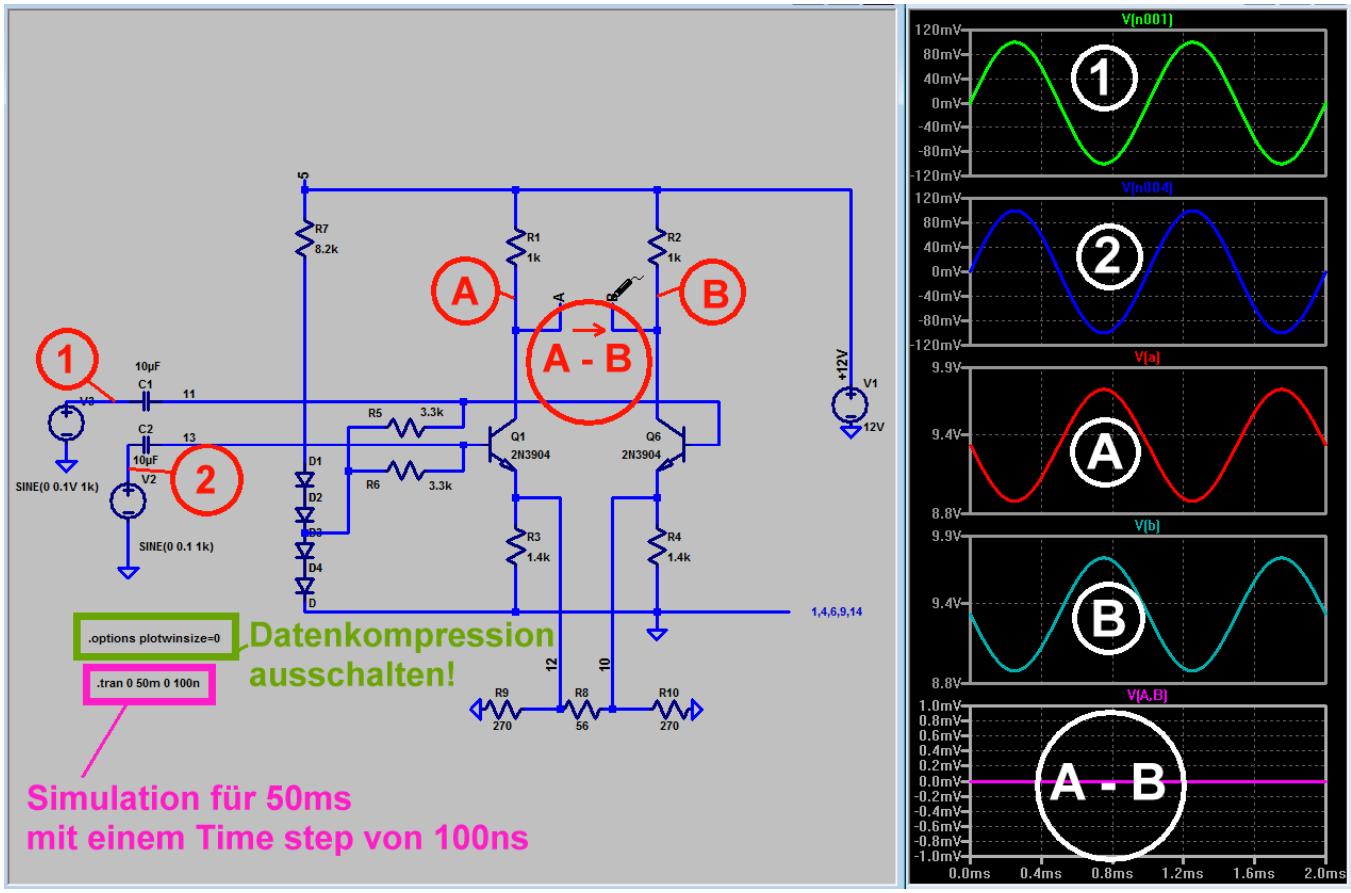
In beiden Transistoren werden nun dieselben Ströme fließen und an den beiden Emitter-Anschlüssen sind dieselben Spannungen zu messen.
Folglich wird über den Emitter-Verbindungswiderstand mit 56Ω kein Strom fließen und an den beiden Kollektoranschlüssen werden gleiche Spannungen gegen Masse zu messen sein.
(Die sind natürlich gegenüber dem Eingangssignal verstärkt und gegenphasig!)

Dann werden die Spannung zwischen „A“ und „B“ aber gleich Null sein!

Aufgabe:
Simulieren Sie diesen Betriebsfall mit LTspice und steuern Sie beide Transistoren mit demselben Spitzenwert von 100mV bei einer Frequenz von 1kHz an! Lassen Sie sich die Eingangs- und Ausgangs-Signale sowie die Differenz zwischen A und B darstellen.

(Na, wissen Sie noch, wie man so eine Spannungsdifferenz in LTSpice ermittelt? Da gibt es unter „View“ den Punkt „Set Probe Reference“. Wenn man den wählt und anschließend „A“ in der Schaltung anklickt, wird beim folgenden Anklicken von „B“ die Potentialdifferenz dargestellt)

Sobald sich jedoch die beiden ansteuernden Signale an den Basisanschlüssen unterscheiden, kann man zwischen A und B die verstärkte Differenz der Eingangssignale messen!



Aufgabe:

Steuern Sie jetzt die linke Basis mit einem Spitzenwert von 100mV, die rechte Basis dagegen nur mit einem Spitzenwert von 80mV an. Zwischen A und B sollten Sie dann die Differenz von 20mV, verstärkt um mehr als 20dB (= also mehr als um den Faktor 10) beobachten können.

23.3.2.3. Symmetrischer Betrieb

In Kapitel 3.2.1. haben wir die Schaltung zur Verstärkung eines Sinussignals verwendet und erkannt, dass der dort verwendete unsymmetrische Betrieb doch zu kleinen Problemen führt (U_A war größer als U_B usw). **Im symmetrischen Betrieb (= gegenphasige Ansteuerung beider Basisanschlüsse) benimmt sie sich dagegen vorbildlich und wird deshalb auch meist so betrieben!**
Deshalb realisieren wir nun eine symmetrische Ansteuerung mit einem „**Symmetrier-Übertrager**“ am Eingang. Der besteht aus zwei Induktivitäten „ind2“ aus dem Bauteilvorrat, denen ein **L-Wert von je 10mH** zugewiesen wird (…10mH ergeben einen induktiven Widerstand von ca. 70Ω bei 1kHz).

Hinweise:
Gleiche Induktivitäten auf der Primär- und Sekundärseite ergeben ein Übersetzungsverhältnis von ü = 1:1. Es gilt nämlich die Beziehung

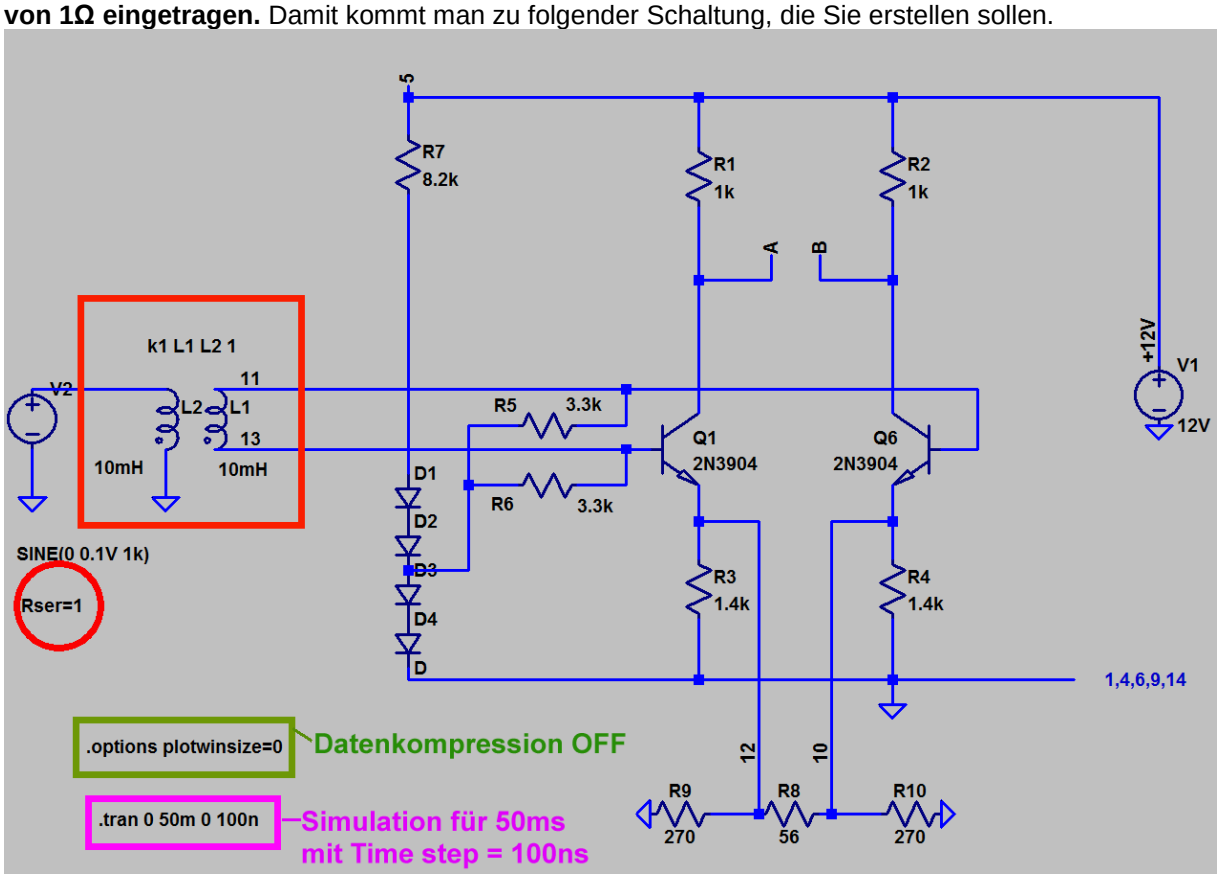
$$L1 = ü^2 L2$$

Die beiden Spulen werden aber erst dann zum Transformator, wenn durch die SPICE-Direktive

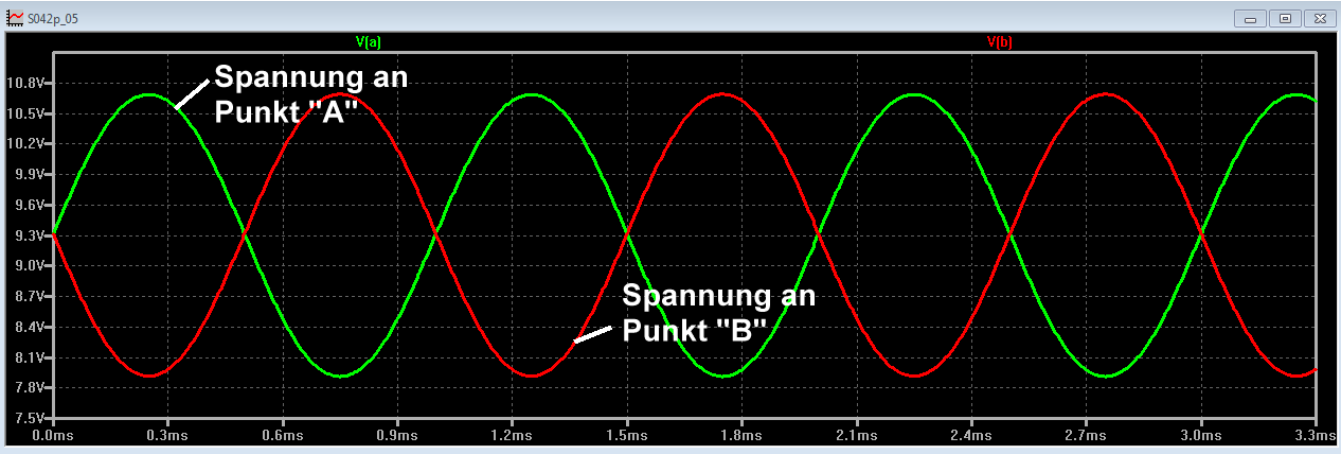
k1 L1 L2 1

eine magnetische Kopplung von k1 = 100% vorgesehen wird.

Da die ansteuernde Signalquelle V2 direkt an die Primärwicklung des Übertragers angeschlossen ist, verlangt SPICE unbedingt einen „**Schutzwiderstand**“, damit der Strom aus der Quelle im Extremfall nicht bis Unendlich steigen kann. **Deshalb wurde im Menü der Quelle gleich ein Serienwiderstand von 1Ω eingetragen.** Damit kommt man zu folgender Schaltung, die Sie erstellen sollen.

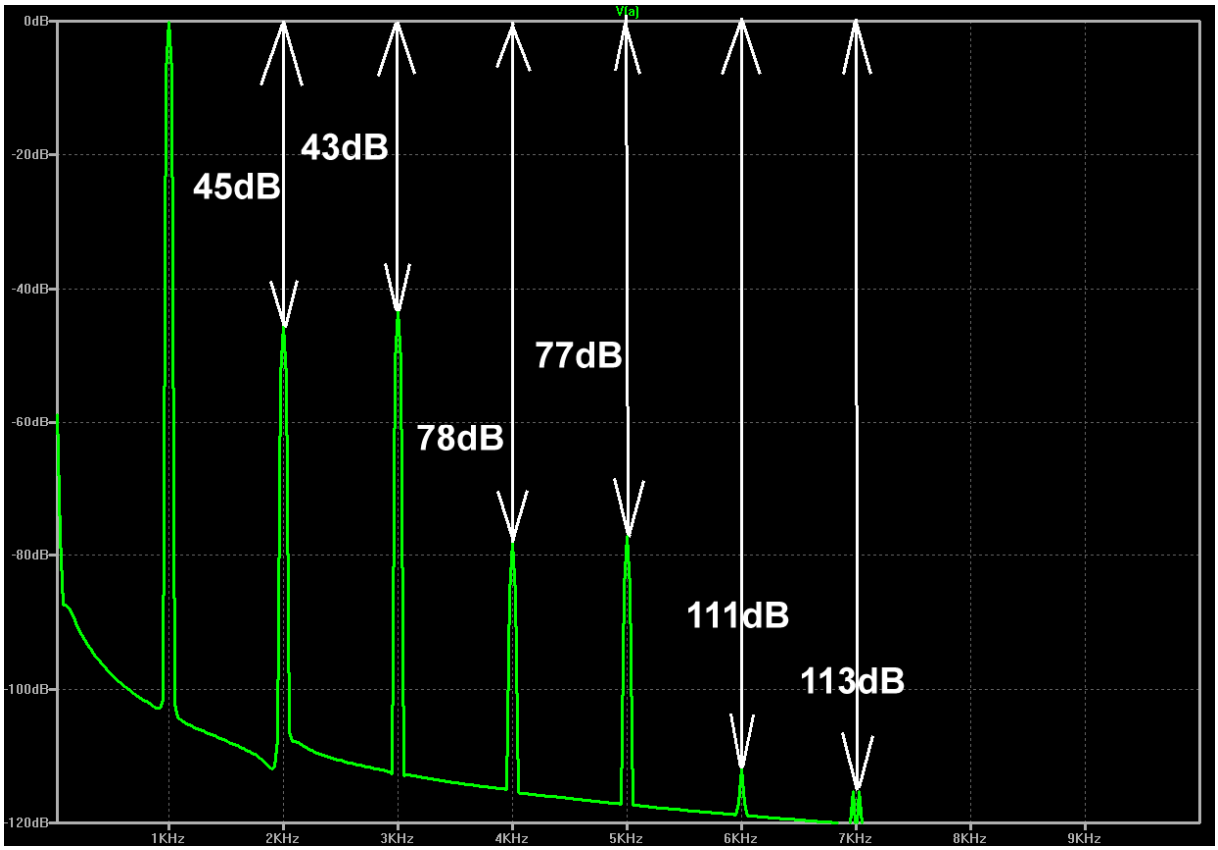


Aufgabe:
a) Vergleichen Sie nun über eine LTspice-Simulation die Amplituden der beiden Spannungen U_A und U_B (gegen Masse gemessen).



Ergebnis: nun sind die beiden Spannungen exakt gleich groß.

b) Bestimmen Sie mit LTspice über eine FFT das **Frequenzspektrum der Spannung U_A** und vergleichen Sie die Dämpfung der entstandenen Verzerrungen mit dem unsymmetrischen Betrieb in Kapitel 3.2.1. **Arbeiten Sie mit einer Darstellung von 0 dB….-120dB, einem Tick von 20dB und einer linearen Frequenzachse von 0….10kHz**.



Ergebnis:
Wegen der verbesserten Symmetrie innerhalb der Schaltung sind die Amplituden der geradzahligen Harmonischen deutlich kleiner geworden. Die Dämpfungswerte der ungeradzahligen Oberwellen bleiben unverändert.

23.4. Die Gilbert-Cell, als Double Balanced Mixer (DBM) eingesetzt.

Dazu brauchen wir nun das **vollständige Innenleben des S042P samt dem elektronischen Umschalter**. Dieser **Umschalter wird vom LO-Signal betätigt**, aber er benötigt zum Schalten der Transistoren (im Vergleich zum Ringmodulator mit Dioden) deutlich weniger Amplitude bei der ansteuernden Schaltspannung.

Betreiben wir ihn mal mit folgenden Signalen:

LO-Signal: Spitzenwert = 1V / LO-Frequenz = 20kHz
RF-Signal: Spitzenwert = 50mV / RF-Frequenz = 1kHz

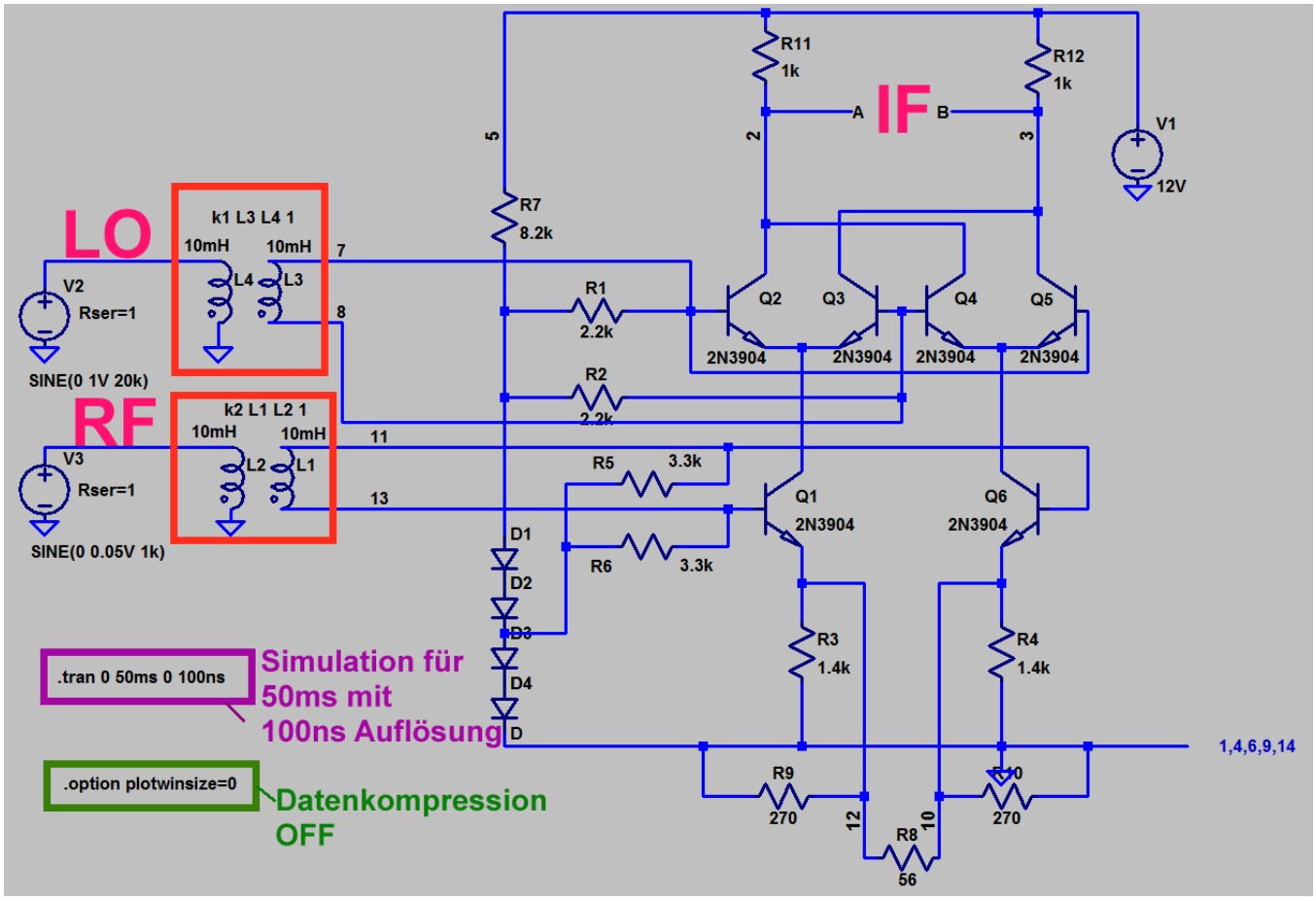
Bitte die SPICE-Direktive **.option plotwinsize=0“**
nicht vergessen, damit für eine korrekte FFT die Datenkompression ausgeschaltet wird.

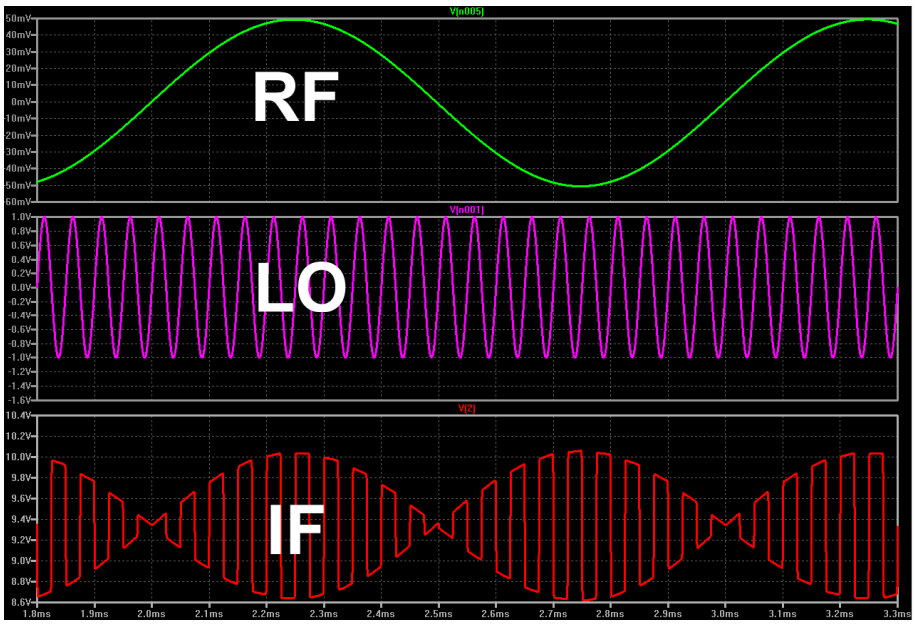
Außerdem sollte man sicherheitshalber den „Button mit dem Hammer“ anklicken und unter „SPICE“ auf **„Gear“** als Integrationsmethode schalten (….ist bei der Simulation hoher Frequenzen sicherer….).

Bei den **Übertragern** bleiben wir bei einer Hauptinduktivität von je 10mH und einem Übersetzungsverhältnis von ü = 1:1

Simuliert wird für **50 Millisekunden mit einem maximalen Timestep von 100 Nanosekunden. Das ergibt 500 000 echte Samples für die FFT**.

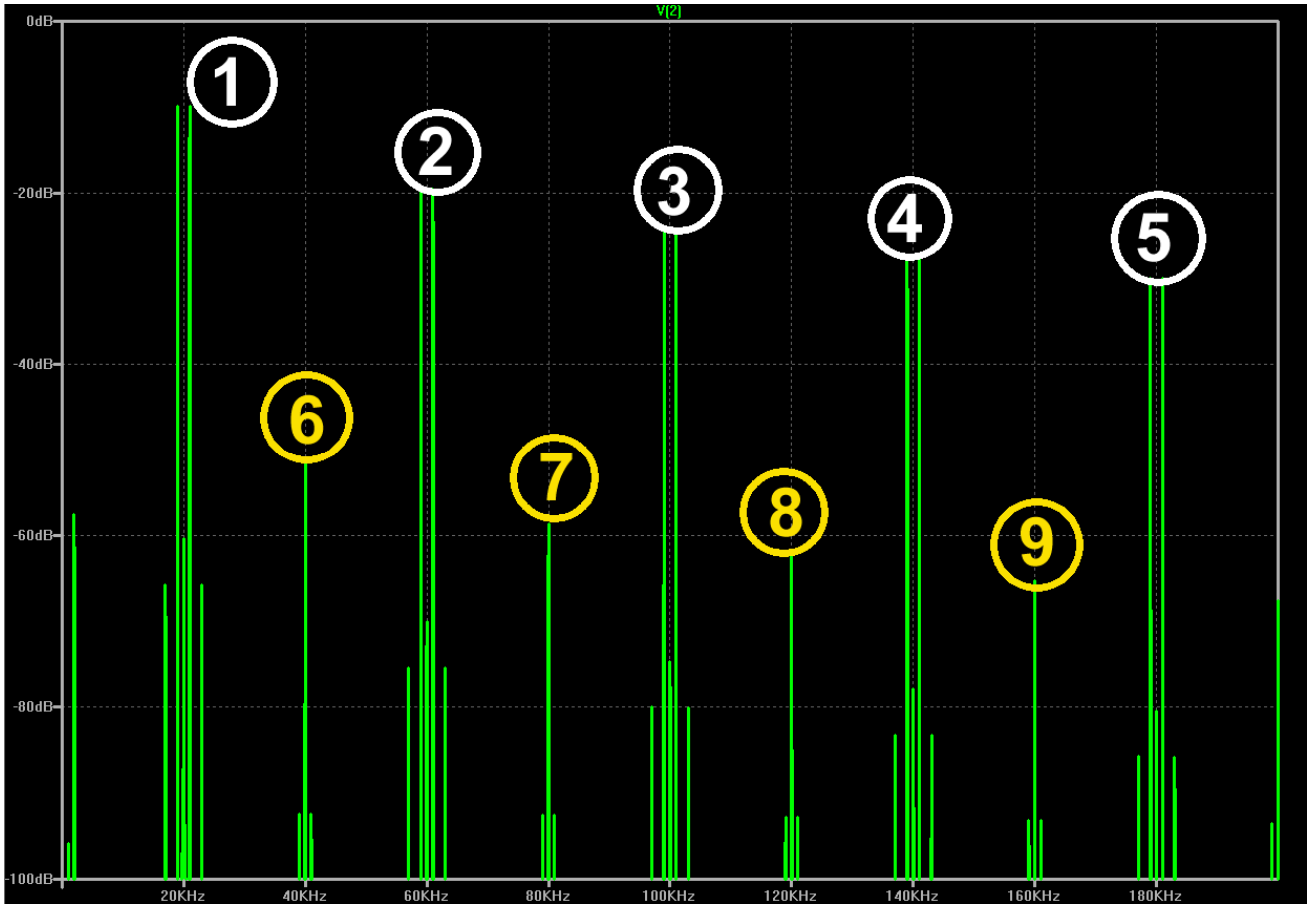
Das ergibt folgende neue Schaltung für den vollständig symmetrischen Betrieb
Bitte **erstellen Sie diese Schaltung** und führen Sie eine Simulation im Zeitbereich durch. **Stellen Sie den zeitlichen Verlauf von RF-Signal, LO-Signal und Signal an Punkt „A“ (= IF-Output) untereinander dar.**





Aufgabe:
Ermitteln Sie nun über eine FFT das **Frequenzspektrum** des obigen IF-Signals an Punkt A.
Berechnung mit Hamming-Fenster, 262 144 Samples und einem Time Range von 5ms bis 50ms.

Ergebnis-Darstellung:
Lineare Frequenzachse von 0…200kHz und einem Tick von 20kHz
Amplitudenbereich von 0dB bis -100dB und einem Tick von 20dB



Aufgabe:

Ordnen Sie den einzelnen Linien oder Liniengruppen die genaue Herkunft zu!

1: **19kHz und 21 kHz = Summen- und Differenzfrequenz aus 20kHz und 1kHz**

2:

3:

4:

5:

6:

7:

8:

9:

Aufgabe:
a) **Vermindern** Sie die Amplitude des LO-Signals **von 1V auf 0,5V**. Simulieren Sie nochmals und lassen Sie die FFT das neue Spektrum des IF-Signals bestimmen.
Welche Änderungen finden sich gegenüber dem vorigen Betriebsfall (RF-Spitzenwert = 50mV)?

Aufgabe:
b) **Verwenden Sie nochmals ein LO-Signal mit 0,5V und erhöhen Sie die Amplitude des RF-Signals von 50mV auf 100mV** und lassen Sie das neue Spektrum berechnen. Welche Änderungen erhält man darin gegenüber dem Fall b)?

Aufgabe:
c) **Erhöhen Sie (bei gleichbleibender RF-Signalamplitude von 100mV) die Amplitude des LO-Signals wieder auf 1V** und prüfen Sie, welche Verbesserungen und Veränderungen sich nun ergeben.

Aufgabe:

Betreiben Sie die Schaltung bei **höheren Frequenzen**. Nehmen Sie dazu folgende Änderungen vor und untersuchen Sie jetzt das Spektrum des IF-Signals.

RF-Signal: **50mV / 500kHz**

LO –Signal: **1V / 5MHz**

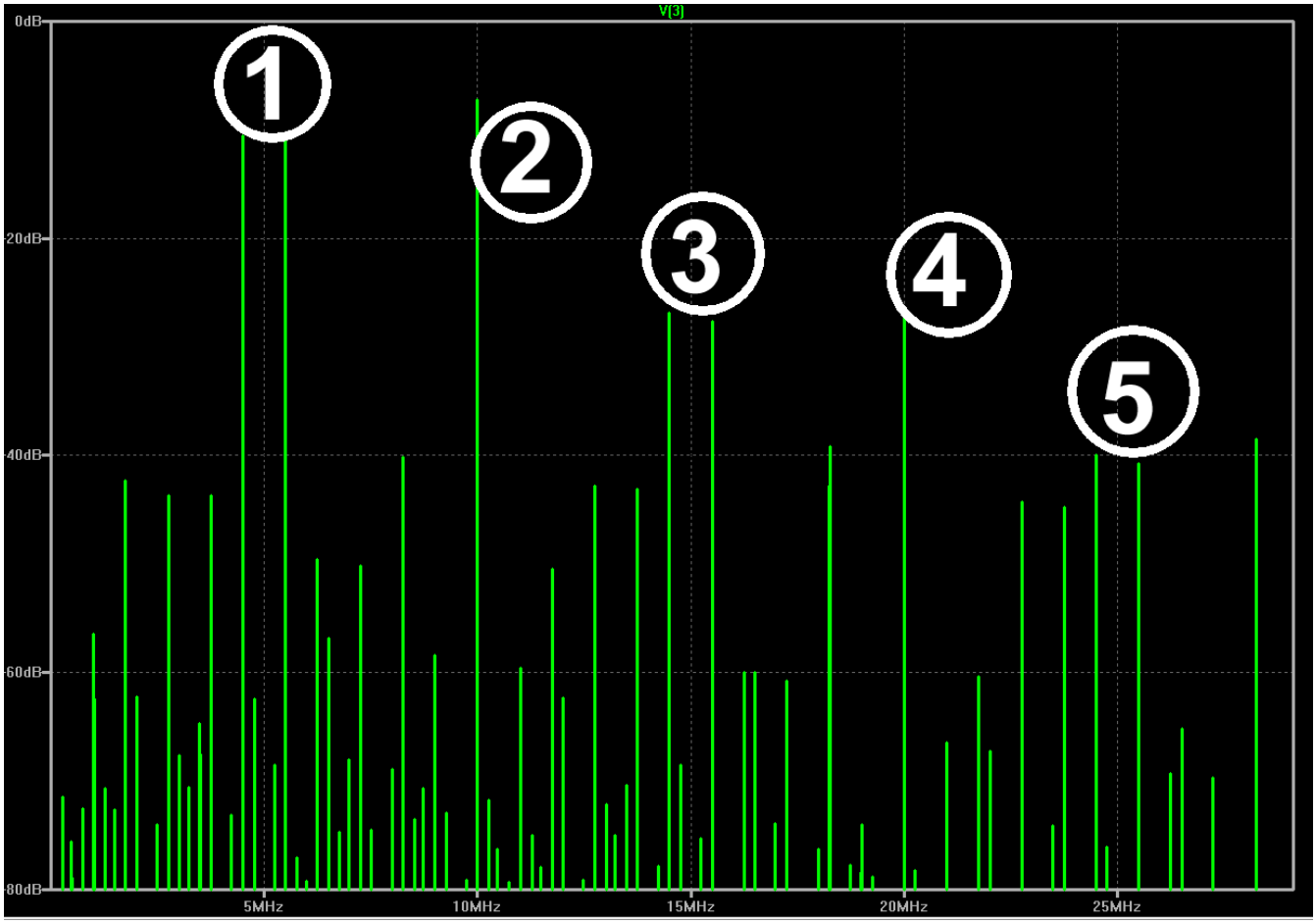
Übertrager-Induktivitäten: je 2µH

Simulationsbereich: 0….5ms / Time step = 10ns

FFT: Hamming –Fenster / Zeitfenster = 0,5ms…..5ms

FFT-Ergebnis-Darstellung: lineare Zeitachse von 0 bis 30MHz / Tick = 5MHz
Amplitudenbereich von 0dB…-80dB mit einem Tick von 20dB

So sollte das Ergebnis aussehen:



Wenn es so aussieht:

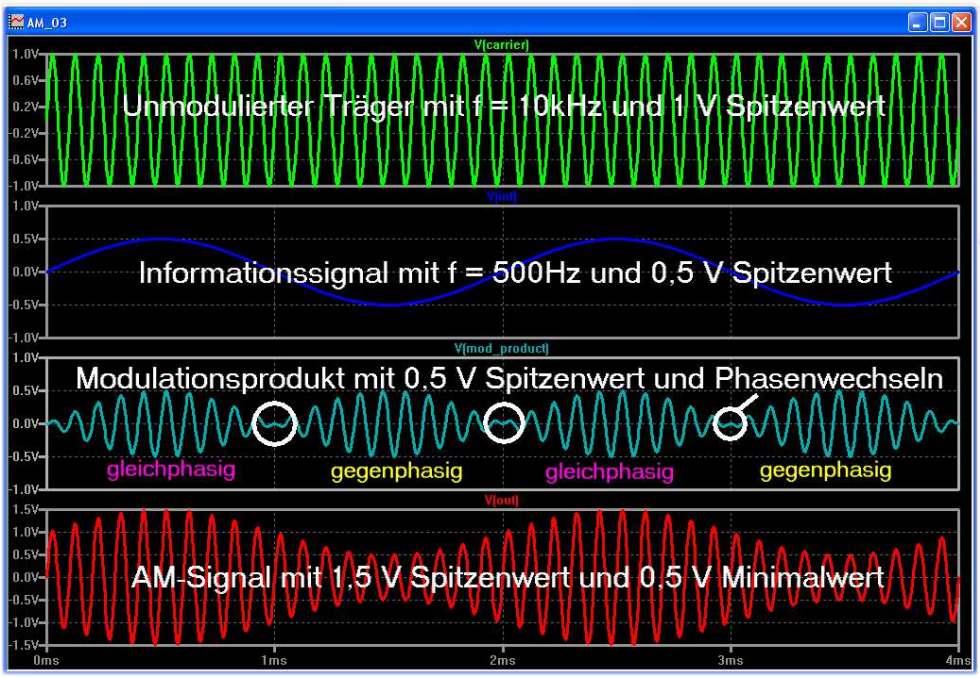
Bitte identifizieren Sie wieder die Signale 1….5.

Vergleichen Sie das Spektrum in Bezug auf die Dämpfung unerwünschter Signale mit der vorigen Simulation (LO-Signal =1V / 20kHz, RF-Signal = 100mV / 1kHz)

24. Achtzehntes Projekt: Modulation

24.1. Prinzip der Amplitudenmodulation (AM)

Sie ist wohl die älteste Modulationsart überhaupt, denn damit hat man schon vor ca. 100 Jahren bei der „Drahtlosen Telegrafie“ mit der Morsetaste angefangen und dann folgte die Übertragung von Sprache und Musik. Der Grundgedanke ist einfach: Audiosignale haben eine viel zu große Wellenlänge, um sie über Antennen mit erträglichem Wirkungsgrad drahtlos abstrahlen zu können. Also sucht man nach einem Lastesel (= Trägersignal mit hoher Frequenz, damit die Antenne klein wird), dem man die Last (= Information) problemlos für den Transport aufbürden kann. Dazu wird die Amplitude des Trägers im Rhythmus der Information geändert. Mathematisch gesehen wird hier eine ***Multiplikation von zwei verschiedenen Signalen und eine zusätzliche Addition*** durchgeführt und das wollen wir uns näher ansehen. **Das folgende Bild** zeigt das Geheimnis:



Ganz oben ist die konstante Trägerspannung mit f = 10kHz und einem Spitzenwert von 1V zu sehen.

Darunter folgt die Information als Sinus mit 500Hz und 0,5V Spitzenwert.

Wenn wir nun diese beiden Signale miteinander ***multiplizieren***, dann muss das Ergebnis ein Signal mit Trägerfrequenz sein, das

a) dem Amplitudenverlauf der Information folgt, aber

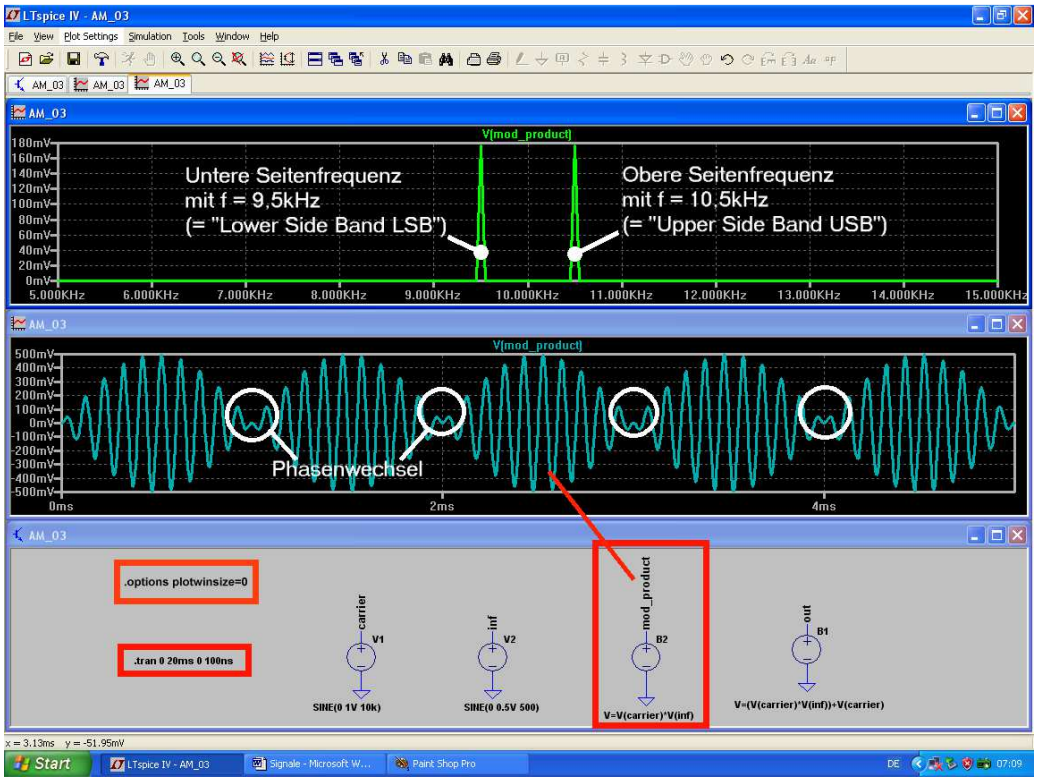
b) jedes Mal die Phase umkehrt, wenn die Informationsspannung negativ ist.

Dieses Multiplikationsergebnis trägt den speziellen Namen „***Modulationsprodukt***“. Es findet sich im dritten Diagramm von oben und wenn wir das zum **unmodulierten Trägersignal addieren**, wird die Gesamtspannung größer, wenn das Modulationsprodukt gleichphasig mit dem Träger ist. Ebenso erhalten wir ein kleineres Gesamtsignal in denjenigen Abschnitten, wo Modulationsprodukt und Träger gegenphasig zueinander sind.

Die Gesamtspannung im untersten Diagramm -- also unser AM-Signal -- schwankt nun wie gefordert im Rhythmus der Information. Also sollte man sich jetzt fragen, wie viele unterschiedliche Frequenzen hier im Spiel sind -- schließlich handelt es sich nicht mehr um einen konstanten Sinus! Beim Träger ist es einfach, denn das ist nun wirklich der „konstante Sinus“ mit nur einer Linie im Spektrum. Beim Modulationsprodukt müssen wir die Mathematik bemühen, denn da gibt es eine hübsche Formel für die Multiplikation von zwei Cosinus-Signalen:

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

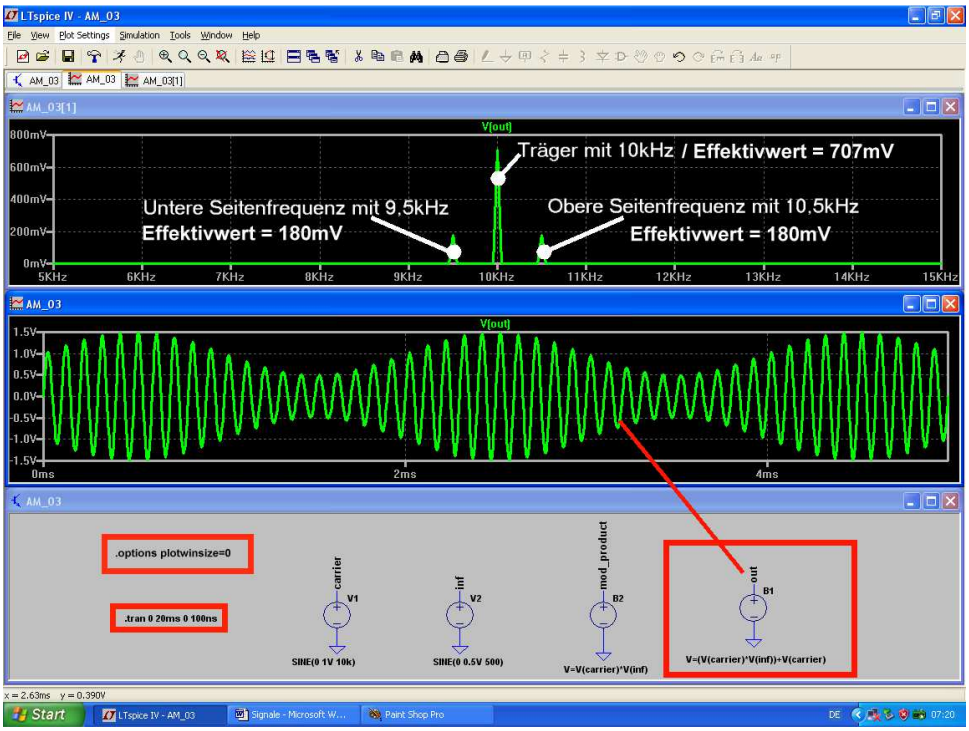
Wir erhalten als Produkt die Summen- und die Differenzfrequenz der beiden beteiligten Cosinussignale. Träger und Information selbst tauchen NICHT MEHR darin auf!



Das muss auch die Simulation zeigen und die **„Behaviour Voltage bv“** aus dem Bauteile-Vorrat liefert uns das nötige Handwerkszeug zur Multiplikation der beiden Signale. Dieses Bild bestätigt das: es zeigt die beiden Anteile:

a) die **Summenfrequenz**, also **10kHz + 0,5kHz = 10,5kHz**
b) die **Differenzfrequenz**, also **10kHz – 0,5kHz = 9,5kHz**

Sie werden als **Obere und Untere Seitenfrequenz** bezeichnet und ihre Amplituden sind gleich groß. Der Spitzenwert beider Signale beträgt je 50% der Information, also 250mV. Und im Spektrum werden daraus -- wegen der dort gültigen Effektivwert-Anzeige -- ca. 180mV.



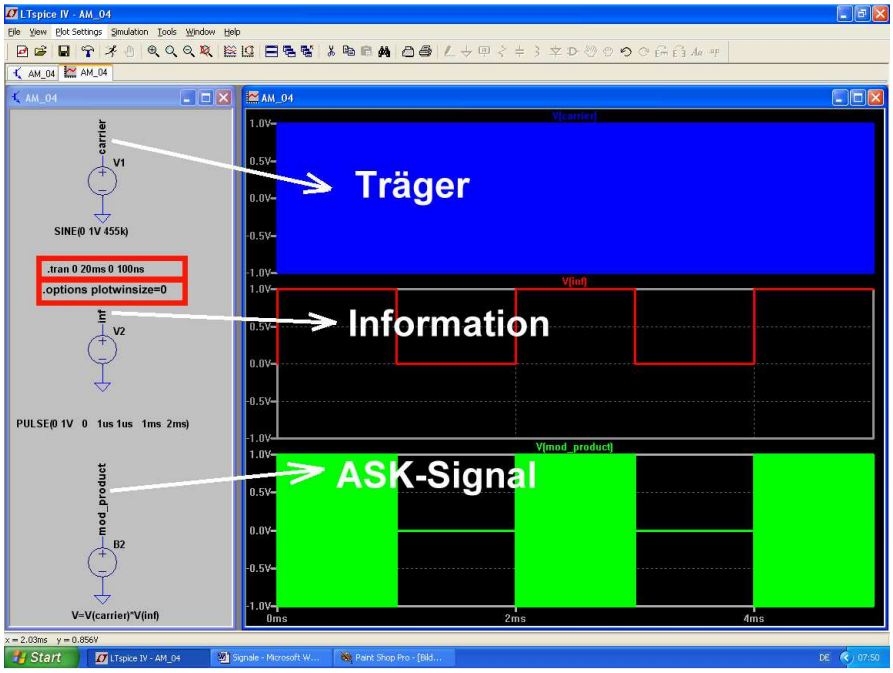
Beim kompletten AM-Signal wird noch das Trägersignal dazuaddiert und das erledigen wir mit einer weiteren „bv-Quelle“. Auch das ist hier gut zu sehen.

Abschließende Anmerkung:
Meist sind bei der Information mehrere Frequenzen oder ein komplettes Frequenzband beteiligt. Deshalb spricht man in diesen Fällen vom ***Unteren und vom Oberen Seitenband, also vom Lower Side Band LSB und Upper Side Band USB*** in einem AM-Signal.

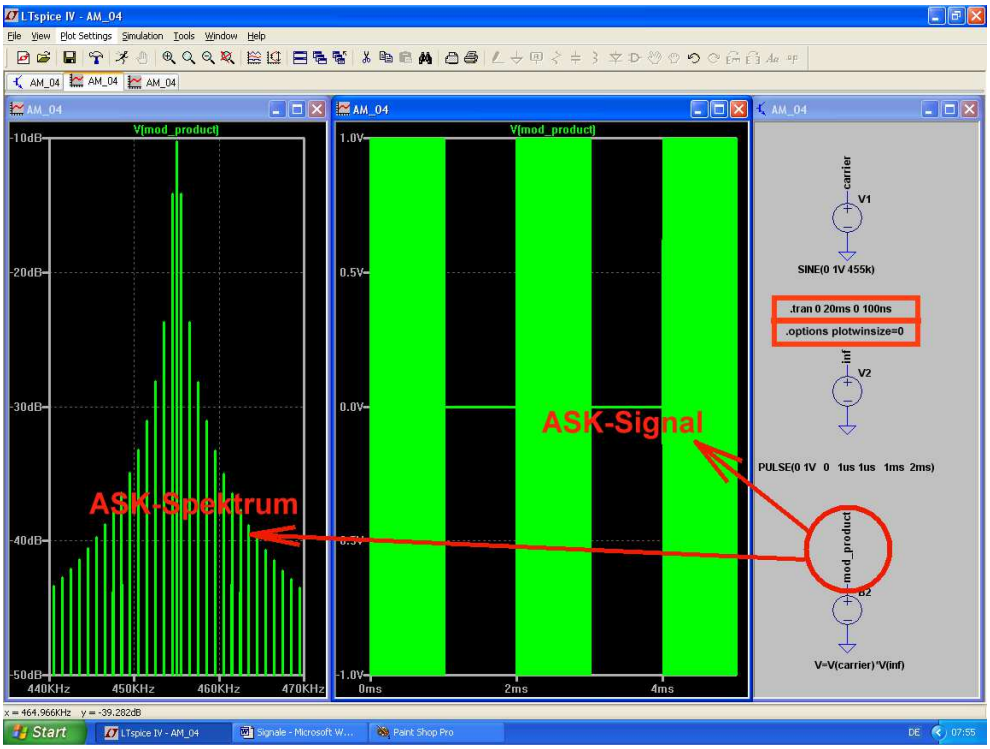
24.2. Amplitude Shift Keying (ASK)

Damit ist nun wirklich die die gute alte Morsetaste mit ihrem „EIN“ und „AUS“ gemeint. Diese so simple Methode hat auch heute noch ihren Platz in der modernen Übertragungstechnik, denn da werden den beiden Zuständen einfach die logischen Pegel Null und Eins zugeordnet. Das funktioniert erstaunlich gut und leidlich störsicher (…wenn man mit zusätzlicher Begrenzung arbeitet). Außerdem hält sich der Hardware-Aufwand in Grenzen.

Bei der Simulation benützen wir wieder die „bv“-Quelle und multiplizieren den Träger mit einem symmetrischen Rechtecksignal von 1kHz als Information. Deren kleinster Spannungswert sei Null Volt, das ergibt „Taste offen“. Die Maximalspannung sei dagegen 1V und dadurch ist „die Taste gedrückt“.



Beim Trägersignal nehmen wir 455kHz (= die AM-Zwischenfrequenz in einem Radio) und bleiben bei einem Spitzenwert von 1V bzw. einem Effektivwert von 0,707V. Das ergibt diese Simulations-Schaltung und die Ausgangssignale der benützten drei Spannungsquellen.



Nun folgt das Spektrum des ASK-Signals und was man da sieht, war zu erwarten. Die symmetrische rechteckförmige Information enthält nämlich außer der Grundwelle noch alle ungeradzahligen Oberwellen von 1kHz. Das Ergebnis sind folglich ein LSB bzw. USB mit all diesen Spektralanteilen der Information, die sich nach links und nach rechts vom Träger ausbreiten.

24.3. Die Frequenzmodulation (FM)

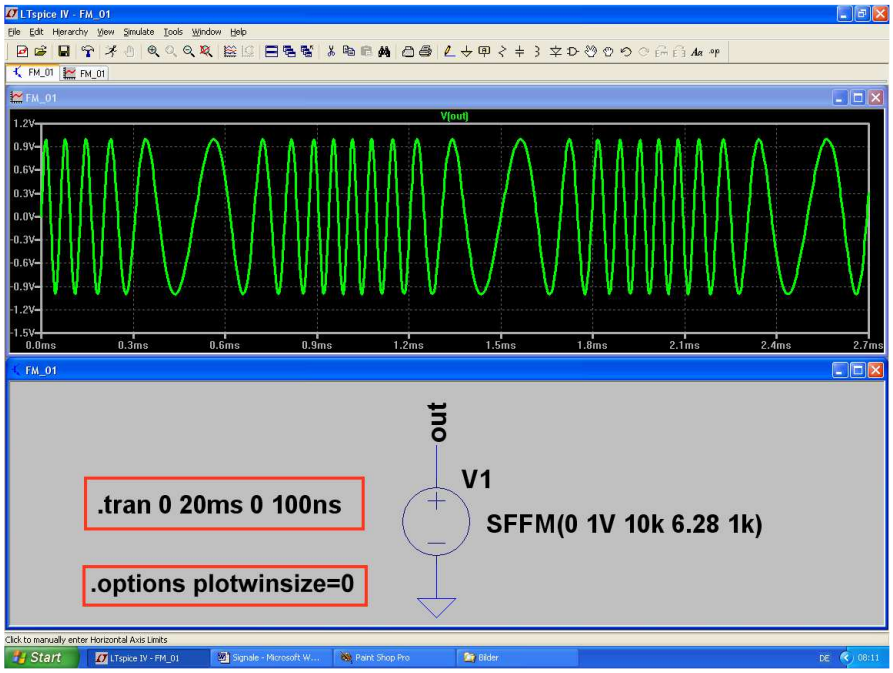
Eine ganz raffinierte Sache, denn man fummelt diesmal nicht an der Amplitude des Trägers herum. Ja, sie ist sogar absolut unwichtig und wird z. B. in einem FM-Empfänger durch Begrenzung von allen Schwankungen und Störspitzen befreit. Dafür ändert man die **Trägerfrequenz im Rhythmus der Informationspannung**. Das kostet zwar deutlich mehr Bandbreite und wesentlich höheren technischen Aufwand, aber der Erfolg ist überzeugend -- man muss nur mal auf UKW gute Musik anhören. Schön ist, dass uns LTspice dafür sogar drei Bauteile zur Verfügung stellt:

a) Man kann bei der schon oft verwendeten Spannungsquelle („voltage“) auf FM-Erzeugung (= „**SFFM**“) umschalten. Dabei wird ein sinusförmiger Träger erzeugt und automatisch eine sinusförmige Information vorausgesetzt.

b) Dann gibt es einen fertigen FM-AM-Generator, den man als „**modulate**“ im Unterordner „Special Functions“ findet. Er eignet sich auch für andere Informations-Kurvenformen.

c) Und endlich gibt des im gleichen Unterordner noch den FM-AM-Generator „**modulate2**“, der Quadraturausgänge besitzt (= zwei getrennte Ausgänge, an denen um 90 Grad verschobene Trägersignale zur Verfügung stehen).

24.3.1. Erzeugung eines FM-Signals mit der Spannungsquelle „voltage“



Dazu sehen wir uns dieses Bild genauer an. Die Frequenz der Trägerschwingung wird bei der positiven Halbwelle der Information höher, bei der negativen Halbwelle dagegen tiefer. Leider lässt sich bei der im Bauteilvorrat bereitgestellten FM-Quelle die modulierende Informationsspannung **NICHT** zusätzlich in einem getrennten Diagramm darstellen. Die Datenkompression ist -- wie immer! -- abgeschaltet. Simuliert wird über 20ms (Auflösung: 100ns), aber zum besseren Verständnis wird nur ein kurzer Abschnitt herausgezoomt.
Den Rest ersieht man aus der Programmierung der SFFM-Quelle. Dort finden wir die Zeile

SFFM(0 1V 10k 6.28 1k)

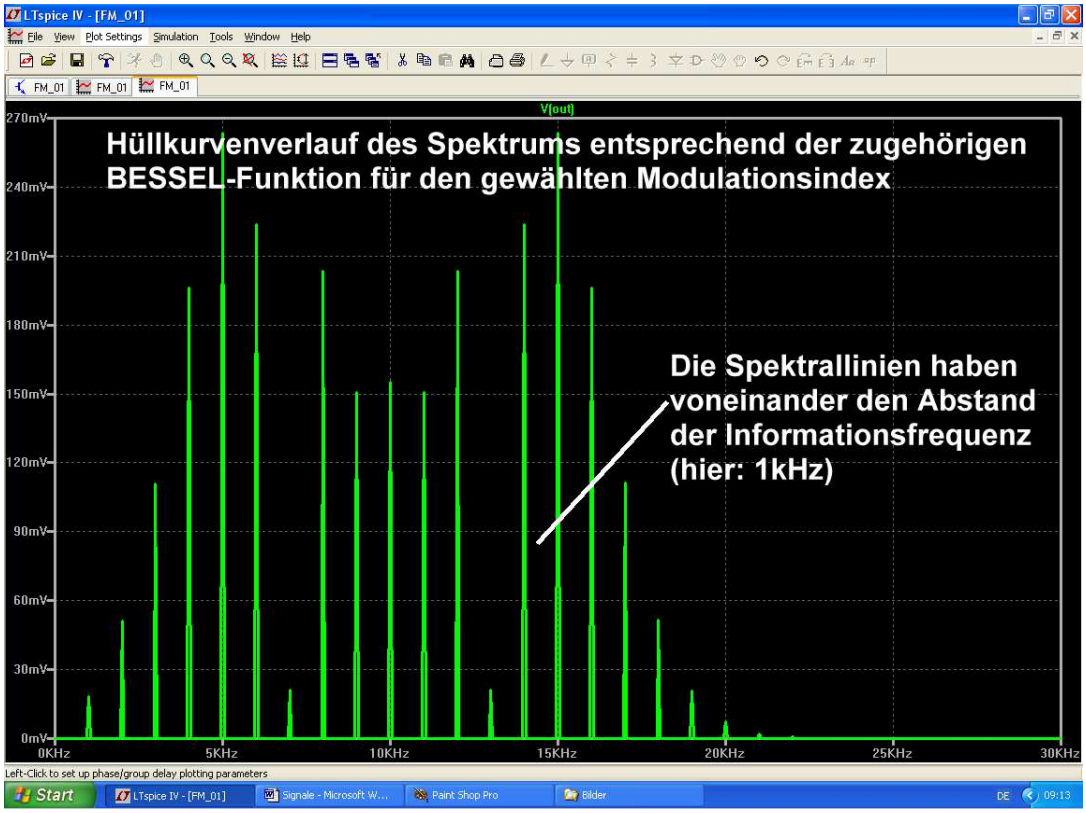
und das bedeutet nacheinander:

0 = kein DC-Anteil
1V = Spitzenwert des Trägers
10kHz = unmodulierte Trägerfrequenz
6,28 = Modulationsindex
1kHz = sinusförmige Information mit f = 1kHz

Da ist natürlich noch ein Wort zum Modulationsindex nötig. Wird die Trägerfrequenz höher, dann kann man das auch als ein Voreilen der Phase gegenüber dem unmodulierten Zustand deuten. Ebenso

bedeutet eine tiefere Augenblicksfrequenz ein Nacheilen der Phasenlage gegenüber dem unmodulierten Trägersignal.
Dieses Vor- und Nacheilen der Phase drückt der Modulationsindex aus. Er stellt nichts anderes dar als die maximale Phasenverschiebung gegenüber dem Ruhezustand und man gibt diese Phasenabweichung im Bogenmaß (also in „Radians“) an. Bei unserem Beispiel steht dort „6,28“ und das ist schlicht und einfach der Umfang eines Kreises mit dem Radius „1“. Also gehören zu einem solchen Vollkreis maximal 360 Grad an Voreilung bzw. Nacheilung! So einfach ist das…

Abschließen wollen wir dieses Kapitel mit der FFT, also der Simulation des zugehörigen Spektrums mit 131 072 Samples.



Da sieht man schon, dass da alles mit Linien belegt ist, die voneinander den Abstand der Informationsfrequenz haben. Bei der Amplitudenverteilung und beim Hüllkurvenverlauf bekommen wir es mit den „Bessel-Funktionen“ zu tun. (Eigentlich nicht ganz so schwer zu verstehen, wenn man sich damit beschäftigt. Die Bandbreite, die das fertige FM-Signal belegt, steigt einfach mit dem Modulationsindex, aber der Hüllkurvenverlauf und die Anzahl der Spektrallinien verändern sich dabei entsprechend einer solchen Funktion). Da gibt es Extremfälle, bei denen sogar beim Steigern des Modulationsindexes der unmodulierte Träger kurz zu Null wird.
Noch ein Wort zum so genannten „**Frequenzhub**“. Er gibt die maximale Abweichung der Augenblicksfrequenz gegenüber dem unmodulierten Zustand an und hängt natürlich mit dem Modulationsindex (für den bisweilen auch die Bezeichnung „Phasenhub“ benützt wird) zusammen. Als Beispiel: beim UKW-Rundfunk im Bereich von 88 bis 108MHz wird mit maximal „+ - 75kHz“ Frequenzhub gearbeitet. Das erscheint sehr wenig, aber es darf nicht mehr sein, da sonst die maximal zulässige Kanalbandbreite (300kHz) überschritten und dann der Sender im Nachbarkanal böse wird. Eine Faustformel sagt dazu:

Erforderliche FM-Kanalbandbreite = 2 x (Frequenzhub + Informationsfrequenz)

Wen es interessiert: aus dem Frequenzhub und der Informationsfrequenz kann man den Phasenhub, also den vorhin diskutierten Modulationsindex in „rad“, leicht ausrechnen:

$$\textbf{Modulationsindex} = \frac{\textbf{Frequenzhub}}{\textbf{Informationsfrequenz}}$$

Sieht man sich diese Formel genauer an, dann erkennt man sofort den Unterschied zwischen den beiden Betriebsarten der **Frequenzmodulation** und der **Phasenmodulation**. Es gilt nämlich:

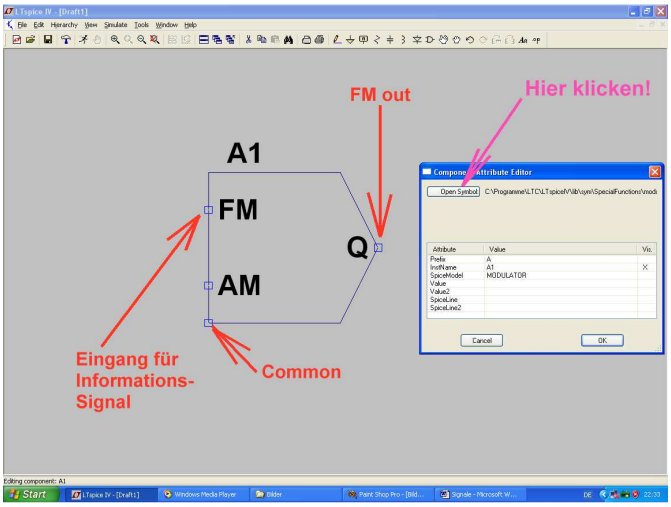
a) Bei der **Frequenzmodulation** hält man den **Frequenzhub konstant**. Nach der obigen Formel **MUSS dann der Modulationsindex und damit der maximale Phasenhub mit steigender Informationsfrequenz ABNEHMEN. Das verschlechtert leider den Signal-Rauschabstand bei hohen Tönen…**

b) Bei der **Phasenmodulation** hält man dagegen den **Modulationsindex = Phasenhub konstant**. Nach obiger Formel wird sich dann der **Frequenzhub mit steigender Informationsfrequenz vergrößern** und das erhöht wiederum die belegte Bandbreite.

24.3.2. FSK = Frequency Shift Keying

Hier handelt es sich um eine digitale Betriebsart (manchmal als Frequenzumtastung bezeichnet), die den beiden Zuständen „log 0“ und „log 1“ zwei unterschiedliche Frequenzen zuordnet. Ihre Wurzeln liegen schon weit zurück, nämlich in der damaligen Fernschreibertechnik. Das einzige, was davon übrig geblieben ist, sind die beiden Bezeichnungen „**mark**“ für die höhere Frequenz und „**space**“ für die tiefere Frequenz. Immerhin brauchen wir aber diese beiden Angaben für die Programmierung des erwähnten **AM-FM-Generators „modulate**“ aus der LTSpice-Vorratskiste!

Dabei ist leider etwas Aufwand erforderlich. Wenn man sich das Bauteil auf den Schirm holt, fehlt nämlich im Normalfall die „Grundeinstellung“ dieses VCO’s (= voltage controlled oscillator). Um das zu ergänzen, klicken wir mit der rechten Maustaste auf das Schaltzeichen und öffnen es damit im „Component Attribute Editor“. Durch einen Klick auf den Button „Open Symbol“ wechseln wir automatisch zum eigentlichen Symboleditor (das ist gut an den kleinen roten Kreisen zu erkennen, die plötzlich an den Enden der einzelnen Linien im Schaltzeichen auftauchen und Fangpunkte darstellen) und können es bearbeiten. Über den Pfad „**Edit / Attributes / Edit Attributes**“ kommen wir an den Symbol Attribute Editor heran und öffnen ihn. In die Zeile „value“ muss jetzt nämlich der Eintrag

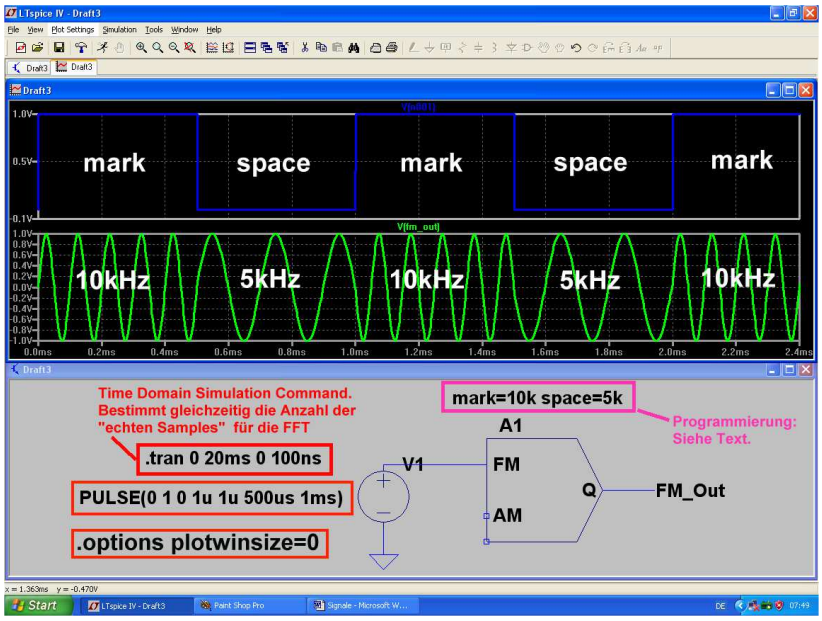


mark=10
space=5k

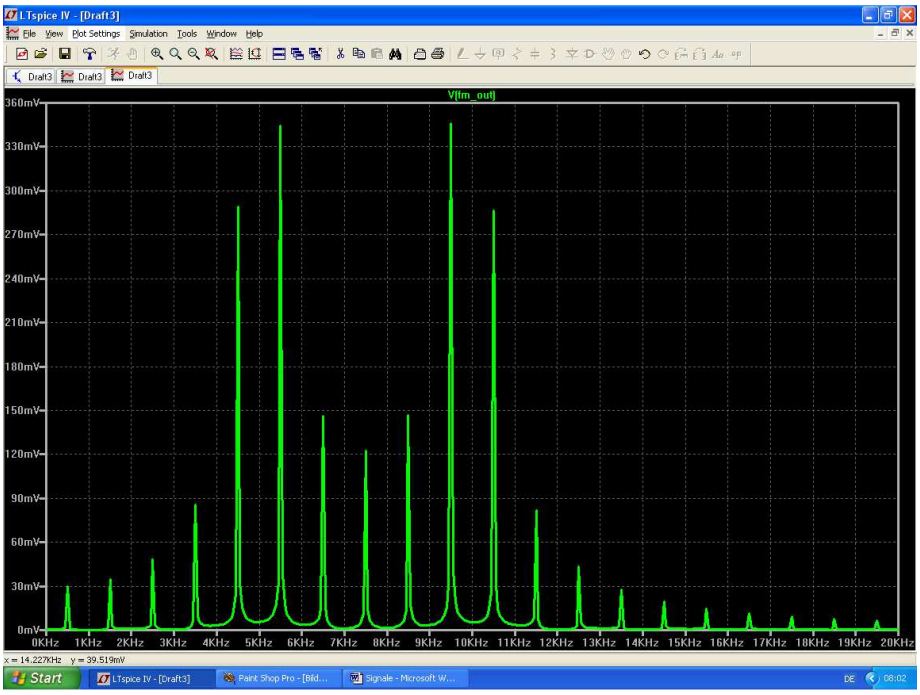
rein, denn so legen wir die Steuerkennlinie des VCO’s fest:

Damit erhalten wir mit einer Spannung von Null Volt am FM-Eingang die Ausgangsfrequenz „space“ von 5kHz.
Mit einer Spannung von 1V an FM dagegen wird die „mark“-Frequenz von 10kHz erzeugt.
(Wem die beiden Werte nicht so gefallen, der darf gern andere nehmen….).

*Wer es noch nie gemacht hat, der tut sich bei diesem Eintrag etwas schwer: man muss nämlich zuerst auf die Zeile „value“ klicken und dann OBERHALB der Tabelle im kleinen Fenster diese Zeile „mark=10k space=5k“ eintippen. Mit OK wird der Eintrag übernommen -- aber er ist nicht auf dem Schirm zu sehen! Dazu dürfen wir nochmals einen Pfad aufrufen und der lautet: „**Edit / Attributes / Attribute Window**“. Klicken wir in dieser Tabelle auf „value“ und OK, dann hängt sofort unsere mark – space - Programmierung am Cursor und kann neben dem Schaltzeichen platziert werden. Jetzt muss nur noch dieses geänderte Symbol am richtigen Ort in der Bauteilbibliothek gespeichert werden (es war der Pfad „**LTspiceIV / lib / sym / Special Functions / modulate**“) und dann haben wir unseren VCO zur Verfügung. Aber erst wenn wir alles geschlossen und neu gestartet und ein neues Projekt angelegt haben, denn sonst weiß das Programm nichts vom geänderten Symbol.*



Jetzt geht es wieder weiter. Da wird eine Pulsspannung mit f = 1 kHz (Umin = 0V / Umax = 1V) an den FM-Eingang angelegt und am Ausgang (der mit dem Label „FM_Out“ versehen wurde) die Spannung simuliert. Außerdem sorgt man (über einen rechten Mausklick auf das Diagramm und „Add Plot Pane / Add Trace“) dafür, dass zusätzlich die Informationsspannung dargestellt wird. Wegen der folgenden FFT wurden die Einstellungen „20 Millisekunden Simulationszeit bei einem Maximum Time Step von 100 Nanosekunden“ gewählt, um 200 000 echte Samples zu erhalten. Und weiterhin wird ohne Datenkompression gearbeitet.



So sieht das mit 131 072 Samples simulierte, linear dargestellte Frequenzspektrum aus. Es weist zwei Maxima auf, nämlich eines im „mark“-Bereich und eines im „space“-Bereich -- eine bekannte Tatsache, die man in der Fachliteratur nachlesen kann. Und es belegt halt doch eine beträchtliche Bandbreite bei einer Übertragung, denn so etwa 20kHz sollten es wohl sein

Das sollte reichen. Aber es folgen (wie in der Schule) noch die Hausaufgaben:

a) Verändern Sie am Informationseingang die minimale und maximale Pulsamplitude. Prüfen Sie, ob auch negative Spannungswerte zulässig sind.

b) Verwenden Sie als Information anstelle des Pulssignals ein Sinussignal mit 1kHz. Wählen Sie die Informationsamplitude so groß, dass exakt die „mark“- und „space“-Grenzfrequenzen erreicht werden und verwenden Sie dazu einen Offset beim Sinus. Vergleichen Sie die jetzt belegte Bandbreite mit den Verhältnissen bei der Pulsübertragung.

c) Benützen Sie als Information eine Dreieckspannung mit derselben Frequenz von 1kHz.

24.4. Quadratur-Amplitudenmodulation (QAM)

Die Quadratur-Amplitudenmodulation ist ein unverzichtbares Werkzeug der modernen Übertragungstechnik geworden. Doch ihr Prinzip und ihre Technik wirken bisweilen etwas geheimnisumwittert. Dieses Kapitel soll -- durch Mitarbeit des interessierten Lesers am PC mit LTspice! - hier Klarheit verschaffen: so kompliziert ist das alles gar nicht!

24.4.1. Grundlegende Informationen

Sehr oft müssen in der Nachrichtentechnik zwei **voneinander völlig unabhängige Informationen gleichzeitig mit derselben Trägerfrequenz im gleichen Kanal** übertragen werden.

Da nimmt man die Mathematik zu Hilfe und erinnert sich daran, dass sich ein Achsenkreuz aus zwei senkrecht zueinander angeordneten Achsen bilden lässt. Ordnet man nun den beiden Signalen, die übermittelt werden sollen, jeweils eine Achse zu, dann ist das plötzlich ganz einfach:

Addiert man nämlich diese beiden Achsanteile zusammen, dann erhält man nur noch ein einziges Gesamtsignal, das man übertragen muss.

Technisch gesehen heißt das, dass wir die beiden verschiedenen Informationen auf zwei getrennte Trägersignale (durch AM) übertragen müssen, die jedoch bei gleicher Frequenz 90 Grad Phasenverschiebung aufweisen. Jeder einzelne Träger ändert dann sowohl seine Amplitude wie auch sein Vorzeichen im Rhythmus der zugehörigen Achseninformation. Anschließend werden diese beiden Trägersignale (Fachausdruck: „I = In phase signal“ und „Q = quadrature phase signal“) einfach zusammenaddiert. **Das ergibt schließlich das „QAM-Signal“.**

Dieses QAM-Signal kann jetzt problemlos als einziger Träger über den vorhandenen Übertragungskanal geschickt werden -- obwohl es eigentlich aus zwei aus völlig verschiedenen Informationen besteht! Dafür ändert sich seine Amplitude und Phase pausenlos und diese beiden Größen muss der Empfänger dauernd auswerten, um wieder an die ursprünglichen Anteile heranzukommen.

Dieser Trick stammt aus der Vektorrechnung und ist seit vielen Jahrhunderten bekannt -- es ist nichts anderes als die Darstellung eines Punktes in einem Achsenkreuz **entweder durch seinen X-Anteil und Y-Anteil ODER durch Betrag und Phase des Zeigers vom Nullpunkt zum gewünschten Punkt im Diagramm.**

In der Nachrichtentechnik wurde er jedoch zum ersten Mal so richtig groß angewandt, als man vom Schwarzweiß-Fernsehen zum Farbfernsehen überging. Da hat man nämlich in denselben Fernsehkanal zusätzlich zum Schwarzweißbild (Y-Signal) die Farbinformationen als „Chroma-Signal“ dazugepackt (…nahe an der oberen Frequenzgrenze des Y-Signals von 5MHz, nämlich in Europa bei 4,43MHz, damit sich die beiden Signale möglichst wenig beißen).
Und dieses Chromasignal selbst besteht aus der „Blau“-Information (korrekt: B-Y) bei der waagrechten Achse bzw. aus der „Rot“-Information (korrekt: R-Y) bei der senkrechten Achse. Da ein Farbbild aber aus den drei Anteilen „Rot, Grün und Blau“ zusammengesetzt wird, scheint noch im Empfänger die Grün-Information zu fehlen. Das ist aber nicht so, denn das immer mit ausgestrahlte Schwarzweißbild besteht aus den drei Anteilen Rot, Grün und Blau in einem **exakt bekannten Verhältnis**. Gewinnt man nun Rot und Blau durch „Quadraturdemodulation“ des Chromasignals zurück, dann ist der Rest kinderleicht:

Vom Schwarzweißbild „Y“ muss man nur die beiden jetzt exakt bekannten Anteile von Rot und Blau abziehen und dann wird „Grün“ übrig bleiben!

Die eigentlichen Probleme hat jedoch der Empfänger. Er muss nämlich pausenlos mit ausreichenden Informationen über die „Bezugsphase“ versorgt werden…..klingt kompliziert, ist es aber nicht. Wir brauchen nämlich im Empfänger genau dasselbe „**stillstehende Achsenkreuz des Senders**“, in das wir den Augenblickswert des ankommenden QAM-Signals als Betrag und Phase eintragen und anschließend die Werte von X-Anteil und Y-Anteil an den beiden Achsen ablesen.
Beim Analogen Farbfernsehen half man sich durch Ausstrahlung von 10…12 Schwingungen mit der Phasenlage Null Grad -- also der „Waagrechten Achse des Sender-Achsenkreuzes“ -- bei jedem Zeilenrücklauf. Da der Bildschirm bei diesem Rücklaufvorgang dunkel getastet wird, stört das nicht.

Aber mit diesem Referenzsignal (= „Burst“) synchronisiert man einen Quarzoszillator und erzeugt durch eine passende Zusatzschaltung zwei um 90 Grad phasenverschobene Referenzträger für die QAM-Demodulation zur Wiedergewinnung von „I – Anteil“ und „Q – Anteil“.

Bei Digitalübertragungen ist die Sache mit der Bezugsphase deutlich schwieriger. Z. B. hat man sich bei Vielkanal-Übertragungen schon so geholfen, dass in einem „Referenzkanal“ dauernd die „Bezugsfrequenz samt Phase“ in Form einer einfachen und überschaubaren, aber exakt festgelegten Bitfolge übertragen und vom Empfänger ausgewertet wird. Es gibt natürlich noch weitere Möglichkeiten, um ohne diesen zusätzlichen Referenzkanal auszukommen -- aber das sind komplexere Themen für Fortgeschrittene und so weit wollen wir heute nicht gehen.

24.4.2. Erzeugung eines QAM-Signals

Wir stellen uns folgende Aufgabe:
Übertragen Sie zwei Informationen durch QAM eines 20kHz-Trägers.

Dabei gelten folgende Daten:

Signal A (im I-Kanal) ist ein Sinus mit f = 1000Hz und einem Spitzenwert von 0,5V.

Signal B (im Q-Kanal) ist ein Rechteck mit f =500Hz, einem Minimalwert von 0V und einem Maximalwert von 1V.

Die beiden Trägerkomponenten (= I bzw. Q) haben Rechteckform. Ihre Frequenz beträgt f = 20kHz, der Minimalwert sei -1V, der Maximalwert dagegen +1V. Das I-Signal eilt dem Q-Signal um 90 Grad nach.

Fangen wir mit der Programmierung der Spannungsquellen für die **Trägerkomponenten** in LTspice an.
Verwendet werden zwei symmetrische Rechtecksignale mit derselben Frequenz (f = 20kHz) und einer Amplitude von 1V (kein DC-Anteil), aber einem Phasenunterschied von 90 Grad. Anstiegs- und Abfallzeit seien jeweils 50 Nanosekunden.

Achtung:
wir können in SPICE keine Voreilung „in Grad“ programmieren. Deshalb lassen wir das I - Signal durch einen Delay-Eintrag von 12,5 Mikrosekunden bei f = 20kHz um 90 Grad nacheilen.

Das ergibt nun folgende Quellenprogrammierung beim Q-Signal:

PULSE(-1 1 0 50ns 50n 24.95us 50us)

Und so sieht es dann beim I-Signal aus:

PULSE(-1 1 12.5us 50ns 50ns 24.95us 50us)

Simuliert wird über **t = 20ms mit einem maximalen Timestep von 20ns**. Dann wird der Button mit dem Hämmerchen gedrückt, auf die Karteikarte „SPICE“ gewechselt und dort „Gear“ unter „Default Integration Method“ aktiviert (...erhöht die Stabilität der Simulation).

Das gibt (mit t = 20ms) eine **Frequenzauflösung von 1 / 20ms = 50Hz,** wenn über die FFT das Spektrum betrachtet werden soll. Der maximale **Timestep von 20ns** stellt sicher, dass das Simulationsprogramm die Anstiegs- und Abfallzeit von 50ns bei diesen Pulssignalen nicht übersehen kann.

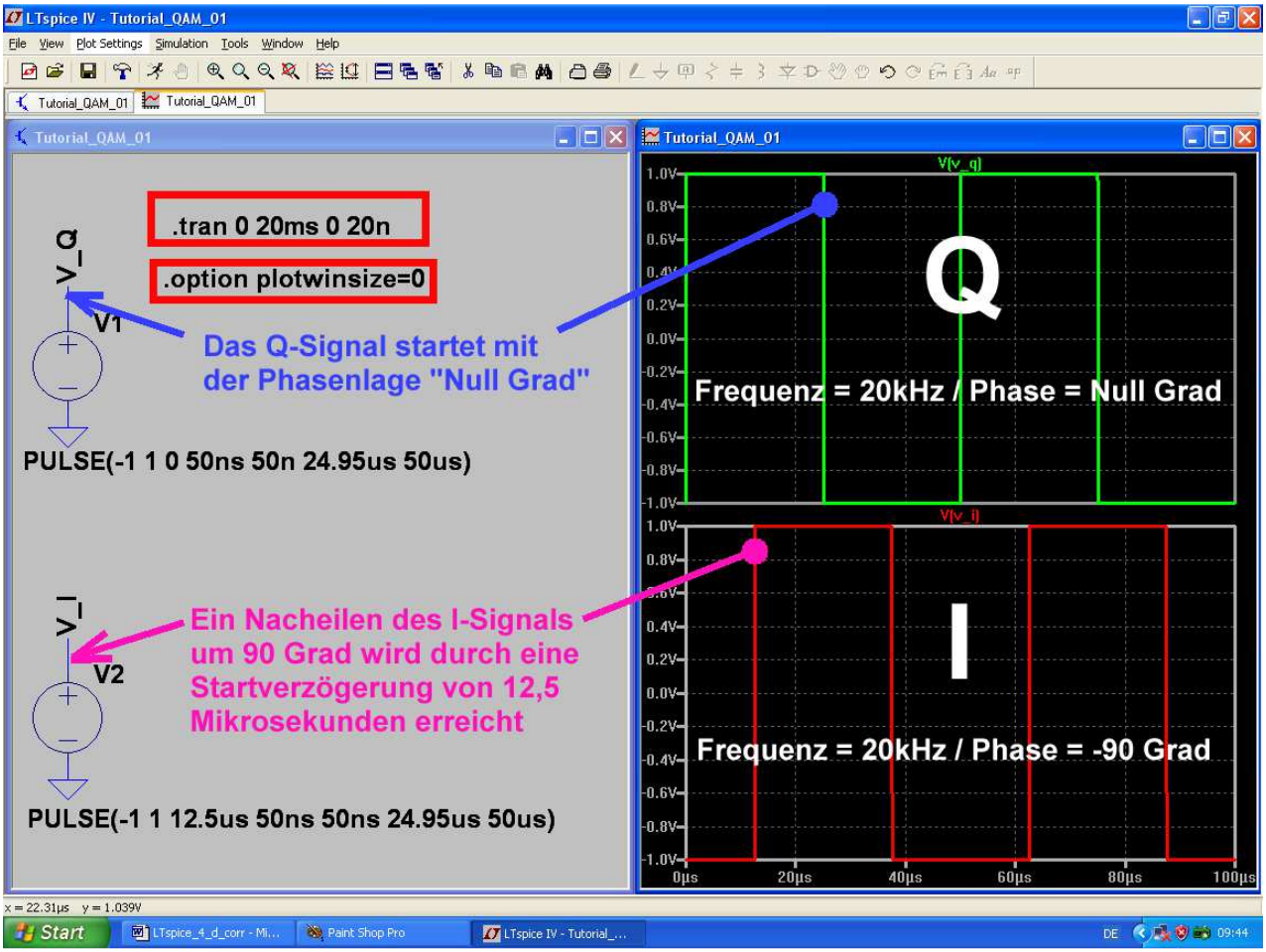
Zusätzliche wird die Datenkompression mit der SPICE-Direktive „**option plotwinsize=0**“ abgeschaltet. Damit steht garantiert eine echte Sample-Anzahl von

20ms / 20ns = 1 000 000 Samples

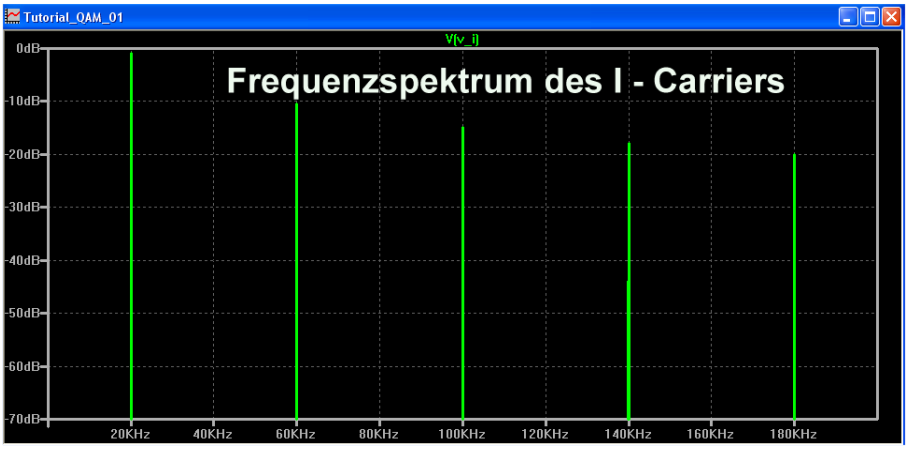
für die FFT zur Verfügung.

Wer sich über die „**24,95µs“ statt „25µs“** bei der Pulslänge wundert, der möge bedenken:
Die Pulslänge ist tatsächlich nur die Zeit, in der das Signal den positiven Höchstwert aufweist. Folglich müssen wir die Anstiegszeit abziehen, wenn das Signal symmetrisch bleiben soll. Eine Symmetriekontrolle ist schnell und einfach möglich, wenn wir anschließend für eines der beiden Signale eine FFT mit 524 288 Samples durchführen (…das ist erlaubt, denn es stehen 20ms / 20ns = 1000 000 echte simulierte Samples zur Verfügung und möglichst viele Samples bei der FFT ergeben einen größeren dargestellten Frequenz- und Dynamikbereich) und den Oberwellenanteil prüfen.

Und so sieht die Simulation dann aus:

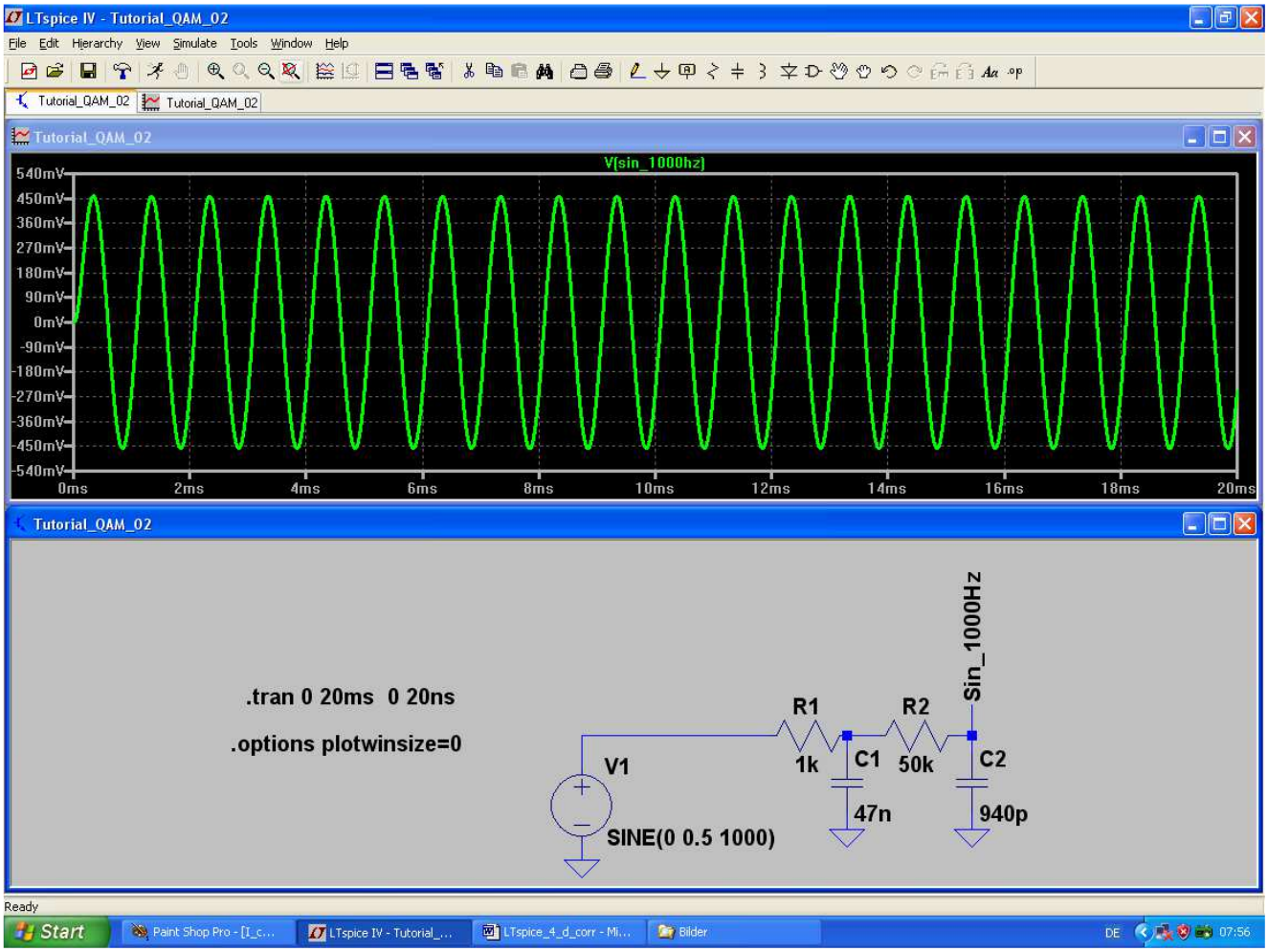


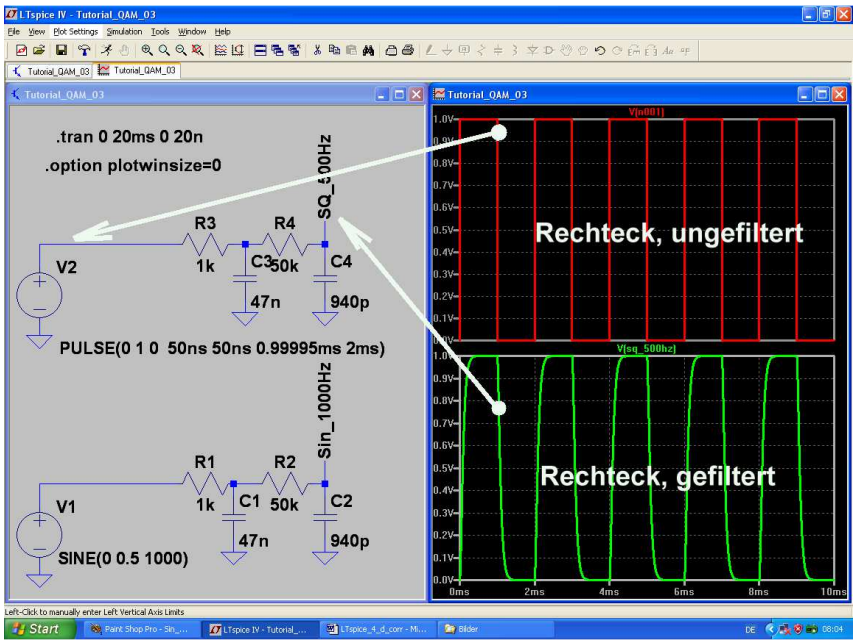
Dazu das Spektrum nach einer FFT mit 524288 Samples (Anleitung: mit der rechten Maustaste auf den Spannungsverlauf klicken, dann „View“ und „FFT“ wählen).



Man sieht, dass nur die ungeradzahligen Harmonischen auftreten und alles Unerwünschte um mehr als 70db gedämpft ist. Das ist ein sicheres Zeichen für nahezu perfekte Symmetrie des Rechtecksignals.

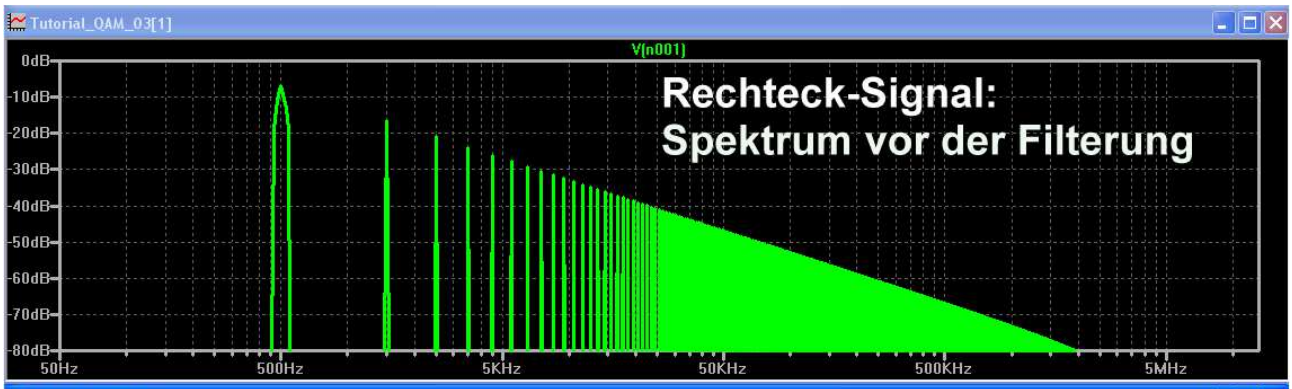
Jetzt programmieren wir die „Basisband-Signale“. Dazu setzen wir zwei weitere Spannungsquellen ein und schalten hinter jede einen einfachen Tiefpass mit einer Grenzfrequenz von ca. 3,4 kHz zur „Bandbegrenzung“ (…das entspricht der oberen Grenzfrequenz eines Telefonkanals). So vermeiden wir Probleme durch Frequenzband-Überschneidungen (Aliasing) bei der nachfolgenden Trägermodulation. **Beginnen wir mit dem Sinussignal (f = 1000Hz / Spitzenwert = 0,5V) im I – Kanal:**

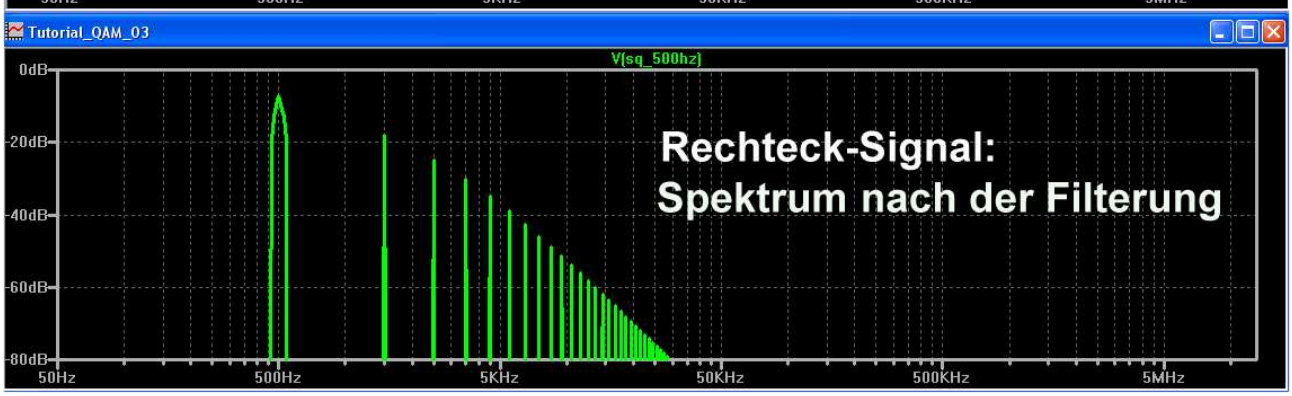




Im geichen Simulationsschaltbild fügen wir nun die **Rechteck-Spannungsquelle für den Q – Kanal** dazu. Auch hier müssen wir wieder die Anstiegszeit von der Pulslänge abziehen, um perfekte Symmetrie zu erhalten.

Interessant ist nun die Wirkung des Filters (= Tiefpass am Ausgang) auf die Oberwellen beim Rechtecksignal durch einen Vergleich der Spektren. Immerhin sind nach dieser Aktion die Signalanteile oberhalb von 5kHz bereits um mehr als 40dB gedämpft:





Auf dem nächsten Blatt folgt die vollständige QAM - Modulatorschaltung:

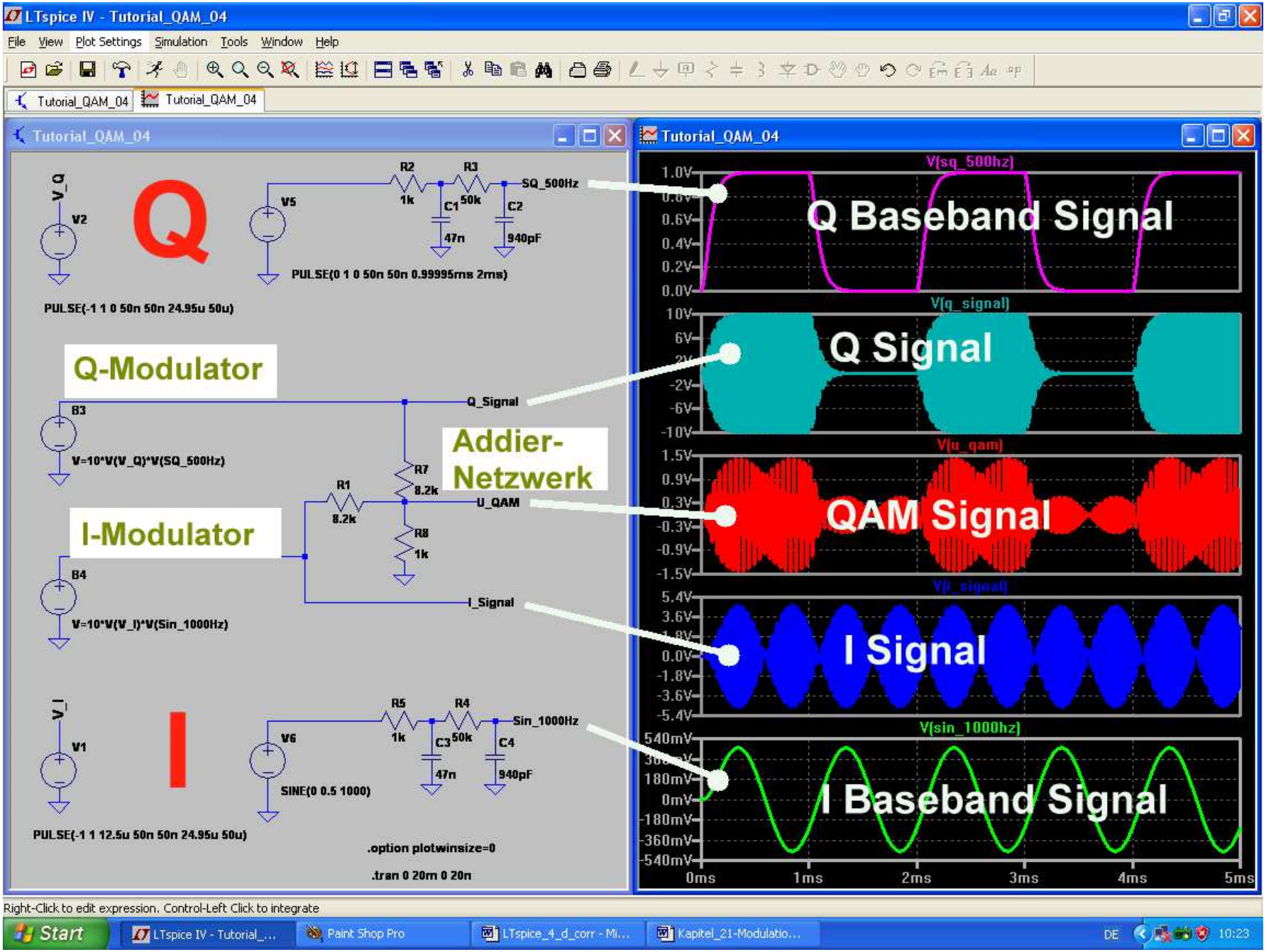
Erläuterungen:
Die „bv“ – Spannungsquelle im Bauteilvorrat von LTspice stellt einen idealen Analog-Multiplizierer dar, mit der sich jede Trägerkomponente und ihr Basisband-Signal perfekt multiplizieren lassen. Damit verwirklicht man bei jeder Diagrammachse die gewünschte **Übertragung des Basisband-Signalverlaufs auf die Trägerfrequenz.** Wie gewünscht, kehrt sich dabei durch diese Multiplizierschaltung auch die Phasenlage des Trägers um, wenn das Basisband gerade negativ ist).

Zur **Addition von „I“- und „Q“- Signal** dient ein einfaches Widerstandsnetzwerk hinter jedem Modulator, das allerdings die Einzelamplituden um den Faktor 10 abschwächt. **Deshalb wird in jedem Multiplizierer noch zusätzlich eine zehnfache Verstärkung vorgesehen.**

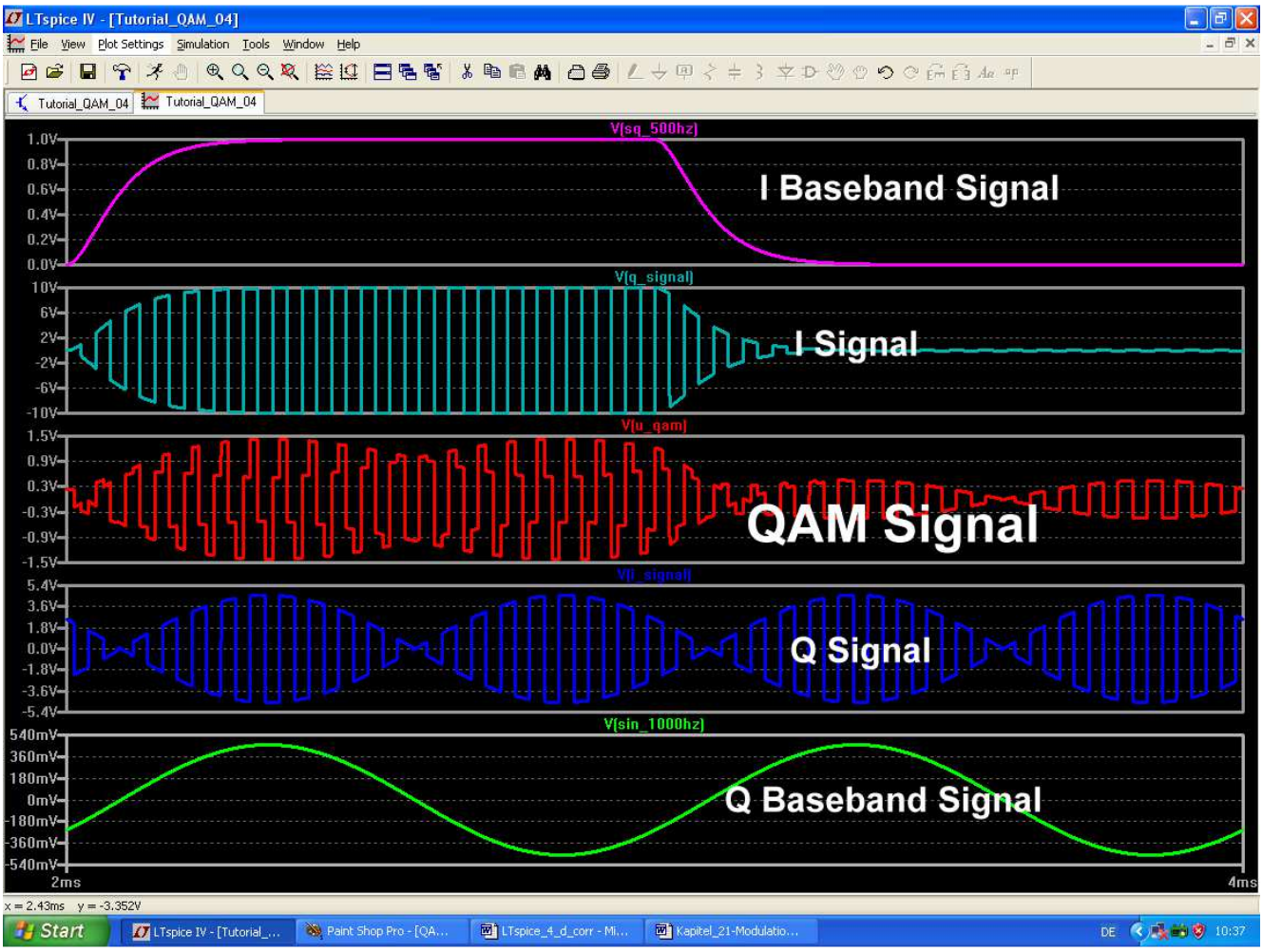
Die erforderlichen Formel-Eingaben bei den beiden Multiplizier-Stufen lauten deshalb

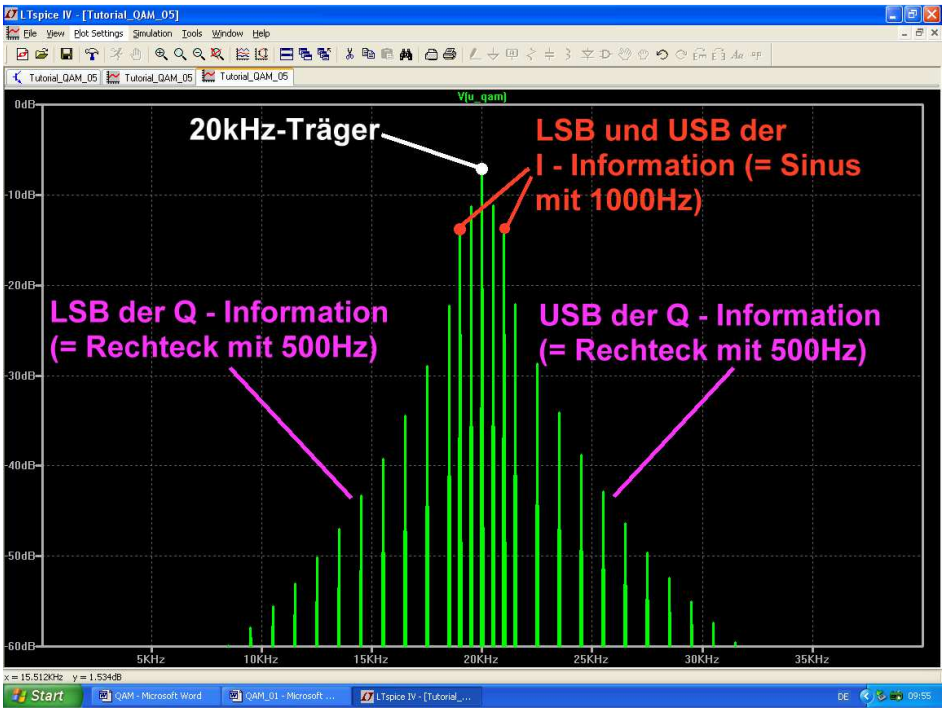
für den „I“ – Pfad: **V = 10 * V(V_I) * V(Sin_1000Hz)**

und für den “Q” – Pfad: **V = 10 * V(V_Q) * V(SQ_500Hz)**


LTspice IV - Tutorial_QAM_04
File View Plot Settings Simulation Tools Window Help
Tutorial_QAM_04
Q
V_Q
V2
V5
R2 1k
R3 50k
C1 47n
C2 940pF
SQ_500Hz
PULSE(0 1 0 50n 50n 0.99995ms 2ms)
PULSE(-1 1 0 50n 50n 24.95u 50u)
Q-Modulator
B3
V=10*V(V_Q)*V(SQ_500Hz)
Q_Signal
Addier-Netzwerk
R7 8.2k
R1 8.2k
U_QAM
R8 1k
I-Modulator
B4
V=10*V(V_I)*V(Sin_1000Hz)
I_Signal
I
V_I
V1
V6
R5 1k
R4 50k
C3 47n
C4 940pF
Sin_1000Hz
SINE(0 0.5 1000)
PULSE(-1 1 12.5u 50n 50n 24.95u 50u)
.option plotwinsize=0
.tran 0 20m 0 20n
V(sq_500hz)
Q Baseband Signal
V(q_signal)
Q Signal
V(u_qam)
QAM Signal
V(i_signal)
I Signal
V(sin_1000hz)
I Baseband Signal
0ms 1ms 2ms 3ms 4ms 5ms
Right-Click to edit expression. Control-Left Click to integrate
Start
LTspice IV - Tutorial_...
Paint Shop Pro
LTspice_4_d_corr - Mi...
Kapitel_21-Modulatio...
DE
10:23

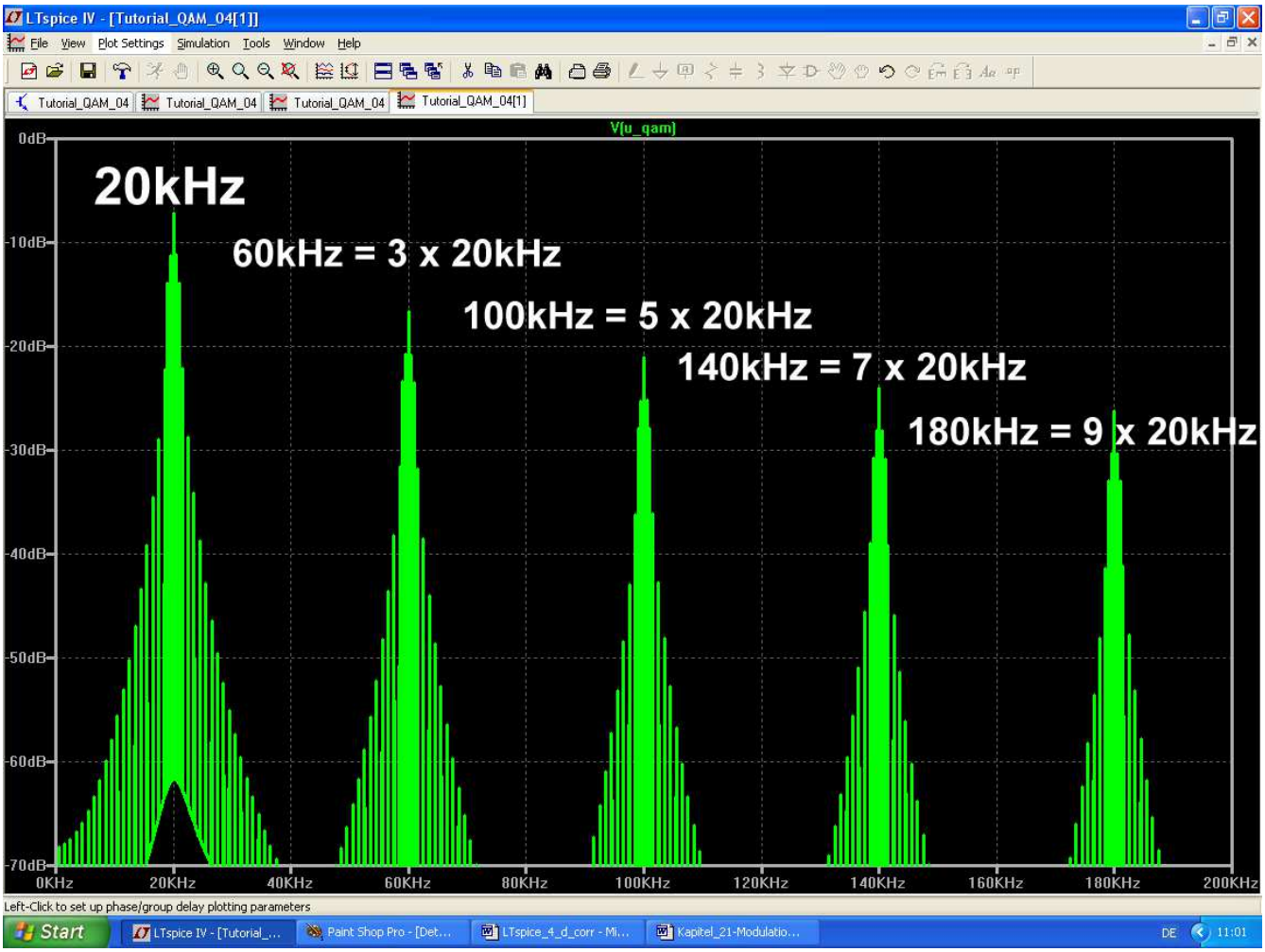
Interessant ist ein Blick auf die Details nach starkem Zoomen der Diagramme. Bei beiden Achssignalen (I–Signal und Q-Signal) ist der Phasenwechsel im Rhythmus der Trägerfrequenz von 20kHz gut zu sehen. Und nach dem Zusammenaddieren der beiden Komponenten wird im QAM-Signal deutlich auch deren Phasenunterschied von 90 Grad sichtbar:





Das Spektrum um 20kHz herum sieht dann so aus.

Da es sich um rechteckförmige Träger-Komponenten handelt, muss sich das Spiel bei jeder ungeraden Oberwelle der Trägerfrequenz wiederholen:



Da werden sich wohl die Nachbarkanäle im System etwas beklagen…

24.4.3. Filterung des QAM-Signals zur Anpassung an die Übertragungs-Kanalbandbreite

Wie man beim obigen Spektrum leicht sieht, eignet sich das QAM-Signal in dieser Form noch nicht zur Übertragung, denn oft werden viele aneinander grenzende Kanäle benötigt (Prinzip von FDM = frequency division multiplex). Die dürfen sich natürlich nicht gegenseitig stören und deshalb muss jeder Kanal durch einen geeigneten steilflankigen Bandpass vom Nachbarn „isoliert“ und von seinen Harmonischen befreit werden. Also lassen wir hinter dem Addiernetzwerk zur QAM-Erzeugung erst einen nichtumkehrenden Operationsverstärker mit zweifacher Verstärkung folgen. Er wird mit einem zusätzlichen Innenwiderstand von 50Ω versehen und kann damit einen Bandpass mit einer **Bandbreite von 6,8kHz und Ri = Ra = 50Ω** speisen.

Der Bandpass-Entwurf ist mit den heutigen Programmen aus dem Internet ein Kinderspiel und eine Freude. Die Auswahl ist groß und eine der besten Homepages ist

http://www.siversima.com/rf-calculator/bandpass-filter-designer/

Aber hier wurde noch mit einem alten (= aber sehr geliebten) DOS – Programm namens „fds.exe“ gearbeitet. Das sind die Eingaben:

Bandpass, Tschebyschef –Typ

Filtergrad = Zahl der Pole = n = 3

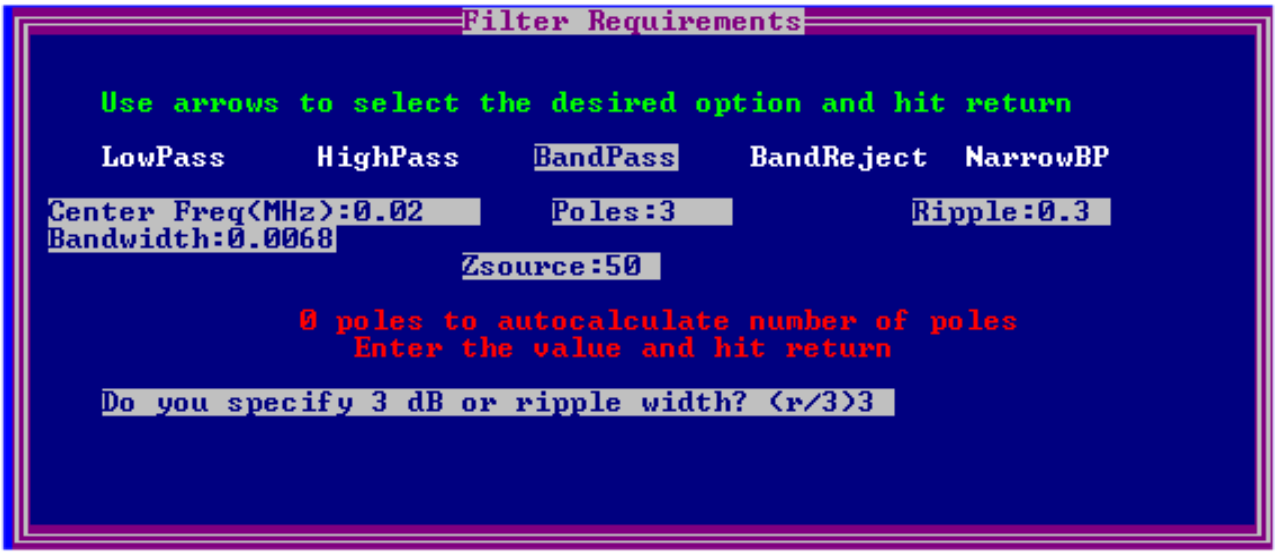
$Z_{in} = Z_{out} = 50\Omega$

Ripple = 0,3dB

Mittenfrequenz = 20kHz

3dB – Bandbreite = 6,8kHz

Dies ist die fertig ausgefüllte Eingabemaske…..



…..und dies das Calculator-Ergebnis:

```
BP/BR filter
The filter values are
794453.632 pf in a PLC with 79709.800 nH  par
1648062.221 nH in a SLC with 38424.362 pF ser
794453.632 pf in a PLC with 79709.800 nH  par
(S)ave to file (P)rint or (C)ont?
```

„**PLC**“ bedeutet „**parallel LC resonant circuit**“ und „**SLC**“ natürlich „**series LC resonant circuit**“…

Nun wird zunächst **in LTspice eine Testschaltung aufgebaut** und die Filterkurve samt Verstärkung geprüft. Sie findet sich auf dem nächsten Blatt und folgende Dinge bedürfen einer Erklärung:

a) **.option plotwinsize=0** bedeutet, dass die Datenkompression abgeschaltet ist.

b) **.ac dec 501 1 100k** ergibt einen dekadischen AC-Sweep von 1Hz bis 100kHz und 501Punkten pro Dekade

c) Für diesen AC-Sweep muss bei der Spannungsquelle V1 die **Amplitude auf „1“ und die Phase auf „0“** gestellt sein (= Eintrag „**AC 1 0**“ bei den properties der Spannungsquelle)

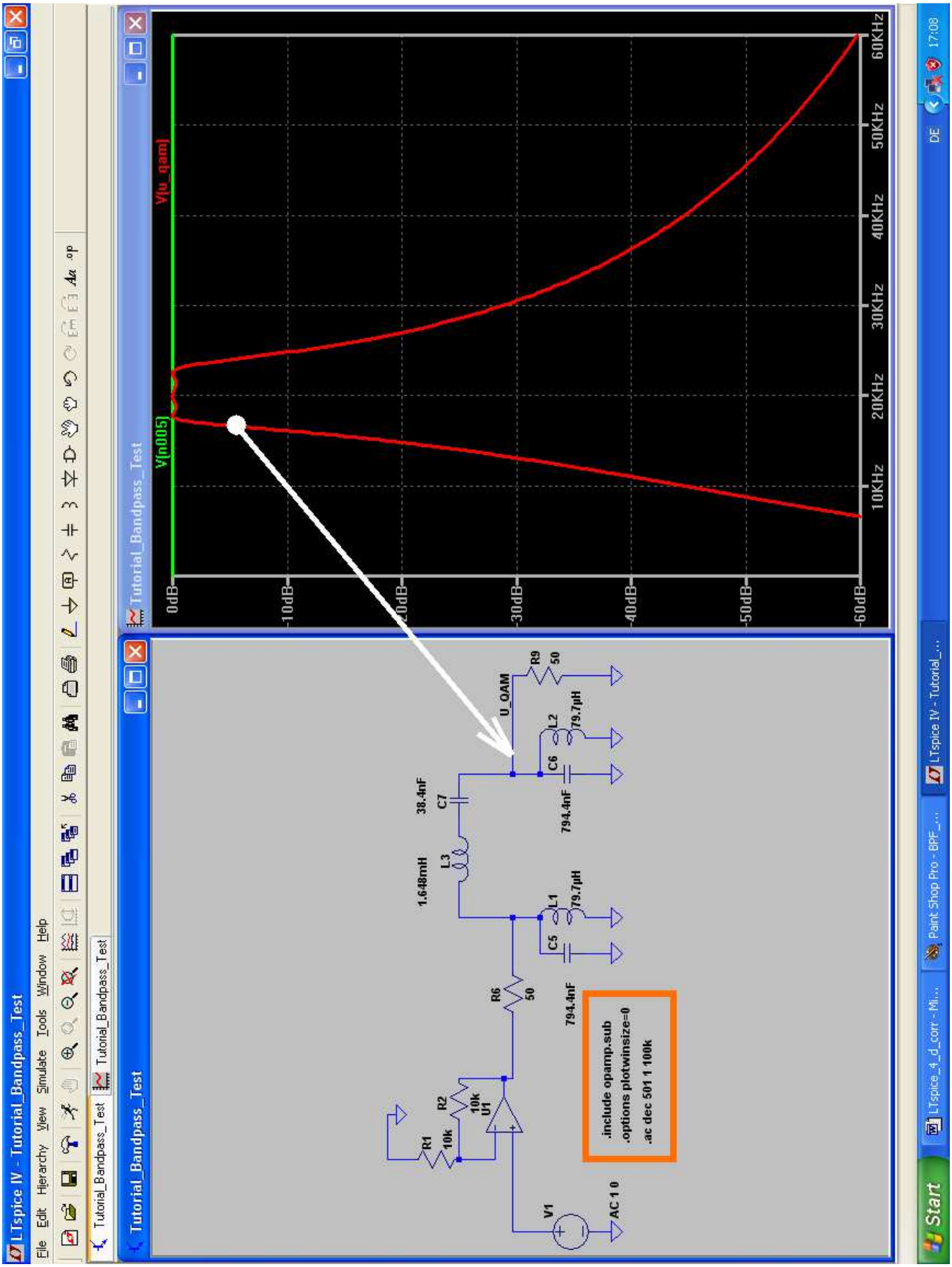
d) Als Operationsverstärker holen wir uns aus der Bibliothek und dem Unterordner „Opamps“ das Bauteil „**opamp**“

e) Diesem OPV muss jedoch ein Spice-Modell zur Verfügung gestellt werden. Das erfolgt durch die Direktive **„.include opamp.sub**“.

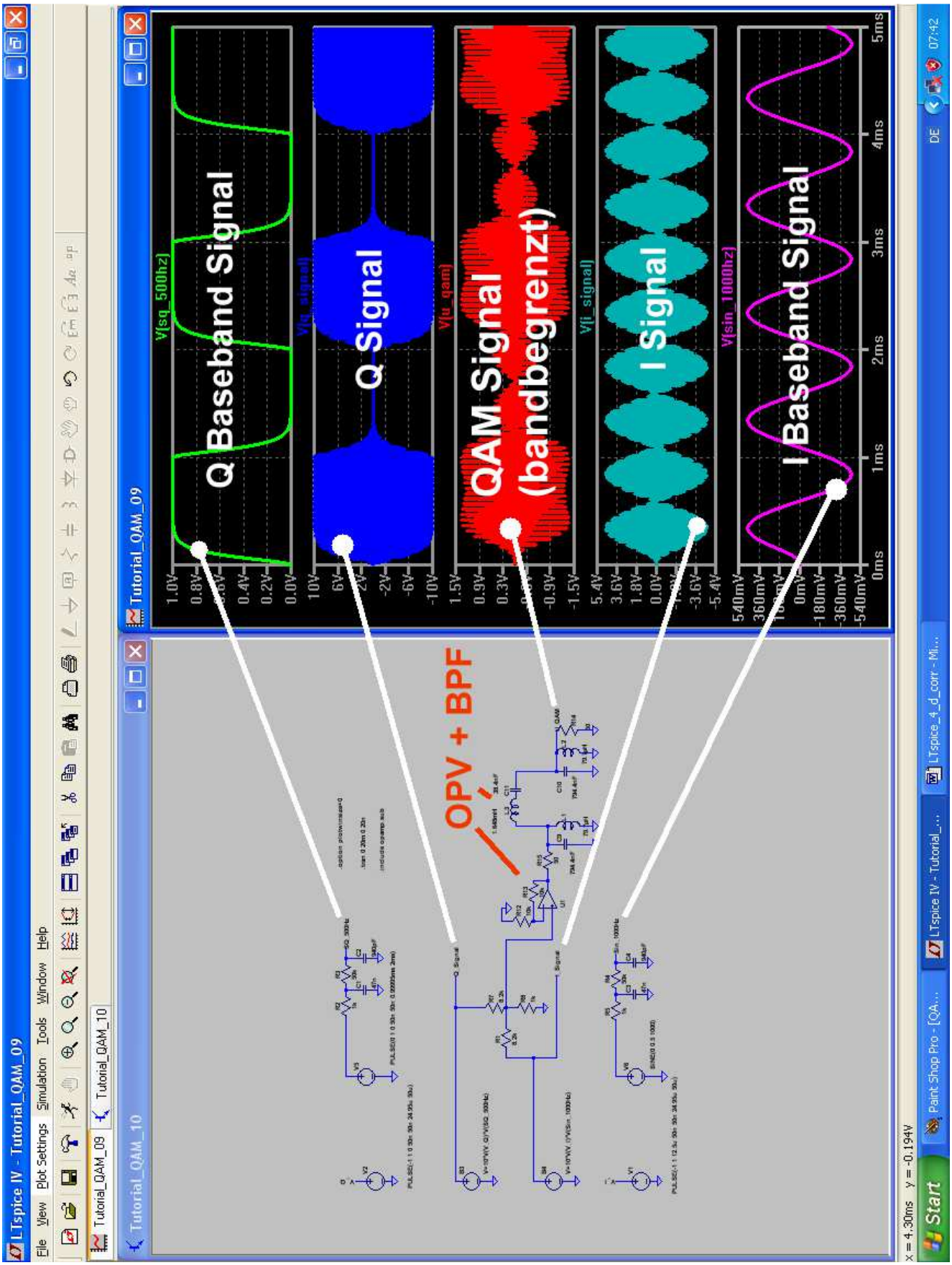
f) Die Ausgangsspannung am Lastwiderstand von 50Ω ist (wegen der **Leistungsanpassung** beim Filter) um 6dB = Faktor 2 gegenüber der Ausgangsspannung des OPVs gedämpft. Deshalb wird der **Operationsverstärker in nichtumkehrender Schaltung mit der Verstärkung V = 2** betrieben, damit am Filterausgang wieder derselbe Pegel herrscht wie am Eingang des OPVs.

Und nach einem Klick auf den „Button mit dem Hammer in der Menüleiste“ kontrolliert man nochmals, ob wirklich unter „SPICE“ die **Integrationsmethode „Gear“** eingestellt ist.

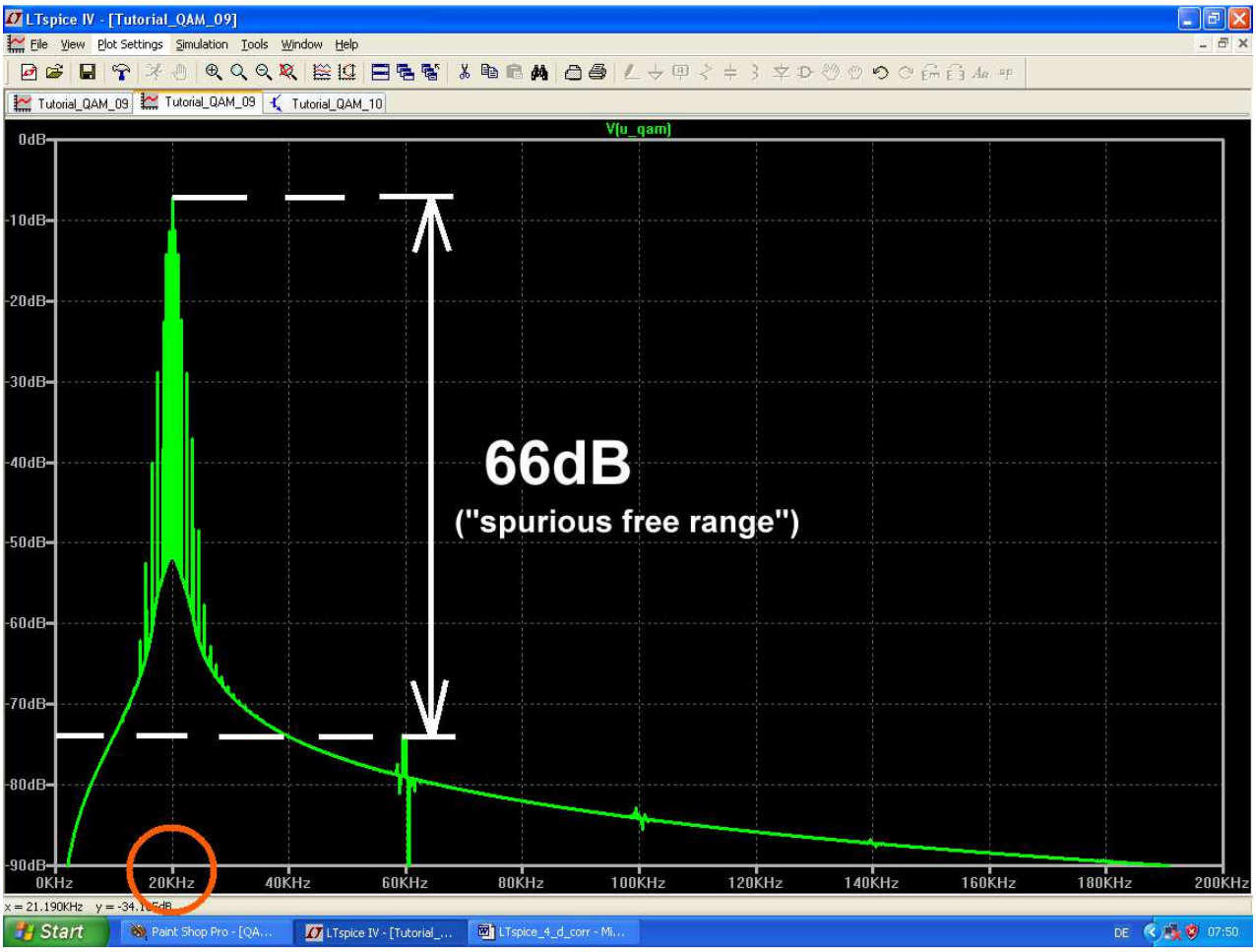
Das nächste Blatt zeigt das Ergebnis des Filtertests.


LTspice IV - Tutorial_Bandpass_Test
File Edit Hierarchy View Simulate Tools Window Help
Tutorial_Bandpass_Test
Tutorial_Bandpass_Test
V(n005)
V(u_qam)
0dB
-10dB
-20dB
-30dB
-40dB
-50dB
-60dB
10KHz
20KHz
30KHz
40KHz
50KHz
60KHz
R1
10k
R2
10k
U1
R6
50
V1
AC 1 0
1.648mH
L3
38.4nF
C7
U_QAM
C5
L1
79.7µH
794.4nF
C6
L2
79.7µH
794.4nF
R9
50
.include opamp.sub
.options plotwinsize=0
.ac dec 501 1 100k
Start
LTspice_4_d_corr - Mi...
Paint Shop Pro - BPF_...
LTspice IV - Tutorial_...
DE
17:08

Jetzt folgt wieder ein Stück Arbeit, denn der OPV samt Bandpass muss in unser „großes Schaltbild zur QAM-Erzeugung“ eingefügt werden.



Interessant ist nun noch ein Blick auf das Spektrum des „bandbegrenzten QAM Signals“:



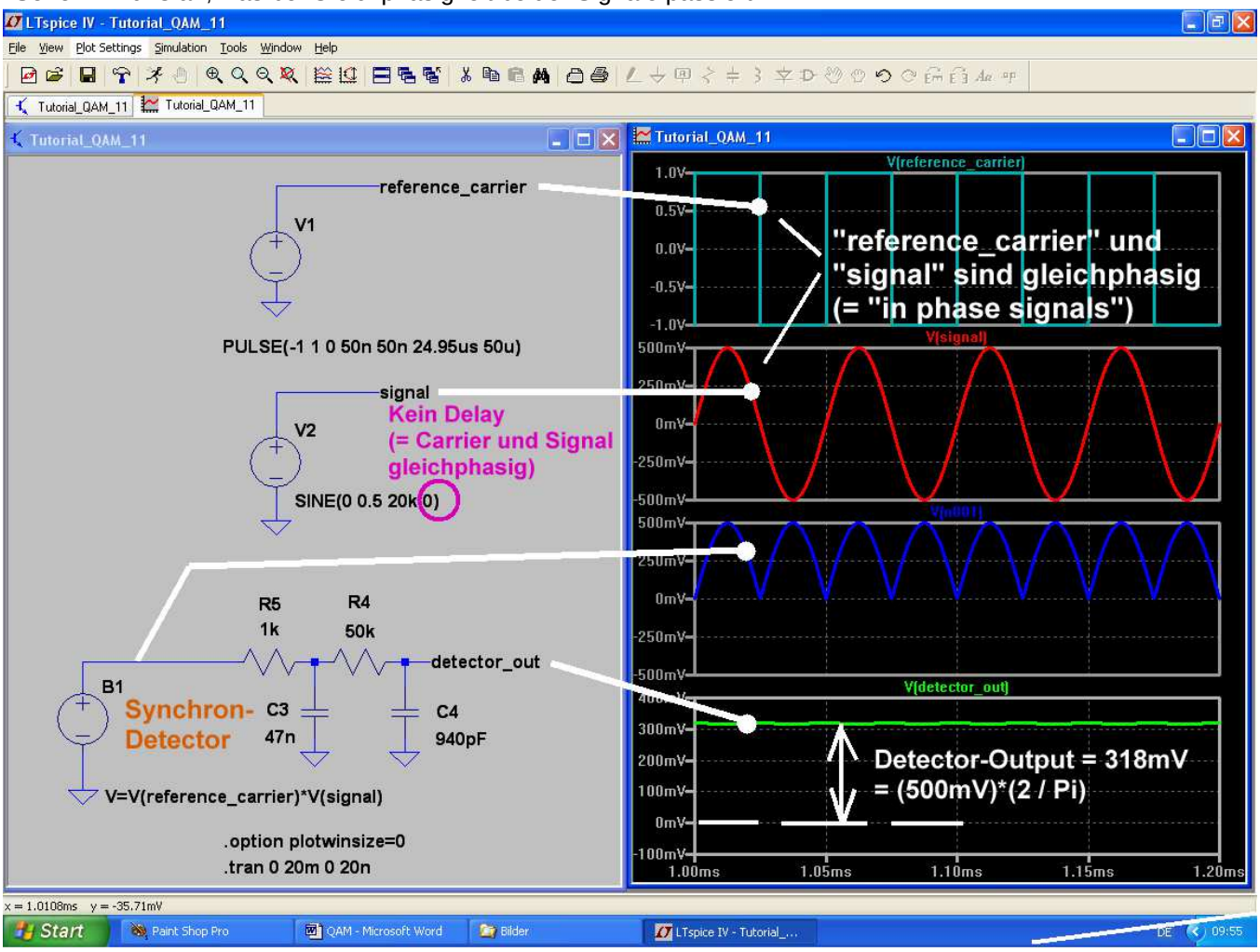
Ab jetzt können sich die Nachbarkanäle sicher nicht mehr beklagen….

24.4.4. Demodulation des QAM-Signals

24.4.4.1. Prinzip der Synchron-Demodulation

Auf der Empfangsseite müssen wir die beiden übertragenen Signale wieder trennen und dazu brauchen wir erneut das „Achsenkreuz“, wie es auf der Senderseite benützt wurde. Wie schon erwähnt, ist diese Maßnahme die erste und allerwichtigste Aufgabe im Empfänger, aber das ist ein eigenes, aufwendiges und komplexes Gebiet.
Deshalb setzen wir einfach voraus, dass dieses Problem gelöst ist und uns die beiden „Carrier-Signale“ für die Achsen mit einem exakten Phasenunterschied von 90 Grad zur Verfügung stehen. Zur Wiedergewinnung der beiden Einzel-Signale „I“ und „Q“ sind wieder zwei Produktdetektoren erforderlich. Ihnen wird gemeinsam das empfangene QAM-Signal zugeführt, aber jeder Detektor wird zusätzlich mit seinem zugehörigen Carrier-Signal gespeist.

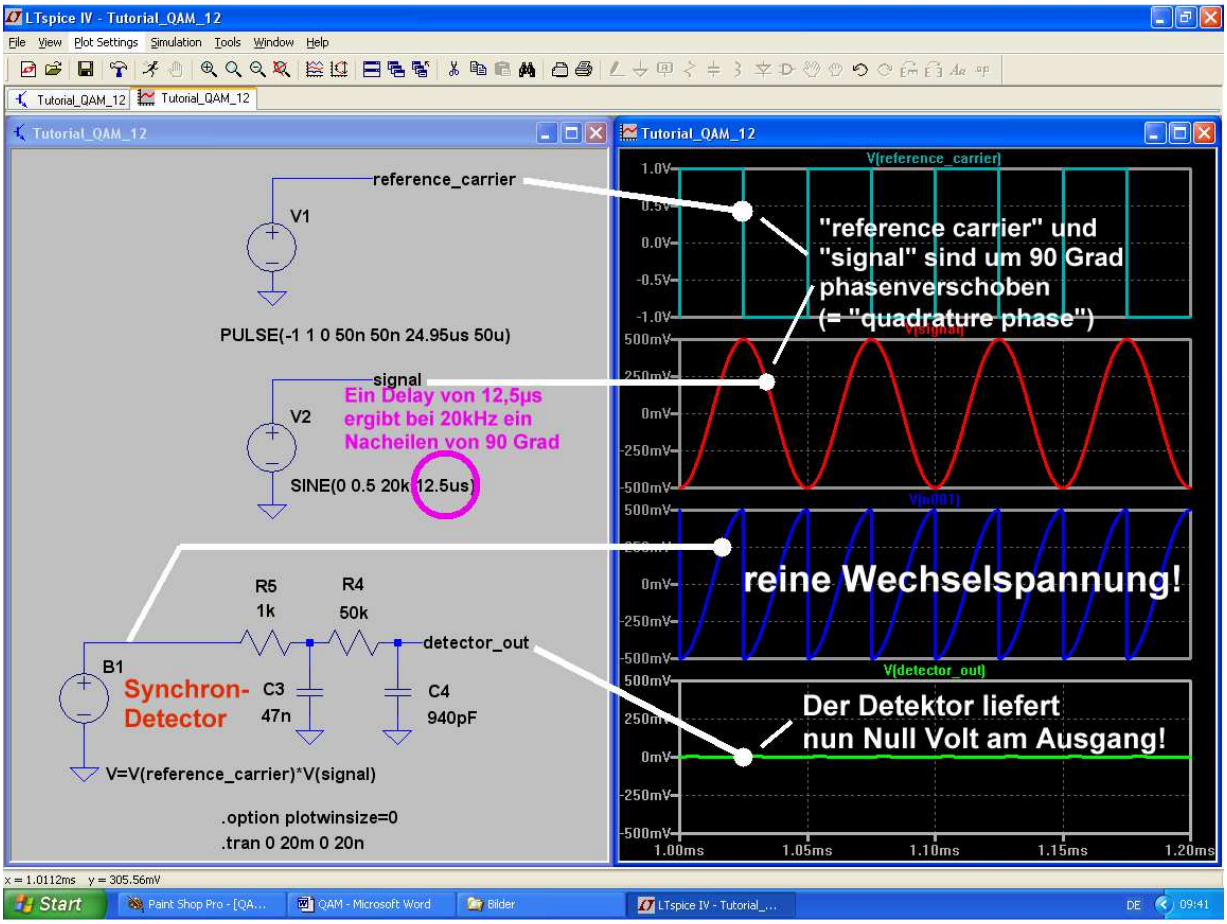
Sehen wir uns an, was bei Gleichphasigkeit beider Signale passiert:



Nach der Multiplikation der beiden Signale sieht man, dass sich die Schaltung wie ein Brückengleichrichter verhält. Die negativen Halbwellen des QAM-Signals werden durch das zum gleichen Zeitpunkt ebenfalls negative Carrier-Rechtecksignal umgepolt (= drittes Diagramm von oben) und bei einem QAM-Spitzenwert von 500mV erhält man dann am Ausgang des Detektors nach der Filterung den berühmten „Arithmetischen Mittelwert“. Er beträgt laut Lehrbuch

$$U_{DC} = \frac{U_{peak} \bullet 2}{\pi} = \frac{500mV \bullet 2}{\pi} = 318mV$$

Und nun die Lösung für diejenigen QAM-Signalanteile, die um ***90 Grad gegenüber der verwendeten Trägerkomponente phasenverschoben*** sind:



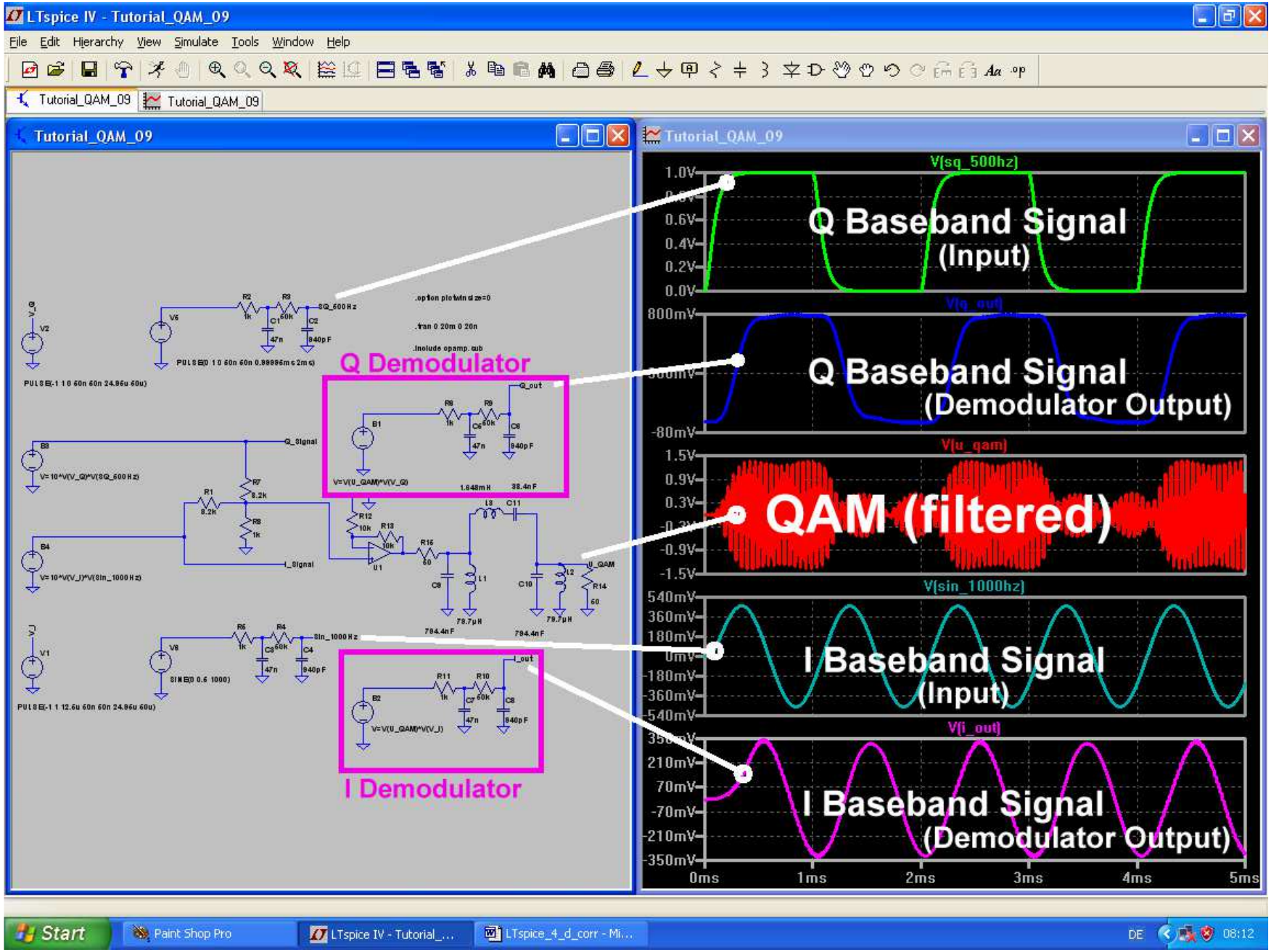
Da wird die Spannung am Detektorausgang zu Null, denn nach der Multiplikation beobachten wir eine reine Wechselspannung mit gleich großen positiven und negativen Flächen ohne DC-Anteil (wieder: drittes Diagramm von oben).

24.4.4.2. Die Simulation der vollständigen QAM-Demodulatorschaltung

Der Kern sind natürlich jetzt **zwei „bv“ – Quellen**, die als **Analog-Multiplizierer** dienen.
Wie besprochen, werden jeder Schaltung das empfangene QAM-Signal sowie die entsprechende Trägerkomponente mit der passenden Phasenlage zugeführt. Durch diese Multiplikationen erreicht man, dass bei beiden Achsen nur die jeweils mit der entsprechenden Achsenrichtung „korrelierenden“(= in der Phasenlage übereinstimmenden) Signalanteile herausgefischt werden.

Die erforderlichen Formel-Eingaben bei den bv-Quellen lauten deshalb	
für den „Q“ – Pfad:	**V = V(V_QAM) * V(V_Q)**
und für den “I” – Pfad:	**V = V(V_QAM) * V(V_I)**

Dann wird sich an jedem Multiplizierer-Ausgang wieder das zugehörige Basisband-Signal einstellen!
Zur Beseitigung der Trägerfrequenzreste (mit doppelter Trägerfrequenz) folgt an jedem Demodulator-Ausgang die schon bekannte Tiefpassschaltung. So kommt man zum Bild auf der nächsten Seite.


LTspice IV - Tutorial_QAM_09
File Edit Hierarchy View Simulate Tools Window Help
Tutorial_QAM_09
Q Demodulator
I Demodulator
Q Baseband Signal (Input)
Q Baseband Signal (Demodulator Output)
QAM (filtered)
I Baseband Signal (Input)
I Baseband Signal (Demodulator Output)
V(sq_500hz)
V(q_out)
V(u_qam)
V(sin_1000hz)
V(i_out)
0ms 1ms 2ms 3ms 4ms 5ms
Start
Paint Shop Pro
LTspice IV - Tutorial_...
LTspice_4_d_corr - Mi...
DE
08:12

Wie man sieht, sind die beiden übertragenen Informationen nach der Demodulation identisch mit den beiden zugeführten Basisband-Signalen. Lediglich die abgerundeten Ecken und die müden Flanken weisen beim Rechtecksignal darauf hin, dass seinen Oberwellen gründlich die Haare geschnitten wurden. Und ein leichtes Nacheilen der beiden Ausgangssignale gegenüber den ausgestrahlten Basisbändern demonstriert die zeitlichen Verzögerungen durch die begrenzten Bandbreiten der Filterschaltungen in jedem Kanal. Aber genau das ist die Sache mit der berühmten „**Gruppenlaufzeit“ (= group delay).**