

SPICE-Schaltungs-Simulation mit LTspice IV



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

Tutorial-Version 2.2

Copyright by

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

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

20. April 2012

Inhaltsverzeichnis	Seite
1. Einführung	5
2. Installation	6
3. Übungen am mitgelieferten Demo-Beispiel „Astable Multivibrator“	6
3.1. Aufruf von Schaltplan und Simulation	6
3.2. Darstellung der Signalverläufe	8
3.3. Löschen von Signalverläufen im Ergebnis-Bildschirm	10
3.4. Andere Farbe für eine Ergebniskurve	10
3.5. Änderung der Simulationszeit	11
3.6. Änderung des dargestellten Spannungs- oder Strombereichs	13
3.7. Cursor-Einsatz	15
3.7.1. Verwendung eines Cursors	15
3.7.1. Verwendung eines zweiten Cursors	15
3.8. Differenzmessungen	16
3.9. Strom-Messungen	17
3.10. Änderung von Bauteilwerten	18
4. RC-Tiefpass als erstes eigenes Projekt	19
4.1. Zeichnen des Stromlaufplans mit dem Editor	19
4.2. Zuweisung neuer Bauteilwerte	20
4.3. Untersuchung von einmaligen Vorgängen	21
4.3.1. Die Sprungantwort	21
4.3.2. Ein- und Ausschaltvorgang	23
4.4. Periodische Signale am Eingang	24
4.4.1. Sinussignal mit $f = 1591 \text{ Hz}$	24
4.4.2. Rechtecksignal mit $f = 1691 \text{ Hz}$	25
4.4.3. Dreiecksignal mit $f = 1691 \text{ Hz}$	26
4.5. AC-Sweep zur Ermittlung des Frequenzganges	27
5. FFT (= Fast Fourier Transformation)	29
6. Zweites Projekt: Gleichrichtung	32
6.1. Einpuls-Gleichrichter ohne Trafo	32
6.2. Eine wichtige Sache: ein Transformator für die Gleichrichterschaltungen	33
6.2.1. Die bequemste Lösung: simpler, idealer Trafo	33
6.2.2. Etwas für die Wirklichkeit: Erstellung eines realistischen SPICE-Modells für einen Transformator mit zwei Wicklungen	33
6.2.3. Erzeugung eines passenden Symbols für diesen Transformator	35
6.3. Einpuls-Gleichrichter mit realistischem Trafo	36
6.4. Verwendung der Diode 1N4007 in der Gleichrichterschaltung	38
6.5. Zweipuls-Gleichrichter mit realistischem Trafo	40
7. Drittes Projekt: Drehstrom	42
7.1. Programmierung eines Drehstromsystems	42
7.2. Drehstrom-Lichtmaschine im Auto	43
8. Viertes Projekt: Darstellung von Bauteil-Kennlinien	45
8.1. Ohm'scher Widerstand	45
8.2. Diode	46
8.3. NPN-Transistor	47
8.4. N-Kanal-Sperrschiicht-FET	49

9. Fünftes Projekt: Schaltungen mit Transistoren	50
9.1. Einstufiger Verstärker	50
9.1.1. Ansteuerung mit einem Sinus-Signal	50
9.1.2. Simulation des Frequenzganges („AC-Sweep“)	52
9.2. Zweistufiger gegengekoppelter Breitbandverstärker	53
9.2.1. Pflichtenheft	53
9.2.2. Simulations-Schaltung und Simulations-Vorgaben	54
9.2.3. Simulation in der Time Domain (= im Zeitbereich)	54
9.2.4. DC-Bias (= Gleichstrom-Analyse)	55
9.2.5. AC-Sweep (= Frequenzgang von 1 Hz bis 200 MHz)	57
9.3. Der Parameter-Sweep	58
10. Sechstes Projekt: OPV-Schaltungen	60
10.1. Einstieg: Umkehrender Verstärker	60
10.1.1. Verwendung des einfachsten OPV-Modells „opamp“	60
10.1.2. Einsatz des Universalen OPVs und die Verwendung von Labels	61
10.1.3. Verwendung eines OPVs vom Typ „TL072“ und Erstellung eines passenden Spice-Symbols	63
10.2. Nicht umkehrender Verstärker mit TL072	68
10.3. Analog-Addierer mit TL072	69
10.4. Probleme beim Betrieb mit nur einer Betriebsspannung	73
11. Siebtes Projekt: DC-DC-Konverter	71
11.1. Bereitstellung des Power-MOSFETs „IRFZ44N“	71
11.2. Der Step-Up-Konverter (= Aufwärtswandler)	73
11.3. Der Flyback-Konverter (= Sperrwandler)	75
11.4. Der Step-Down-Konverter (= Abwärtswandler)	77
12. Achte Projekt: Phasenanschnitt-Steuerung mit Thyristor	79
12.1. Das eingesetzte Thyristor-Modell	79
12.2. Schalten von Ohm'schen Lasten	80
12.3. Schalten von induktiven Lasten	81
12.4. Zündung des Thyristors über einen Gate-Transformator	81
13. Neuntes Projekt: Echos auf Leitungen	83
13.1. Leitungen -- nur zwei Drähte?	83
13.2. Echos auf Leitungen	85
13.3. Simulation des vorigen Rechenbeispiels mit LTSpice	87
13.4. Leerlauf oder Kurzschluss als Last am Kabelende	90
13.5. Verwendung von Kabel mit Verlusten (Beispiel: RG58 / 50Ω)	92
13.5.1. Wie simuliere ich RG58-Kabel?	92
13.5.2. Simulation der Kabeldämpfung bei 100MHz	93
13.5.3. Speisung der RG58-Leitung mit einer Pulsspannung	96
13.5.4. Ein Kurzschluss am Ende der RG58-Leitung	97
14. Zehntes Projekt: S-Parameter	98
14.1. Jetzt nochmals Echos, aber mit System	98
14.2. Praxisbeispiel: 110MHz – Tschebyschef – Tiefpassfilter (LPF)	101
15. Elfte Projekt: Double Balanced Mixer (= Ringmodulator)	105
15.1. Etwas Grundlagen	105
15.2. Standardschaltung des Ringmodulators	106
15.3. Ein Problem: die erforderlichen Übertrager	107
15.4. Simulation des DBM-Verhaltens mit idealen Übertragern	108

16. Zwölftes Projekt: Digitale Schaltungssimulation	110
16.1. Was man vorher wissen sollte	110
16.2. Einfacher Anfang: die Umkehrstufe (= NOT oder Inverter)	111
16.3. Der AND-Baustein	112
16.4. Das D-Flipflop	113
16.5. Dreistufiger Frequenzteiler mit D-Flipflops	114
17. Dreizehntes Projekt: Rausch-Simulation	115
17.1. Etwas Grundlagen	115
17.1.1. „Rauschen“ -- woher kommt das?	115
17.1.2. Weitere Rauschquellen	117
17.1.3. Rauschtemperatur und Noise Figure eines Twoports	118
17.2. Simulation der Spektralen Rauschleistungs-Dichte	118
17.3. Simulation der Noise Figure in dB	121
18. Untersuchung eines Sinus-Oszillators	125
19. Signalformen und Oberwellen	130
19.1. Grundlagen	130
19.2. Simulation des Spektrums eines einmaligen Pulses	133
19.3. Simulation des Frequenzspektrums bei einem periodischen Pulssignal	135
19.4. Der ideale Sinus	136
19.5. Der einseitig begrenzte Sinus	137
19.6. Der symmetrisch begrenzte Sinus	138
20. Die Sache mit der Impulsantwort	139
20.1. Erstes Beispiel: Dirac-Impuls speist einen 160Hz-RC-Tiefpass	144
20.2. Zweites Beispiel: Dirac-Untersuchung des 110MHz-Tiefpasses aus Kapitel 14.2	144
20.2.1. Bestimmung von S21 (= Forward Transmission)	144
20.2.2. Bestimmung von S11 (= Input Reflection) oder S22 (= Output Reflection)	149
21. Modulation	151
21.1. Prinzip der Amplitudenmodulation (AM)	151
21.2. Amplitude Shift Keying (ASK)	153
21.3. Die Frequenzmodulation (FM)	154
21.3.1. Erzeugung eines FM-Signals mit der Spannungsquelle „voltage“	154
21.3.2. FSK = Frequency Shift Keying	156
21.4. Quadratur-Amplitudenmodulation (QAM)	158
21.4.1. Grundlegende Informationen	158
21.4.2. Erzeugung eines QAM-Signals	159
21.4.3. Filterung des QAM-Signals zur Anpassung an die Übertragungs-Kanalbandbreite	166
21.4.4. Demodulation des QAM-Signals	170
21.4.4.1. Prinzip der Synchron-Demodulation	171
21.4.4.2. Die Simulation der vollständigen QAM-Demodulatorschaltung	172

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 aber gleich wissen: der Industriestandard für Schaltungssimulationsprogramme lautet **SPICE**.

Es 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 tummeln sich 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 selbstverständlich komplett simuliert wird. Diese Programme kosten aber kräftig Geld (...der größte und nobelste Lieferant ist hier ORCAD....) und deshalb gibt es auch einfachere Sachen.

Man findet aber auch sehr schöne „pädagogische“ Programme, die während der Ausbildung oder des Studiums eingesetzt werden können und sollen (Stichwort: „Workbench“) und mit tollen Details aufwarten (z.B. völlig realistische Darstellung und Bedienung der Messgeräte, wie Generatoren oder Oszilloskope oder Spectrum-Analyzer usw.). Allen diesen Programmen gemeinsam ist aber ein Nachteil: es gibt zwar kostenlose Demo- und Studentenversionen, aber irgendwo steckt da immer ein Pferdefuß: entweder ist die Zahl der maximal verwendbaren Bauteile begrenzt oder die Schaltplangröße und Platinengröße wird fast bis auf Briefmarkenformat beschnitten oder der maximal belegbare Arbeitsspeicher darf einen Maximalwert nicht überschreiten oder ganz moderne Bauteile mit ihren sehr aufwendigen und umfangreichen Modellen werden gleich mal wegen ihrer Größe abgelehnt oder das Übernehmen eines neuen Bauteils in die Bibliothek gleicht vom Aufwand her einem Hausbau oder....oder....oder....

Deshalb ist es sehr erfreulich, dass man bei genauer Suche (...und nach entsprechenden Tipps von Gleichgesinnten!) 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 Einsteigerhit ist hierbei „**5spice**“. Es ist wegen seines absichtlichen Minimalaufbaus mit Abstand am leichtesten erlernbar, aber sobald man mehr Komfort und weitere zusätzliche Funktionen sucht, stößt man auf

LTspice IV (vorher: LT SwitcherCAD)

von Linear Technologies. Diese Software will nur Schaltungen simulieren und weist keinerlei Einschränkungen auf, da sie als freundliches Angebot der Firma an alle Entwicklungingenieure gedacht ist (...natürlich zuerst zur Simulation von Schaltreglern und Schaltnetzteilen unter Verwendung von Halbleitern dieser Firma, aber das ist doch absolut in Ordnung....). Sie wird in der Zwischenzeit auch für sehr viele andere Zwecke eingesetzt und hat sich einen großen Freundeskreis erworben.

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 heraußhat, läuft es prima. Allerdings wie in einer guten, langjährigen Ehe: Viele Eigenschaften liebt man, aber ebenso muss man vieles in Liebe ertragen...

Zum Schluss noch eine ganz wichtige Vorbemerkung:

Diese unterschiedlichen Eingabemöglichkeiten bedeuten,

dass viele Funktionen des Programms (Beispiel: „Simulierte jetzt die Schaltung“) auch auf völlig verschiedene Weise zugänglich sind oder gestartet werden können!

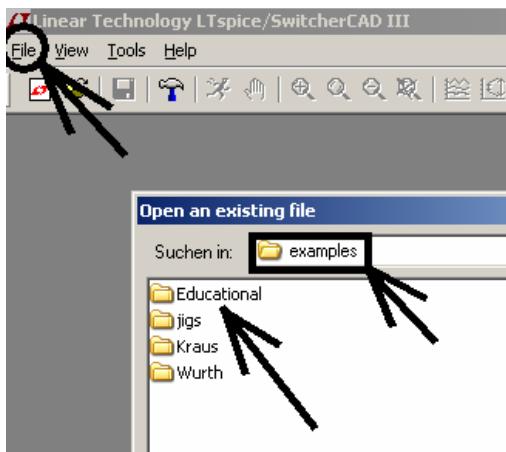
Im Tutorial werden jedoch NICHT ALLE dieser unterschiedlichen Methoden einzeln besprochen. Hier muss der Anwender selbst prüfen, wozu etwas da ist bzw. welche Methode er in Zukunft verwenden will.

2. Installation auf dem PC

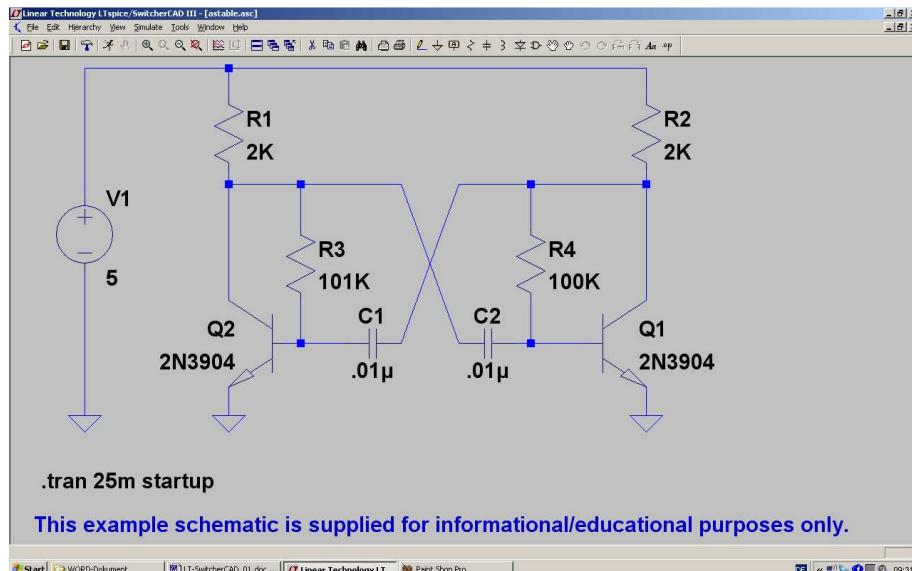
Nichts leichter als das: nach dem Download aus dem Internet (www.linear.com) klickt man die „*.exe-Datei“ an und wird in gewohnter Weise durch die Windows-Installation geführt. Der zugehörige Icon wird hierbei auf dem Windows-Startbildschirm automatisch angelegt. Fertig!

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

3.1. Aufruf von Schaltplan und Simulation



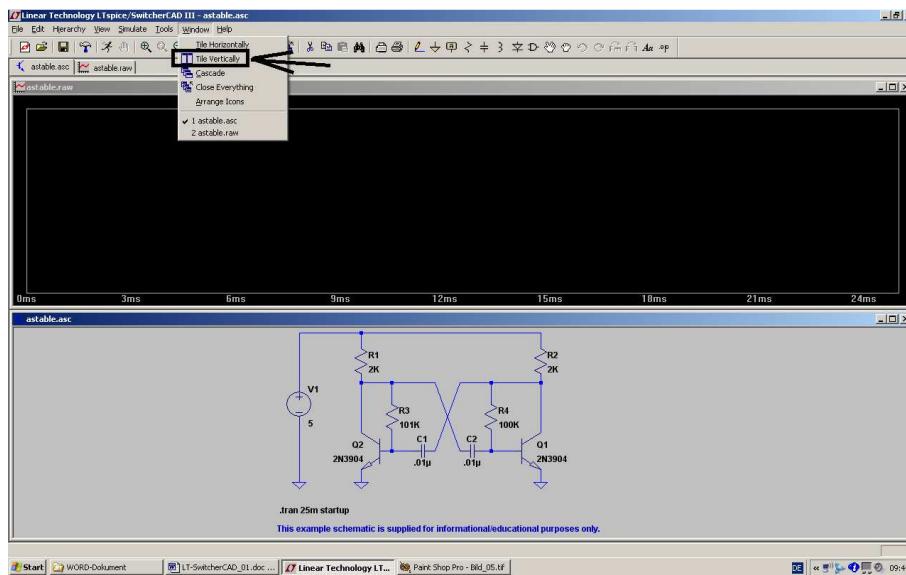
Nach dem Programmstart klicken wir im linken oberen Eck erst auf „File“ und dann „Open“. Im Verzeichnis unseres Programms gibt es dann mehrere Ordner. Wenn wir darin „Examples“ und anschließend „Educational“ öffnen, können wir jetzt einen **astabilen Multivibrator** auswählen (= „astable.asc“).



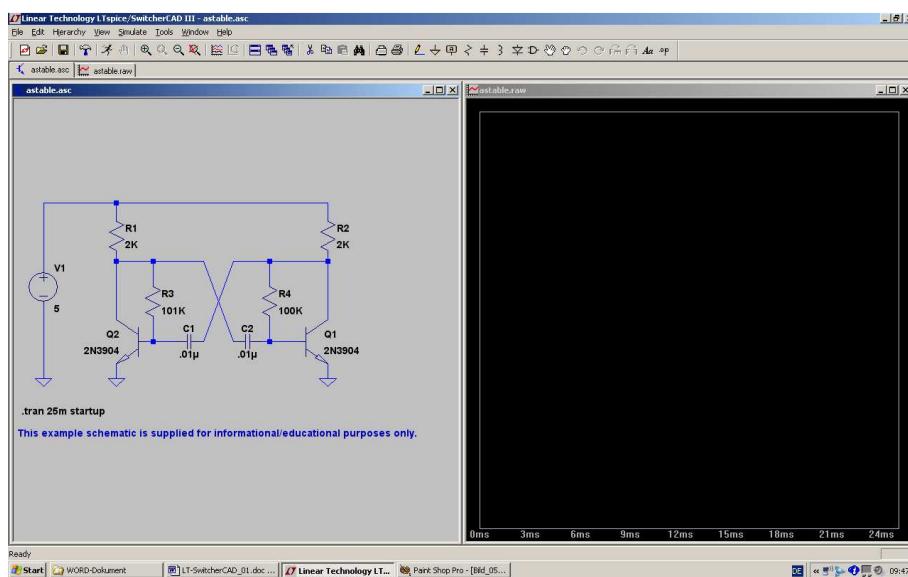
Haben wir alles richtig gemacht, dann empfängt uns der nebenstehende Bildschirm mit der fertigen Schaltung.



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 Ergebnisdarstellung gedacht, aber noch leer.
Dass die Schaltung zusätzlich in der unteren Bildhälfte präsentiert wird, hat seinen Grund...

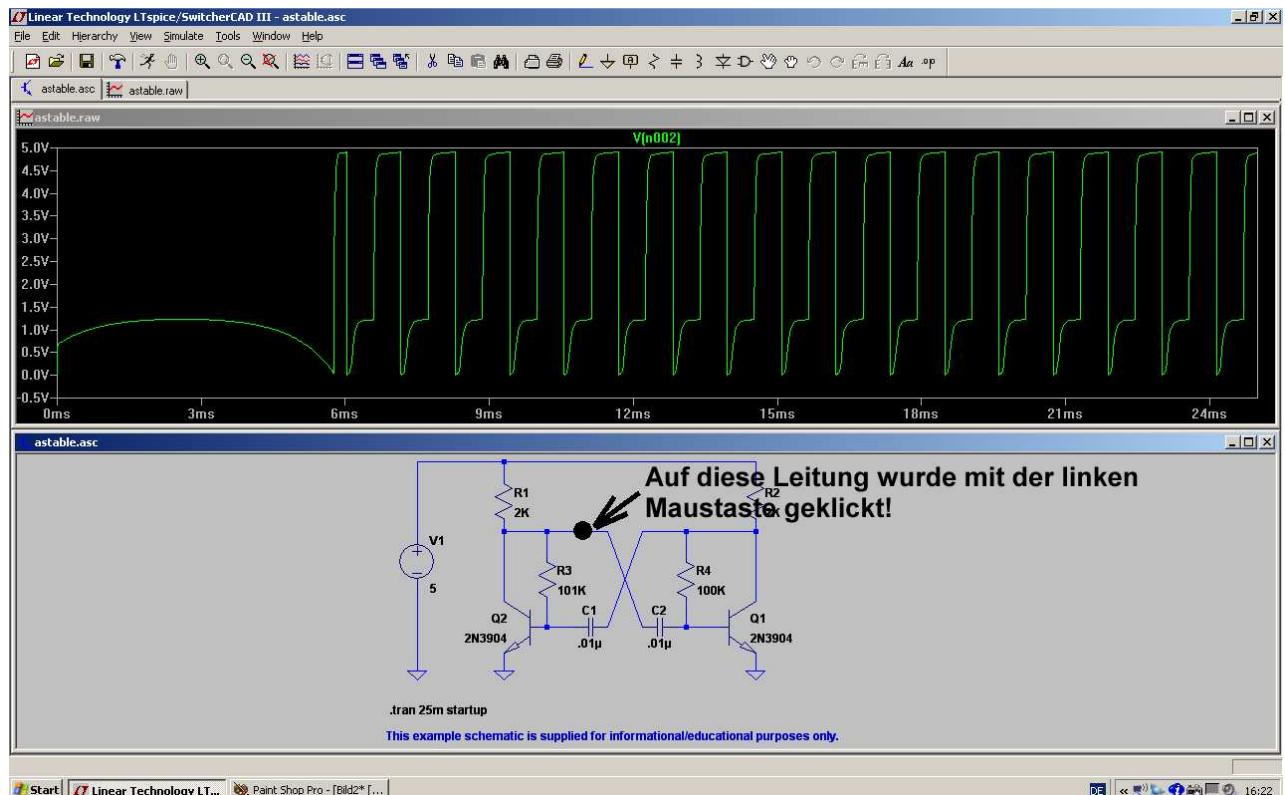


Übrigens:
Wem diese Darstellung nicht behagt, weil er lieber links die Schaltung und rechts die Signalverläufe sehen möchte, der öffne das Menü „Window“ und rufe „Tile vertically“ auf. Dann sieht das Alles so aus:

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

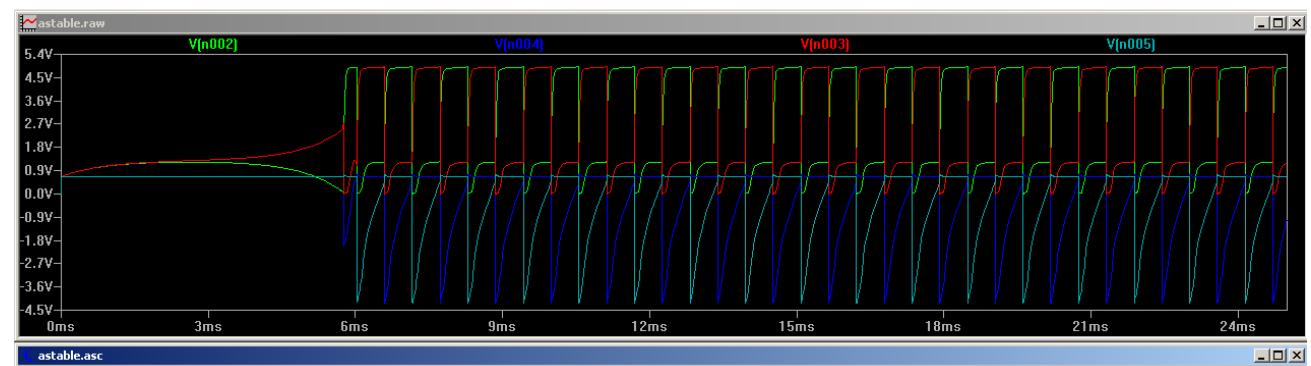
3.2. Darstellung der Signalverläufe

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 in der Schaltung gemessen, dann sieht das Ergebnisfenster so aus:



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

- 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**“.

```

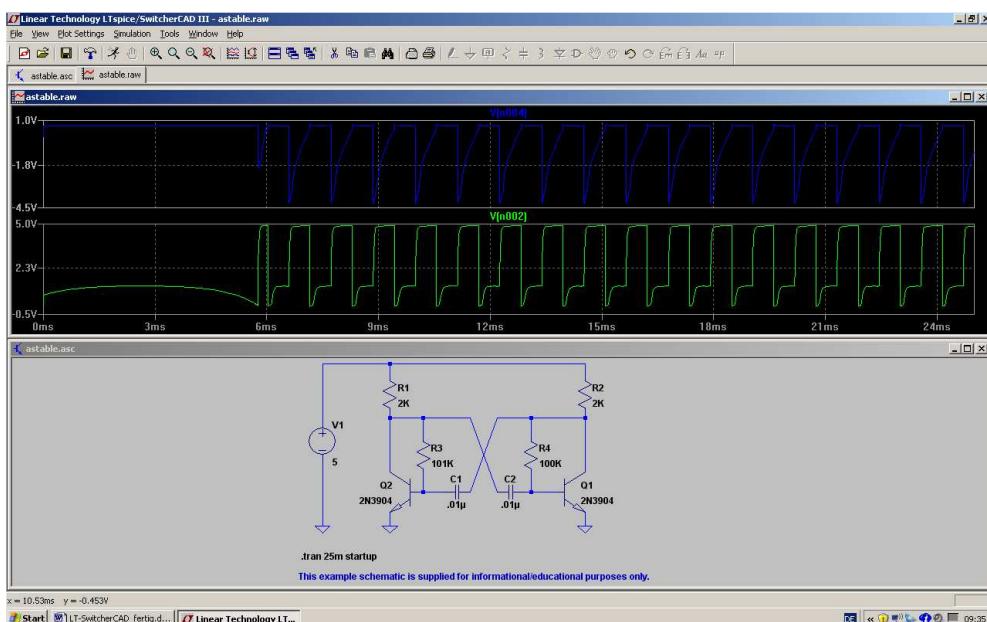
SPICE Netlist: C:\Programme\LTC\SwCADIII\examples\Educational\astable.net
* C:\Programme\LTC\SwCADIII\examples\Educational\astable.asc
R1 N001 N002 2K
R2 N001 N003 2K
R3 N002 N004 101K
R4 N003 N005 100K
C1 N003 N004 .01p
C2 N005 N002 .01p
V1 N001 0 5
Q1 N003 N005 0 0 2N3904
Q2 N002 N004 0 0 2N3904
.model 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 only
.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 benutzen**.

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!

3.3. 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 Fenster 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 löschen 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.

3.4. Andere Farbe für eine Ergebniskurve

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



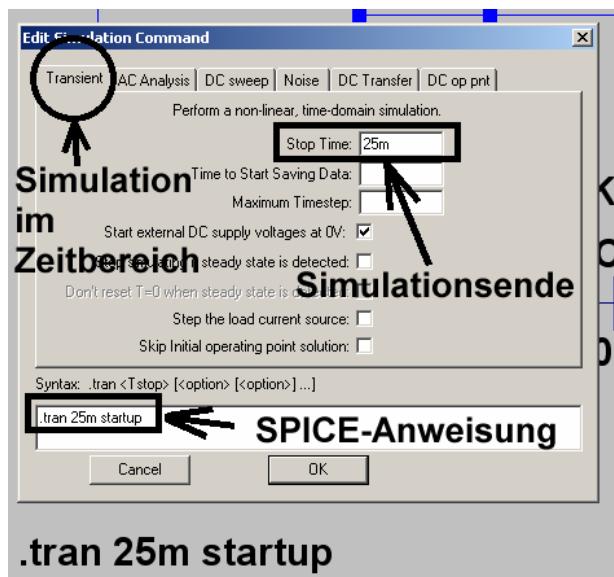
3.5. Änderung der Simulationszeit



In dieser Anweisungszeile versteckt sich der SPICE-Simulationsauftrag! Er sagt:

Simuliere die Verläufe aller Signale in der Zeit von Null bis 25 Millisekunden.

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

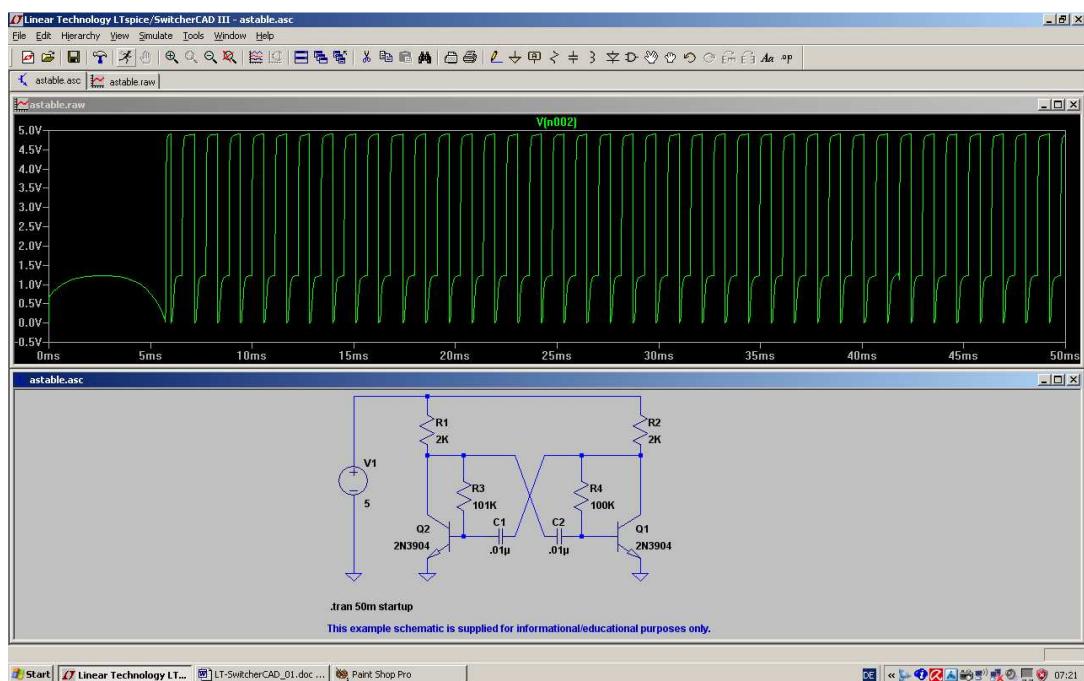


Stellt man nun den Mauszeiger mitten in den Text dieser Zeile und drückt **die rechte Maustaste**, dann erscheint dieses zugehörige Menü.

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 diesem Programm 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:

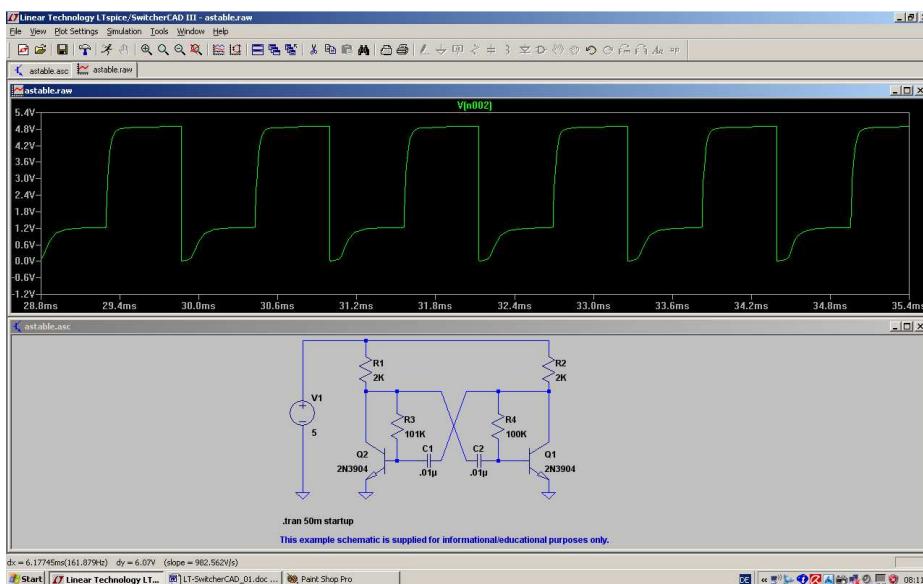
Aufgabe:

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

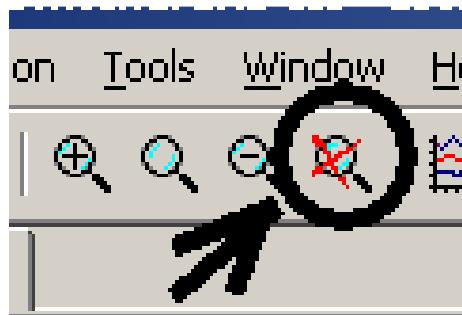
Lösung:



- 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“

3.6. Änderung des dargestellten Spannungs- oder Strombereichs

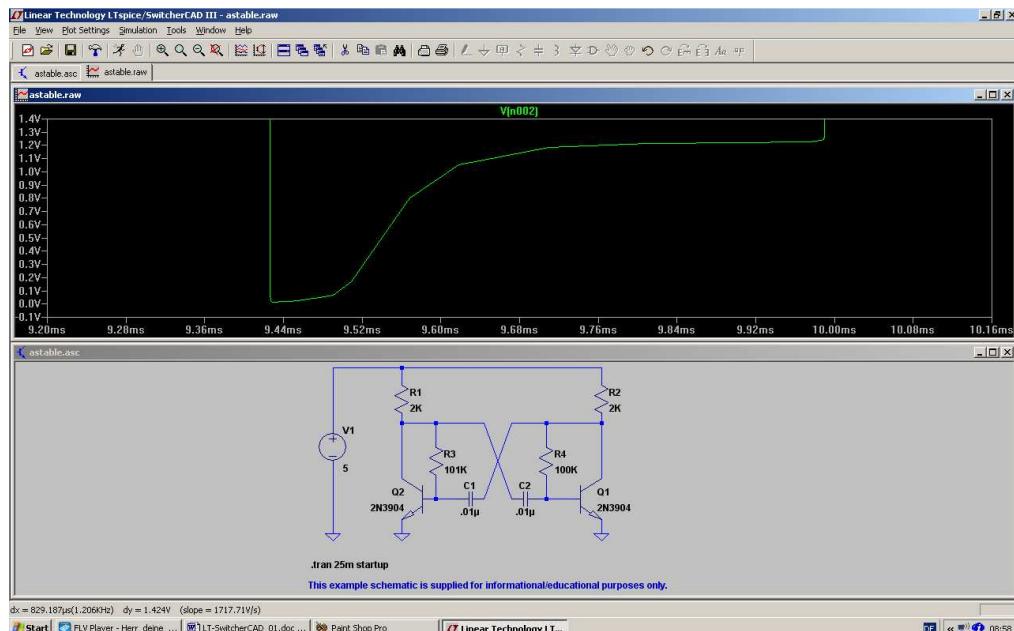
Oft interessieren bei einer Schaltungsentwicklung nur bestimmte Details bei den Signalverläufen und man möchte deshalb den angezeigten Wertebereich eingrenzen. Da gibt es zwei Möglichkeiten:

- Man arbeitet mit der eben besprochenen **Zoom-Funktion** oder
- Man schaltet die „automatische Bereichswahl“ (= Autoranging) aus und gibt den angezeigten Wertebereich selbst vor.

Zu a):

Aufgabe: Bei der Kollektorspannung des linken Transistors unseres astabilen Multivibrators soll die kurze negative Spitze der Spannung genauer untersucht werden. Zoomen Sie diesen Bereich heraus.

Lösung:



Zu b):

Aufgabe: Sorgen Sie dafür, dass bei der senkrechten Diagrammachse nur der Bereich von 0....+1,5V dargestellt wird.



1. Schritt:

Man klickt mit der **RECHTEN Maustaste** auf die Werteskala der senkrechten Achse. Dadurch öffnet sich das zugehörige Menü.

Hinweis:

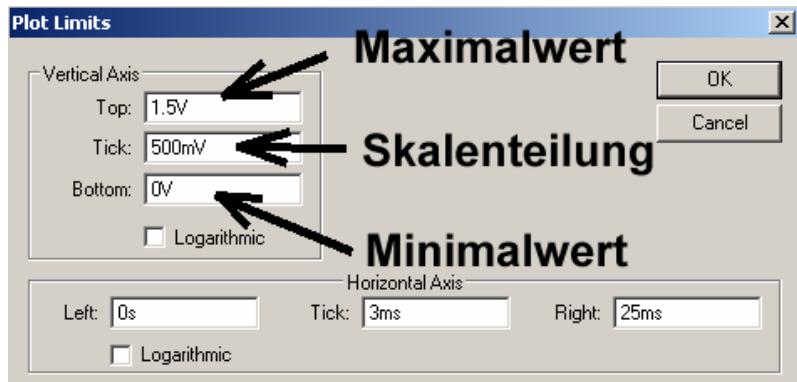
wer einfach nur die Maus auf die linke oder untere Werteskala rollt, bekommt nun **ein Lineal** als Cursor angeboten. Mit einem linken Klick öffnet sich dann

ein **KLEINES** Menü

für Änderungen an der Achsendarstellung.

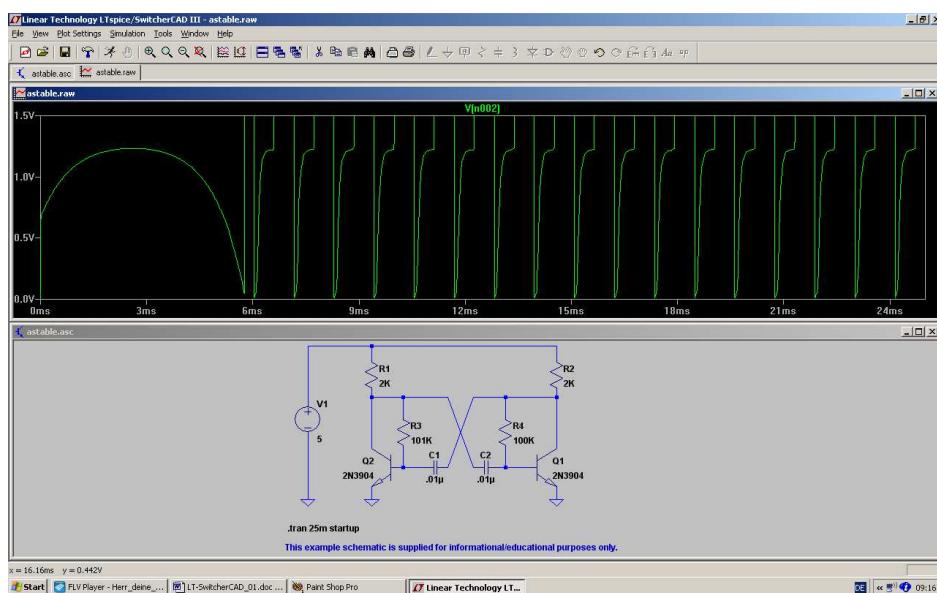
Bitte mal ansehen, aber wir wollen uns doch gleich mit der „Großen Lösung“ beschäftigen und die geht so:

Im auftauchenden Menü (Siehe nebenstehendes Bild) wird „**Autoranging**“ aus- und „**Manual Limits**“ eingeschaltet.



2. Schritt:
Darin werden nun die neuen Grenzen für die senkrechte Achse eingetragen (... „Tick“ steht hierbei für „Skalenteilung“).

Hier können wir auch die Einteilung für die waagrechte Achse ändern und einen anderen Zeitabschnitt einstellen!

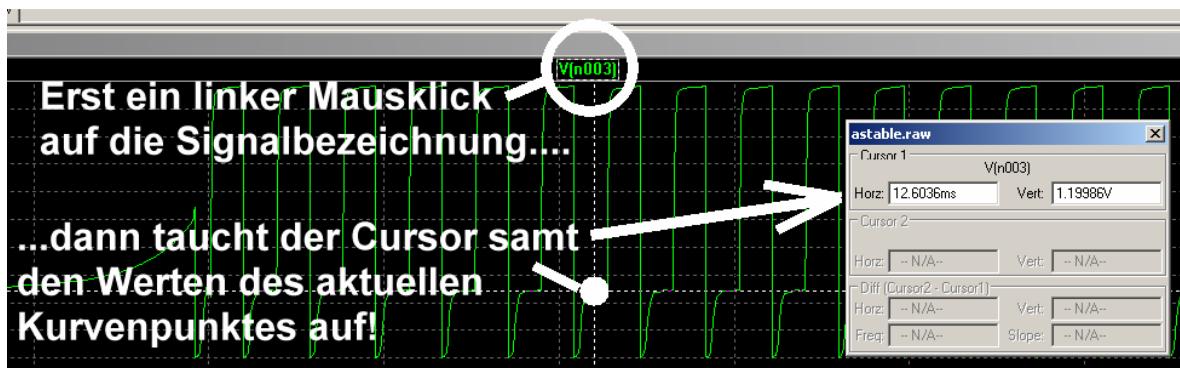


Damit sind wir am Ziel und können wieder zum Fenster mit der Schaltung zurückkehren, um Schaltungsänderungen vorzunehmen.

3.7. Cursor-Einsatz

3.7.1. Verwendung eines Cursors

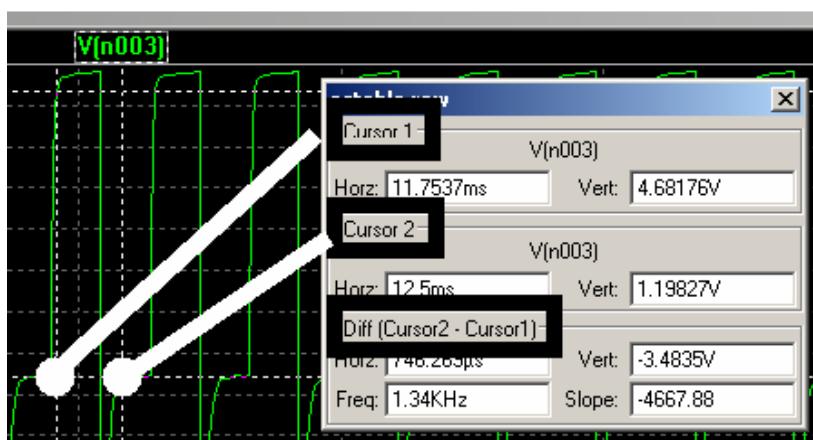
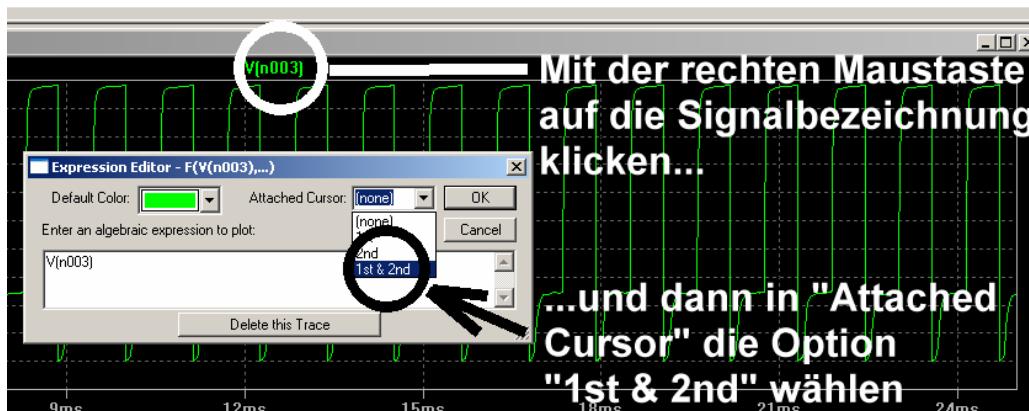
Oft möchte man eine Ergebniskurve „abfahren“, um an bestimmten Zeitpunkten den dort gültigen Spannungs- oder Stromwert zu ermitteln. Das geht so:



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.

3.7.2. Verwendung eines zweiten Cursors

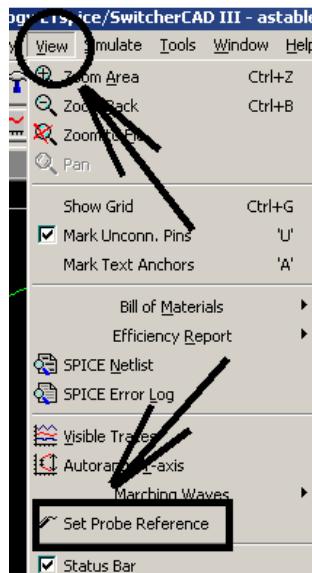
Dazu klickt man mit der **RECHTEN Maustaste** auf die Signalbezeichnung und geht im auftauchenden Menü „Attached Cursor“



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

3.8. 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?

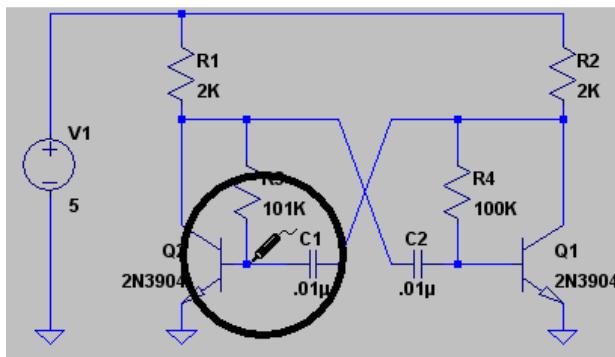


Nun, beim LT-SwitcherCAD geht das etwas anders: man muss zuerst bewusst einen neuen Referenzpunkt in die Schaltung setzen und dann am anderen Bauteilanschluss klicken!

Das geht so:

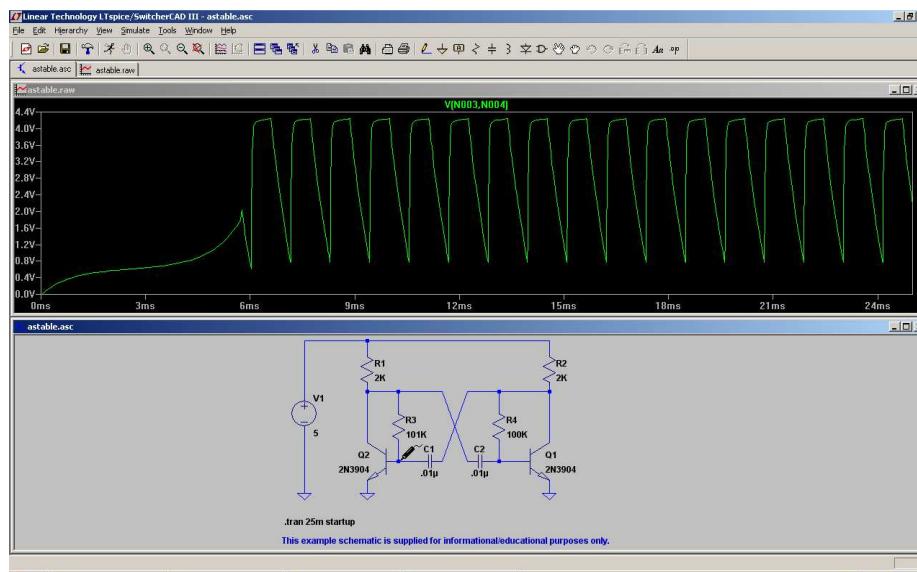
Man öffnet das Menü „View“ und findet darin die Funktion „**Set Probe Reference**“, die angeklickt wird.

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.

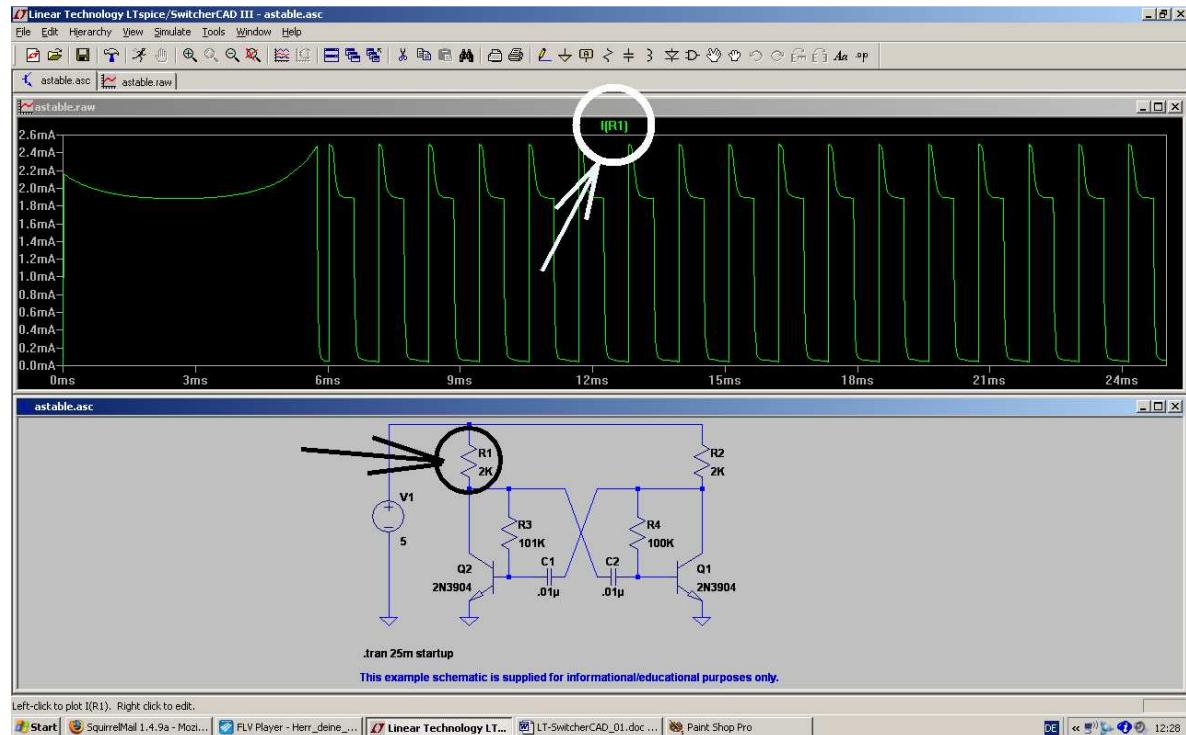
Übrigens:
Den Reference-Point bringt man durch einen Druck auf die Escape-Taste zum Verschwinden, wenn man wieder „normal“ (= gegen Masse) messen möchte.

3.9. 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.

Beispiel:

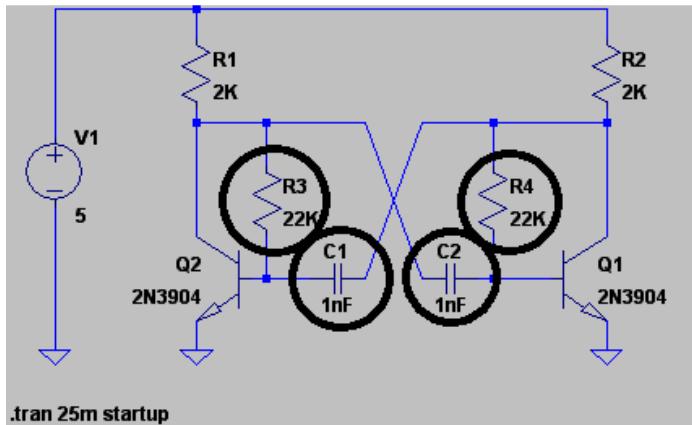
Stromverlauf im Widerstand R1 (= Kollektorwiderstand des linken Transistors):



3.10. Änderung von Bauteilwerten

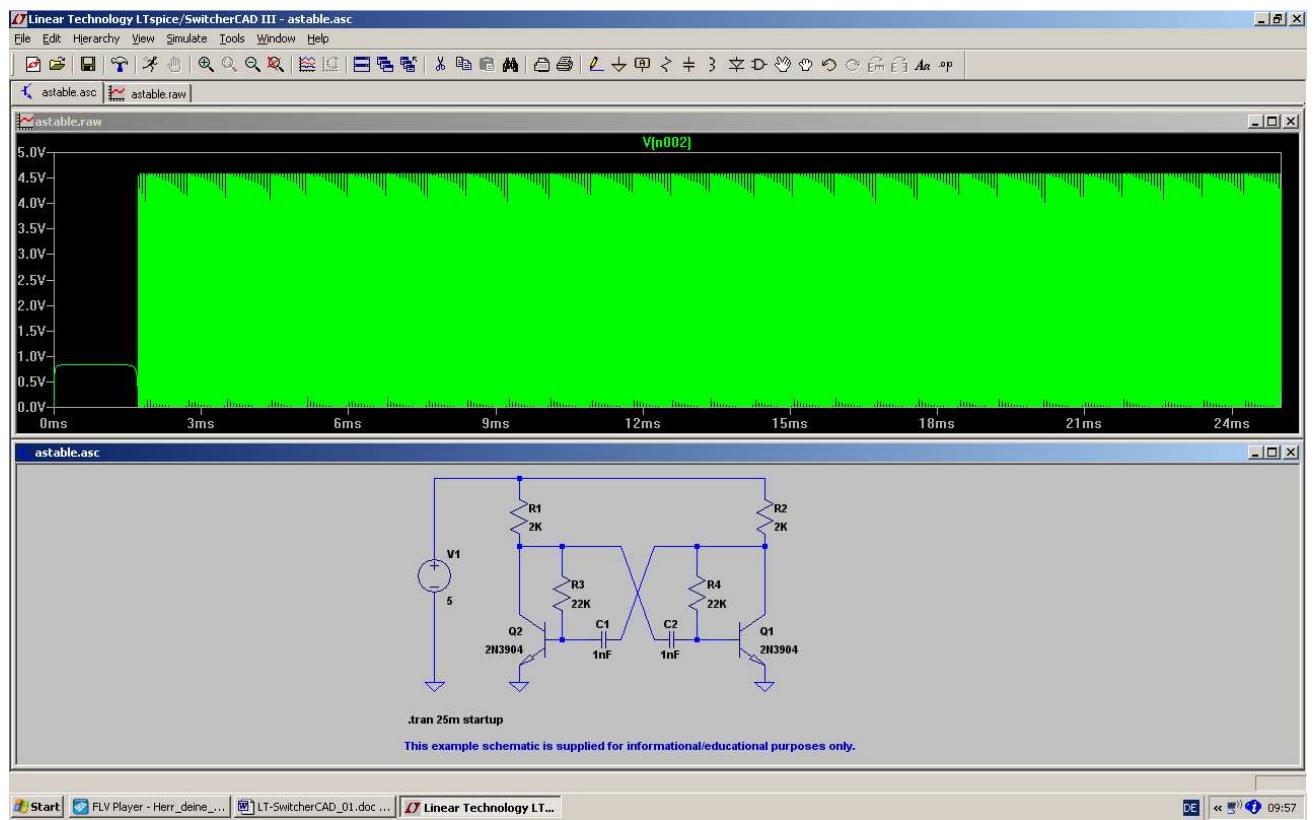
Nun wollen wir die Simulation mit anderen Bauteilwerten wiederholen und ändern beide Kondensatoren auf je 1000pF sowie die beiden Basiswiderstände auf je $22k\Omega$ ab.

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:

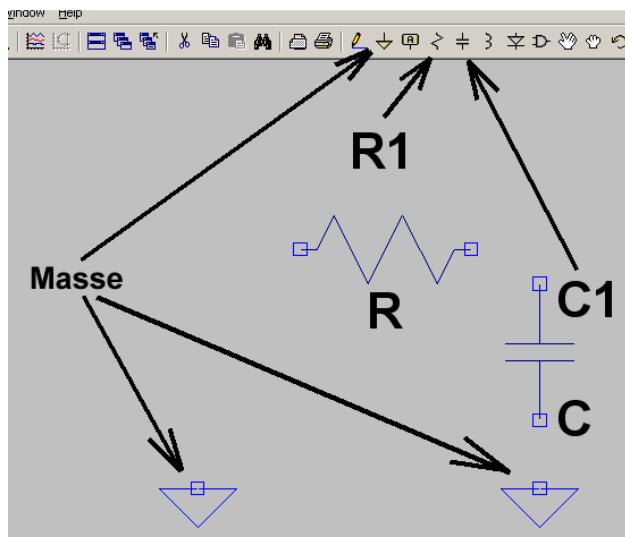


Da muss man nun natürlich mit der Zoom-Funktion drangehen oder gleich die Simulationszeit ändern, wenn man Details sehen oder die neue Frequenz messen möchte.

4. RC-Tiefpass als erstes eigenes Projekt

4.1. 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>

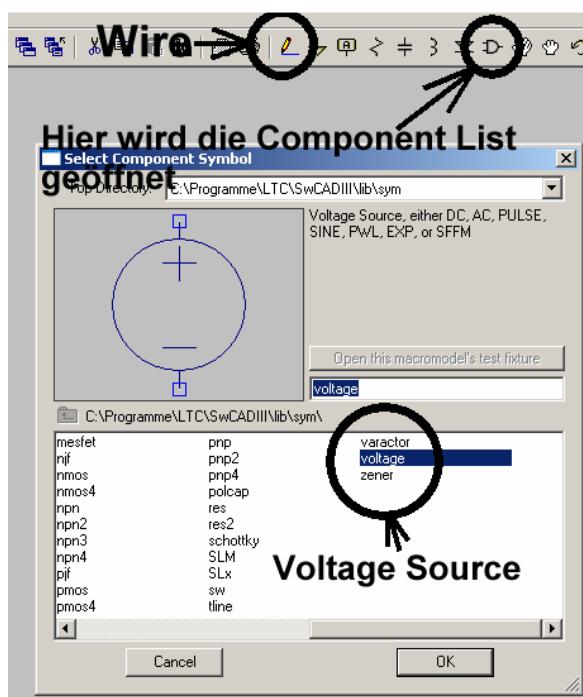
um 90 Grad und setzen es in der Schaltung ab. Anschließend klicken wir mit der **rechten Maustaste** kurz auf den Bildschirm (...oder drücken „**ESCAPE**“...), denn erst dadurch verschwindet das Widerstandssymbol am Cursor wieder.

Jetzt wiederholen wir die Prozedur mit dem **Kondensator** sowie **zwei Massezeichen** und haben dann den nebenstehenden Stand:

Dazu zwei Tipps:

- 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.
- Soll ein Bauteil dagegen auf dem Bildschirm **verschoben** 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.

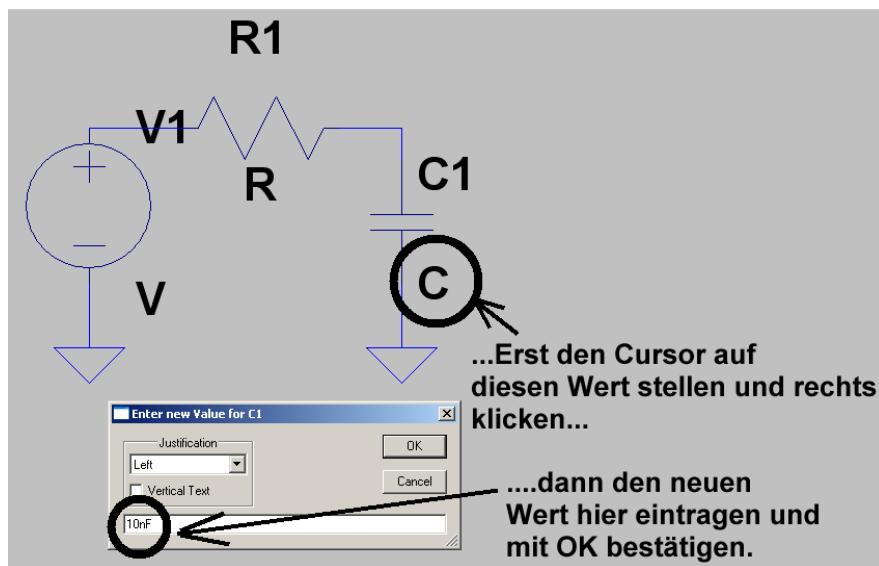
In der auftauchenden Component-List (...darin ist alles aufgeführt, womit man arbeiten kann...) gehen wir zu „**voltage**“ und bestätigen mit OK. Dann hängt die Quelle am Cursor und kann in die Schaltung eingesetzt werden.

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 eine Verbindungsdräht 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.

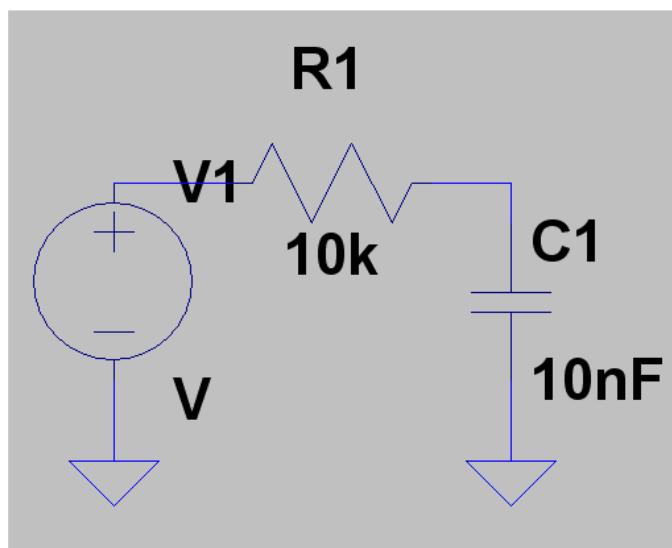
4.2. Zuweisung neuer Bauteilwerte



Die Zuweisung oder Änderung eines Bauteilwertes 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

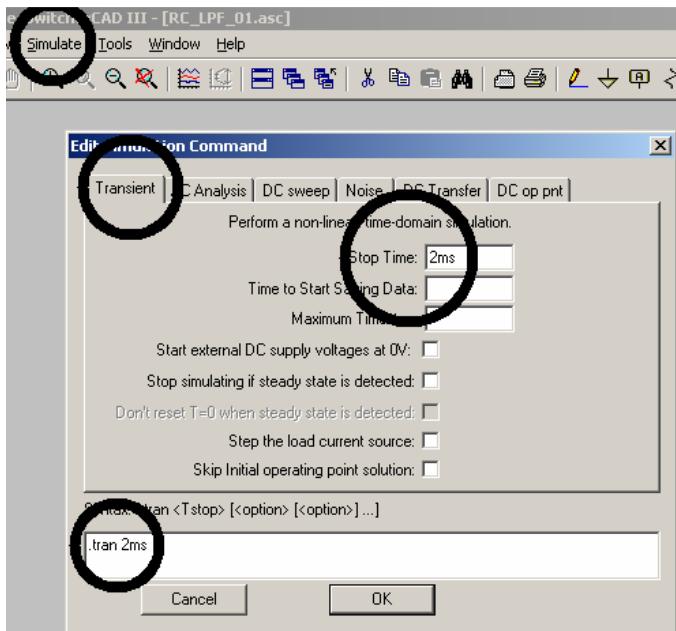
(= time domain oder frequency domain)

wir simulieren wollen.

4.3. Untersuchung von einmaligen Vorgängen

4.3.1. 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.

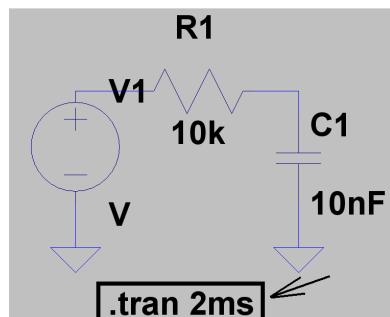


Hinter dem **Simulate**-Menü versteckt sich die Zeile „**Edit Simulation Command**“.

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

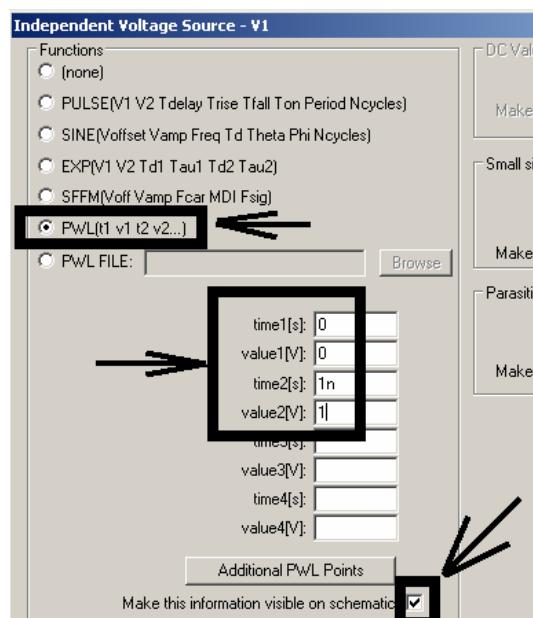
.tran 2ms

übernommen wurde. Bitte schließlich mit OK bestätigen.



Dieser Eintrag „**tran 2ms**“ hängt am Cursor und muss sich nun im Schaltplan absetzen lassen..

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 das „ADVANCED“ – Menü.



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

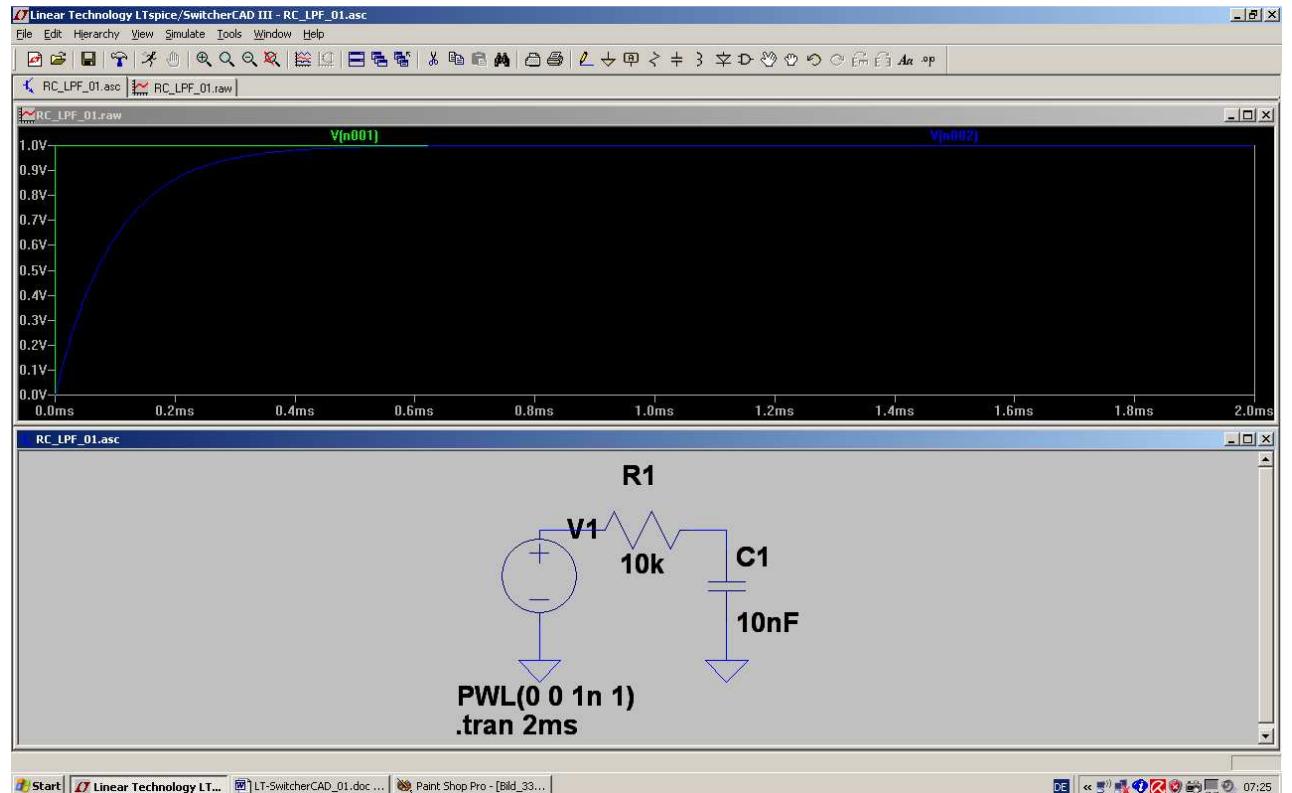
PWL = piecewise linear

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).

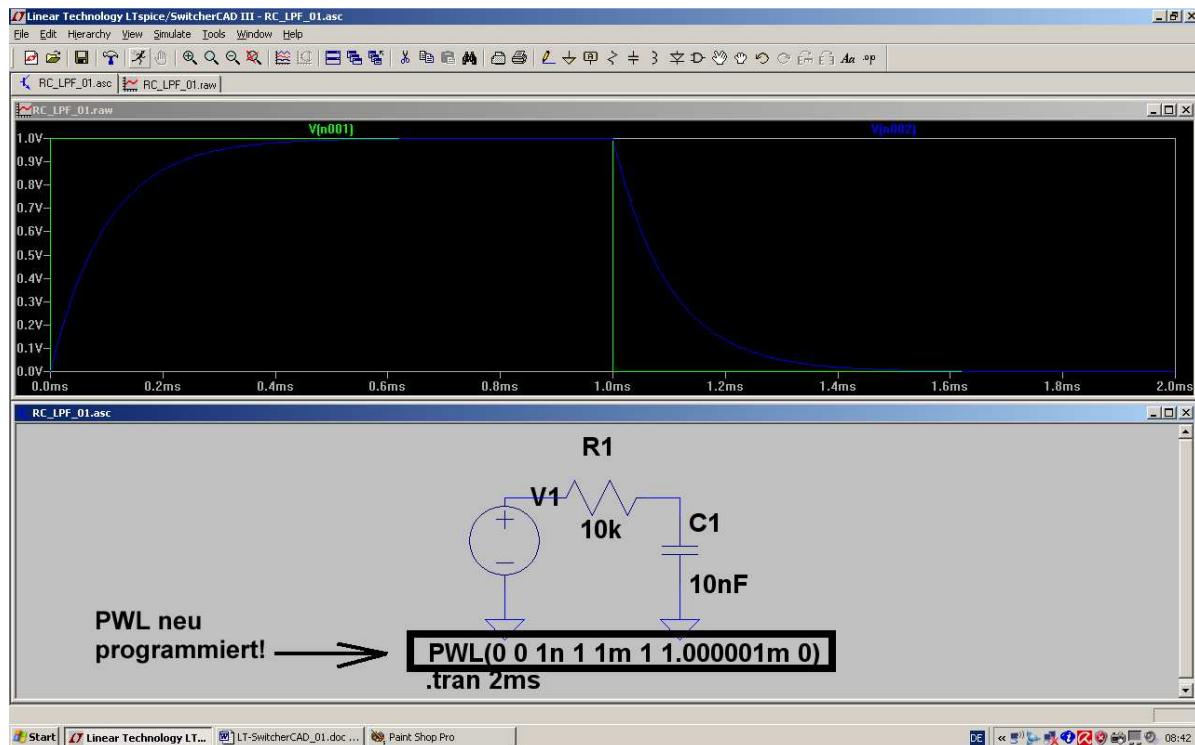
Nun kommt der Druck auf die Run-Taste mit den laufenden Männchen, das Anklicken von Eingangs- und Ausgangspunkt in der Schaltung und das Bewundern des Ergebnisses:



4.3.2. Ein- und Ausschaltvorgang

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

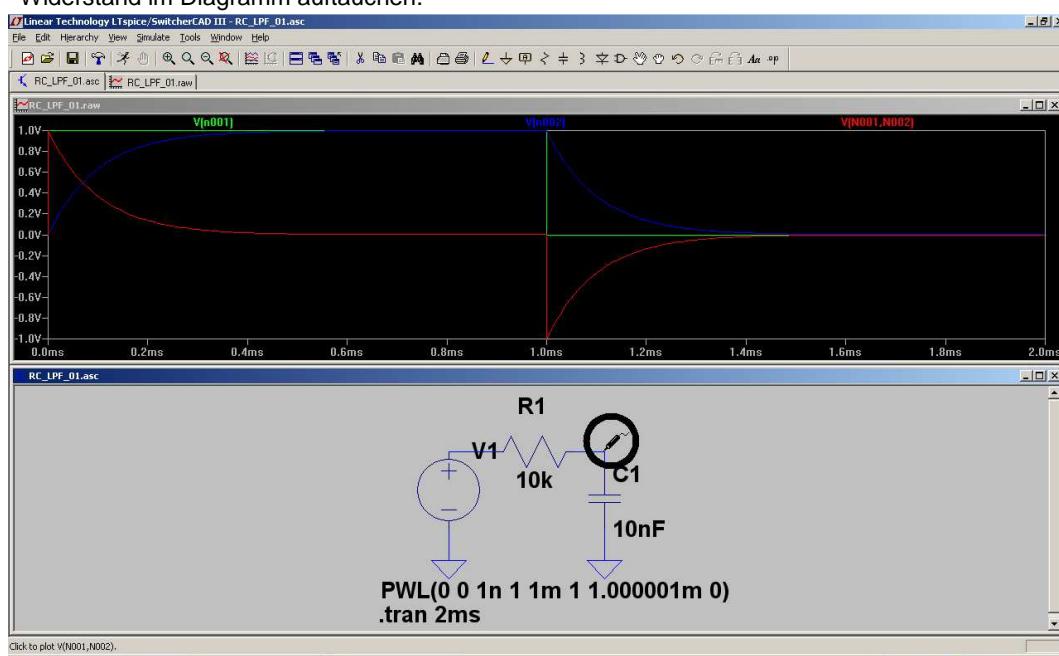
0	0	(Null Volt beim Startpunkt)
1n	1	(1 Volt nach einer Nanosekunde)
1m	1	(immer noch 1 Volt nach einer Millisekunde)
1.000001m	0	(eine Nanosekunde später wieder auf Null Volt zurück)



Zusatzaufgabe:

Stellen Sie zusätzlich im Diagramm die Spannung am Widerstand dar.

Lösung: wir setzen einen „Reference Point“ am Ausgang (= Verbindung zwischen Widerstand und Kondensator). Tiptt man dann an die Verbindung von Spannungsquelle und Widerstand, so sollte die gesuchte Spannung am Widerstand im Diagramm auftauchen.



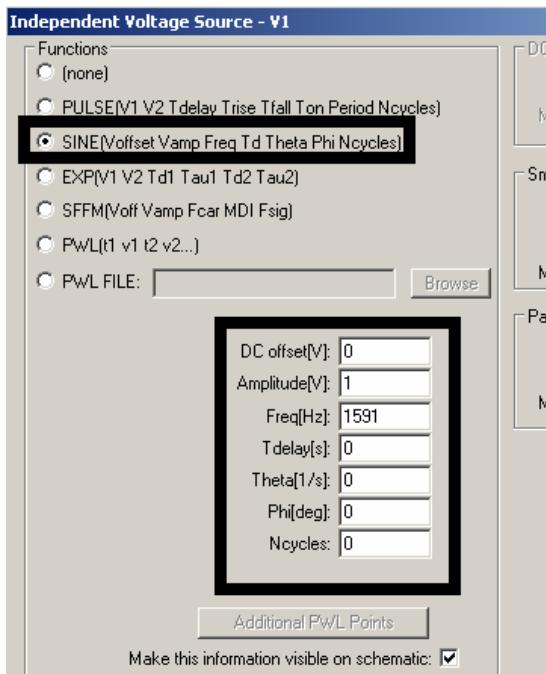
4.4. Periodische Signale am Eingang

4.4.1. 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:

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

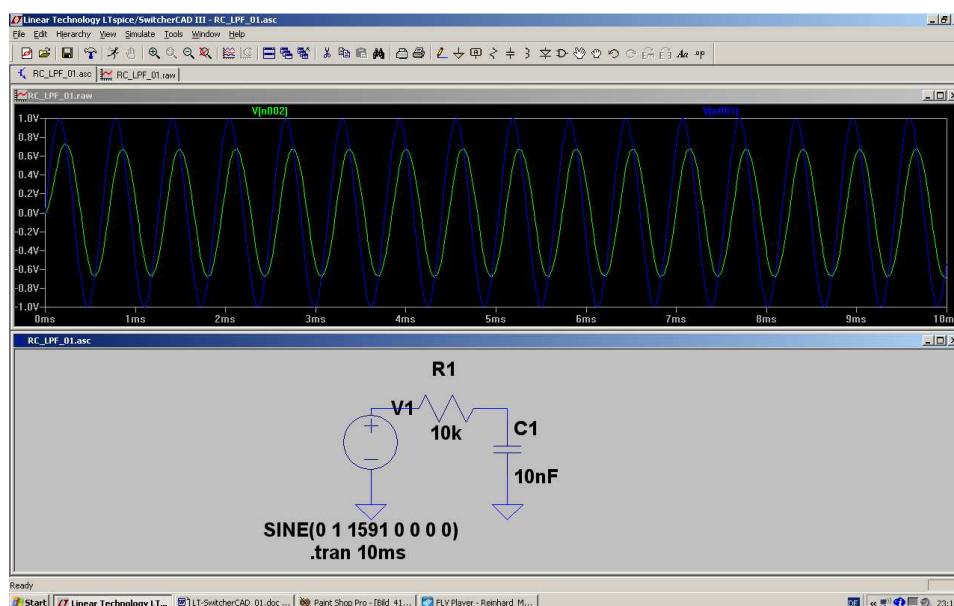


Dazu muss wieder das Symbol der Spannungsquelle rechts angeklickt und auf „Sinusbetrieb“ umgestellt werden.

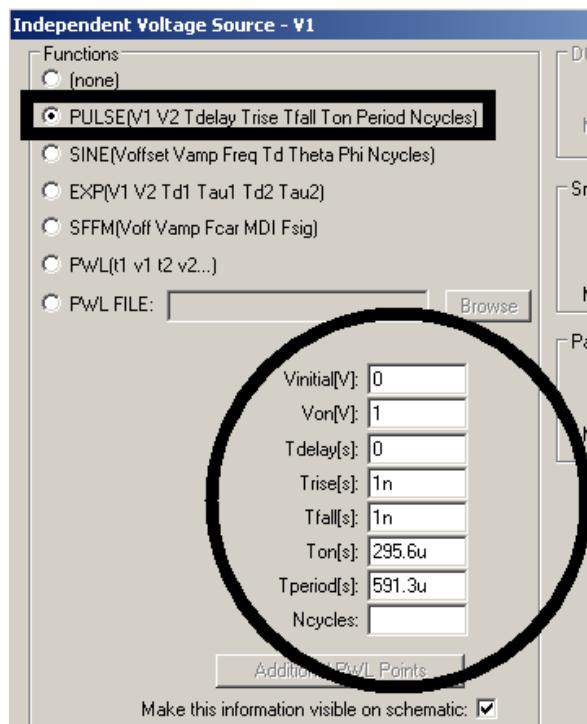
Außerdem sind folgende Einträge nötig:

- DC Offset** = Null
- Amplitude** = 1 Volt (= Spitzenwert)
- Frequenz** = 1691 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 Ausgangsspannung.



4.4.2. Rechtecksignal mit $f = 1691$ 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)

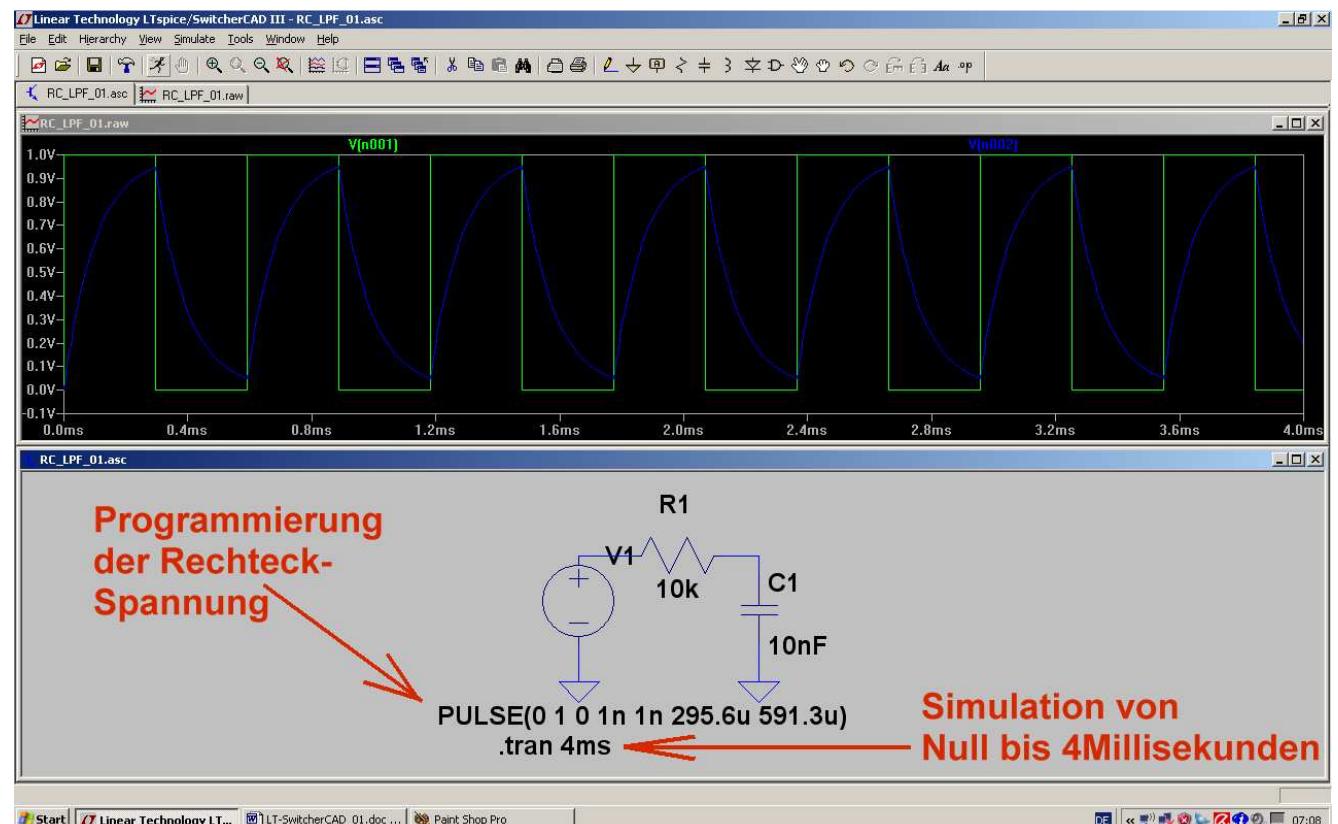
Tfall = 1 Nanosekunde (= Abfallzeit)

Ton = 295,6 Mikosekunden (= Pulsdauer)

Tperiod = 591,3 (= Periodendauer)

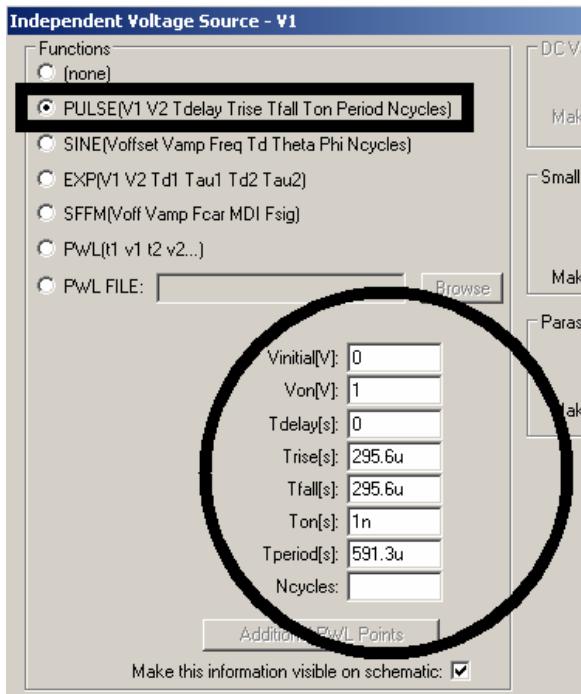
Ncycles bleibt ohne Eintrag (= Zahl der Zyklen, wenn ein Burstsignal erzeugt werden soll)

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



4.4.3. Dreiecksignal mit $f = 1691$ 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 = 295,6 Mikrosekunden (= Anstiegszeit)

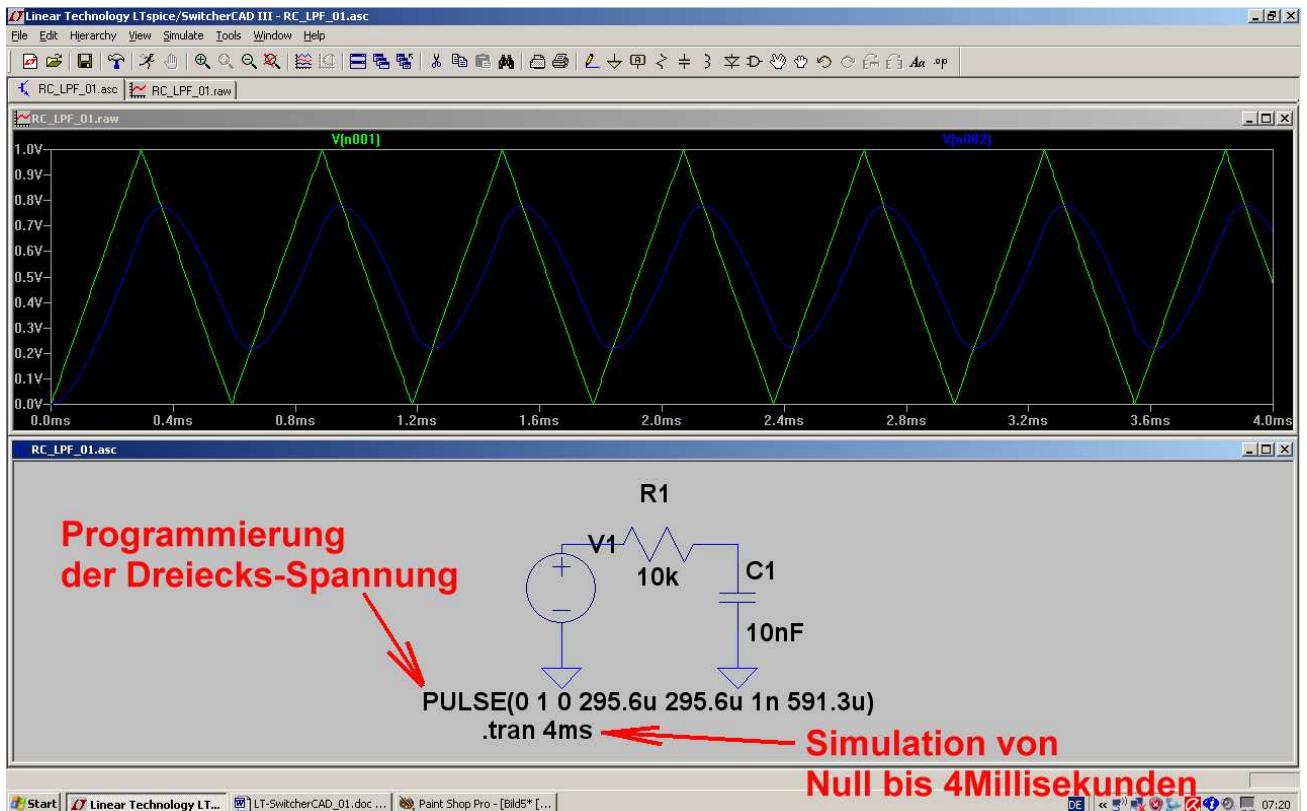
Tfall = 295,6 Mikrosekunden (= Abfallzeit)

Ton = 1 Nanosekunde (= Pulsdauer)

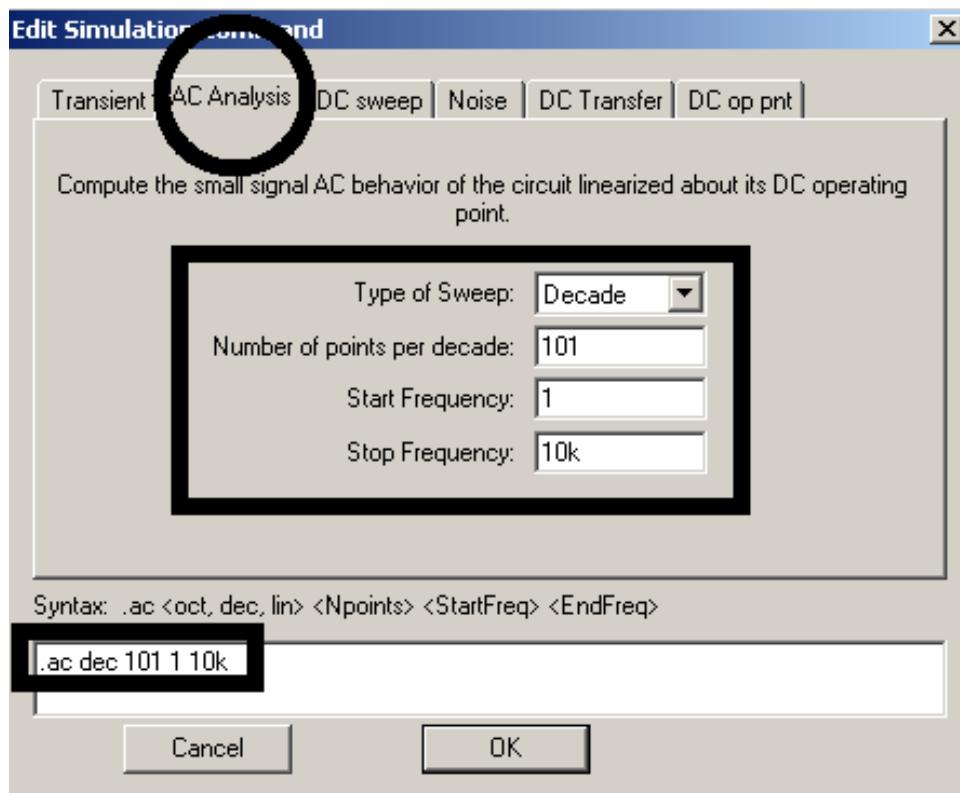
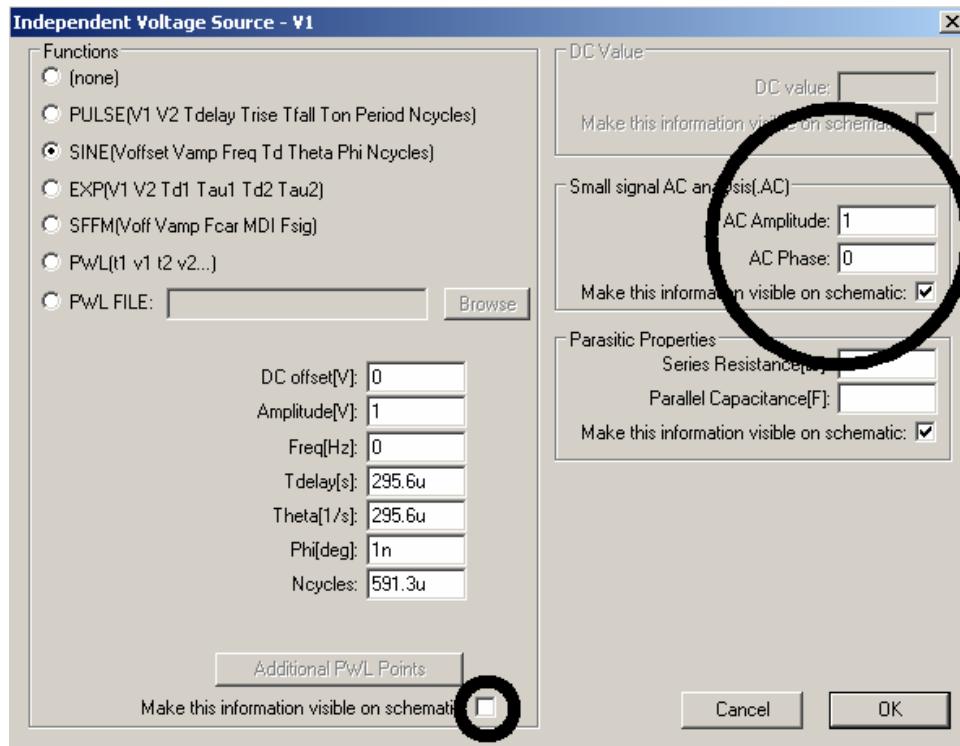
Tperiod = 591,3 (= Periodendauer)

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:



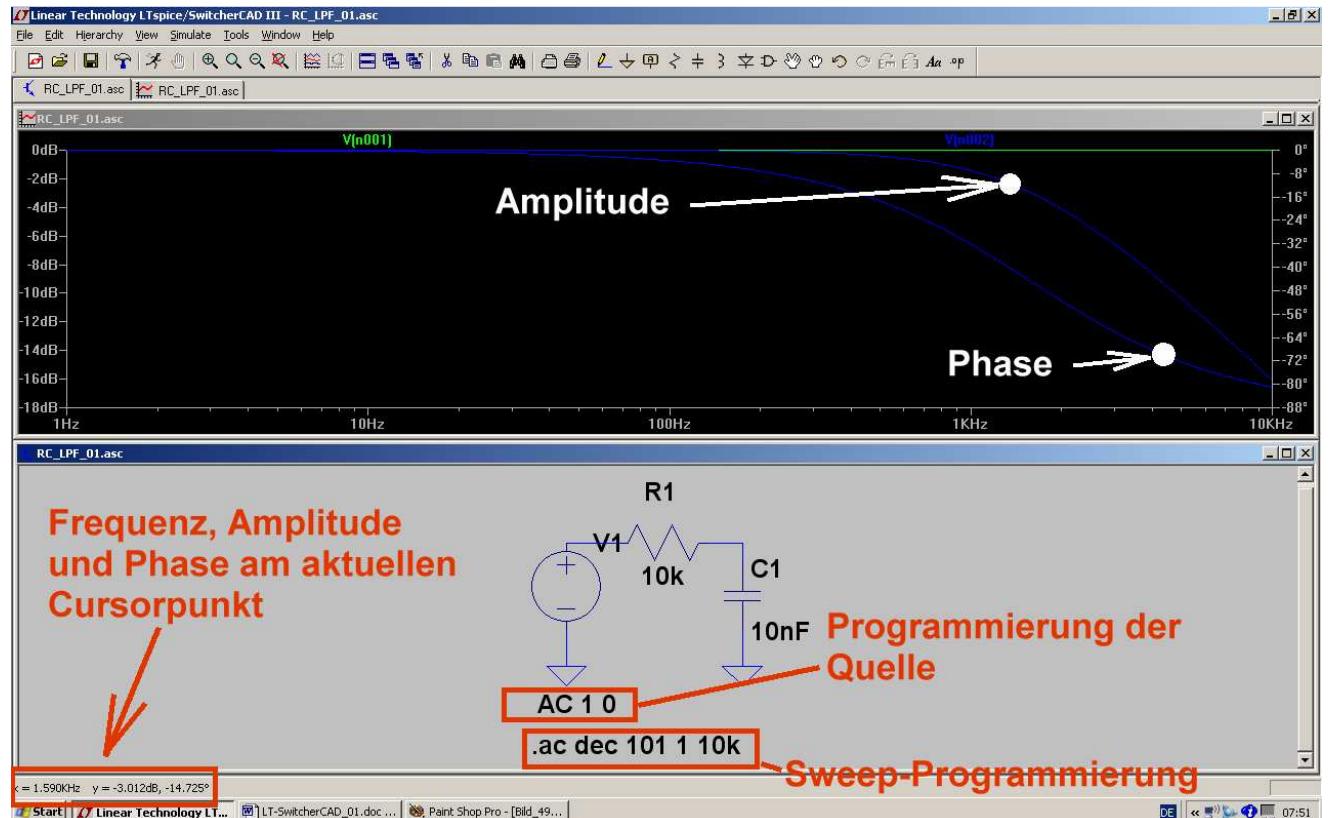
4.5. AC-Sweep zur Ermittlung des Frequenzganges



Bitte die korrekte Übernahme in die Kommandozeile prüfen!

So einen Bildschirm müssen wir dann nach der Simulation vor uns haben.

Bitte mal die Sache mit der Anzeige über den Cursor selbst ausprobieren (...er steht gerade auf der Grenzfrequenz...)



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. FFT (= Fast Fourier Transformation)

Oft interessiert nicht nur der zeitliche Verlauf der einzelnen Spannungen (oder Ströme) in der Schaltung. Sobald nämlich ihre Kurvenform nicht mehr sinusförmig ist, gibt es außer der Grundwelle noch „Oberwellen“, also Signale mit doppelter, dreifacher, vierfacher Frequenz usw. Die **FFT** berechnet ein solches Frequenzspektrum aus dem zeitlichen Verlauf des Signals.

Allerdings sind 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** bestimmt:

Maximaler Frequenzschritt = Startfrequenz = Linienbreite = $1 / \text{Simulationszeit}$

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“.

(Zusatzbemerkung für ganz obengenaue 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 wenig Perioden für die FFT benutzt werden oder der Timestep sehr grob ist).

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

- a) Er soll so klein sein, dass die simulierte Kurve nicht „eckig“ aussieht.
- b) Die zugehörige „**minimale Samplefrequenz = $1 / \text{Maximum Timestep}$** “ 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, mehr Samples für die FFT vorzugeben als tatsächlich simuliert wurden (...gibt sonst Probleme und gelogene Details beim Spektrum). Diese Anzahl echter Samples kann leicht durch die Beziehung

Echte Samplezahl = Simulationsdauer / Maximum Timestep

ermittelt und kontrolliert werden.

LTS spice 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

.options plotwinsize=0

abschalten.

Die für eine FFT verwendete **Anzahl an Samples muss immer eine Zweierpotenz** sein. Ist kein Problem, denn LTS spice listet uns automatisch die gemäß dieser Bedingung zugelassenen Samplemengen zur Auswahl in einem Menü auf. Unter immer muss man im Hinterkopf behalten: **Mehr Samples für die FFT geben zwar bessere Amplitudenauflösung und höhere Grenzfrequenz, erhöhen aber entsprechend die Rechenzeit und die Größe der Ergebnisdatei.....und setzen genügend „echte“ Samples voraus.**

Noch etwas für die Spezialisten: die Samplerate ist 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“. 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 benutzten Samples im Vergleich zu dem bei der Simulation angelegten Wertevorrat gewählt wird.

Doch jetzt ein praktisches Beispiel:

Wir verwenden den vorigen Tiefpass und speisen ihn mit einem Rechtecksignal (Frequenz = 1kHz, Minimalspannung = Null Volt, Maximalspannung = +1V).

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

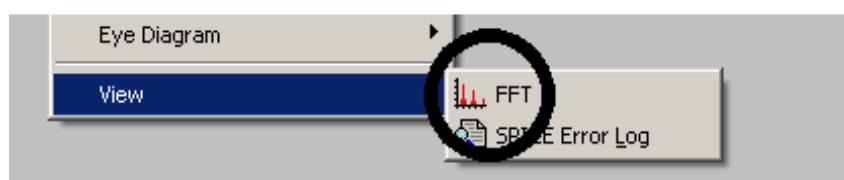
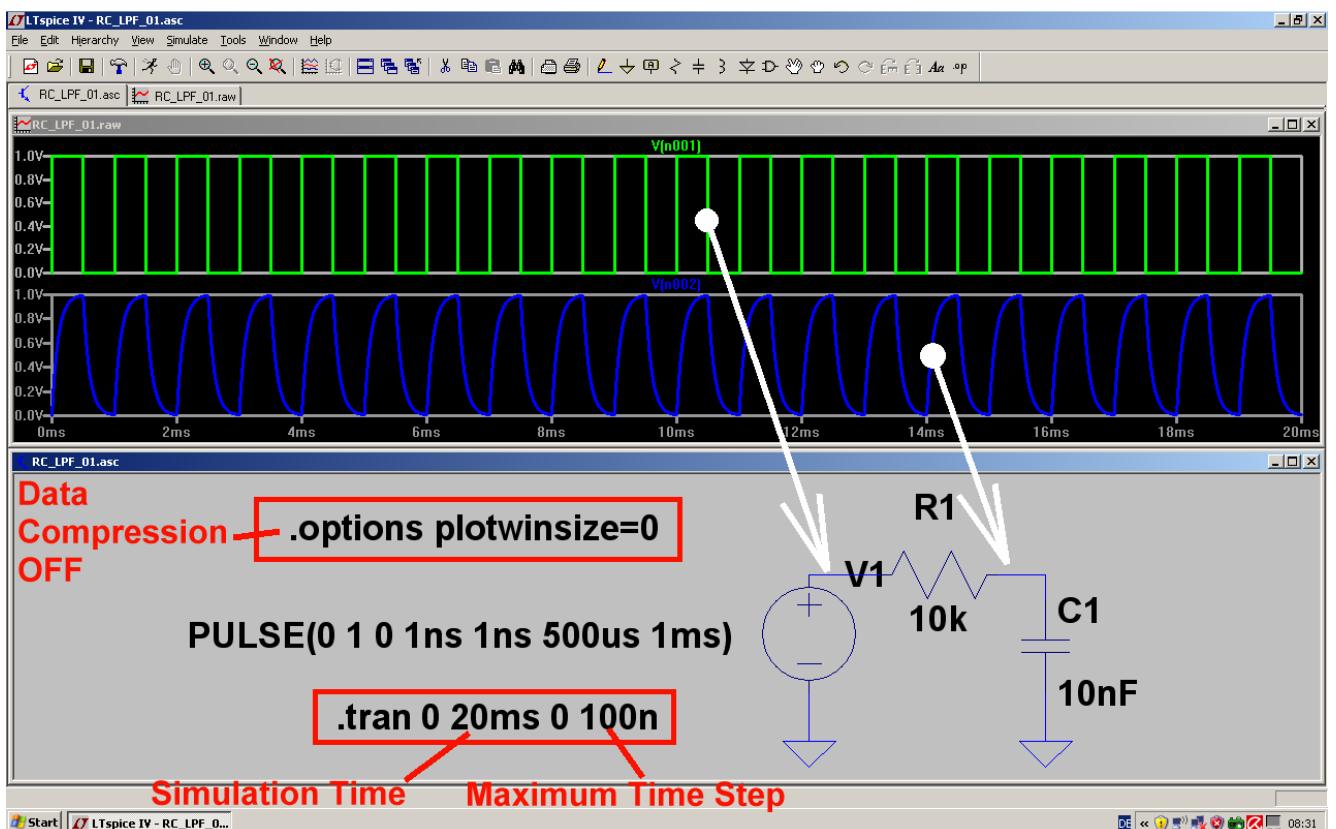
- Die Simulationszeit von 20ms ist ein ganzzahliges Vielfaches der Periodendauer (1ms).
- Der Maximum Timestep von 100ns ergibt eine minimale Samplefrequenz von $1/100\text{ns} = 10\text{MHz}$. Die Schmerzgrenze = Shannon-Grenze = höchste zulässige Oberwellenfrequenz würde dann bei 5MHz liegen.
- Die Anzahl der „echten Samples“ ergibt sich nun zu $20\text{ms} / 100\text{ns} = 200\,000$ Samples. Also können wir im FFT-Menü gefahrlos

„131 072 sampled data points in time“

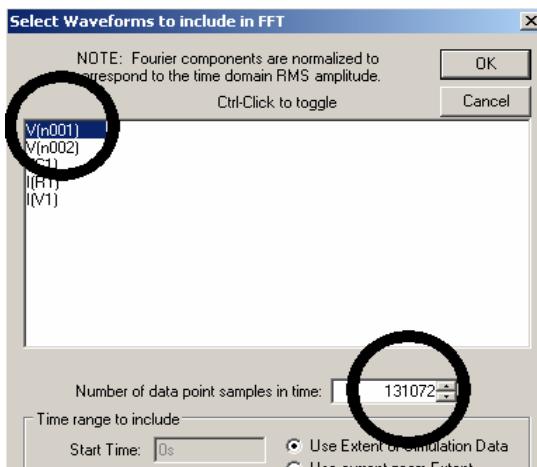
auswählen.

Jetzt zeichnen wir die Schaltung, programmieren die Spannungsquelle, legen Simulationszeit samt Timestep fest und schreiben die Direktive zum Abschalten der Datenkompression (..findet sich im EDIT-Menü).

Die Ergebnisdarstellung mit zwei Plot Panes erfolgt mit „Tile Horizontally“ Außerdem wird gleich auf „dicke Linien“ umgestellt. Das geht über „den Button mit dem Hammer“ auf die Karteikarte „Waveforms“. Dort steht schließlich „Plot data with thick lines“.



Nun klicken wir mit der **RECHTEN MAUSTASTE** auf die Rechteckspannung im Ergebnisdigramm und wählen (ganz unten) erst „View“, dann „FFT“ im Menü

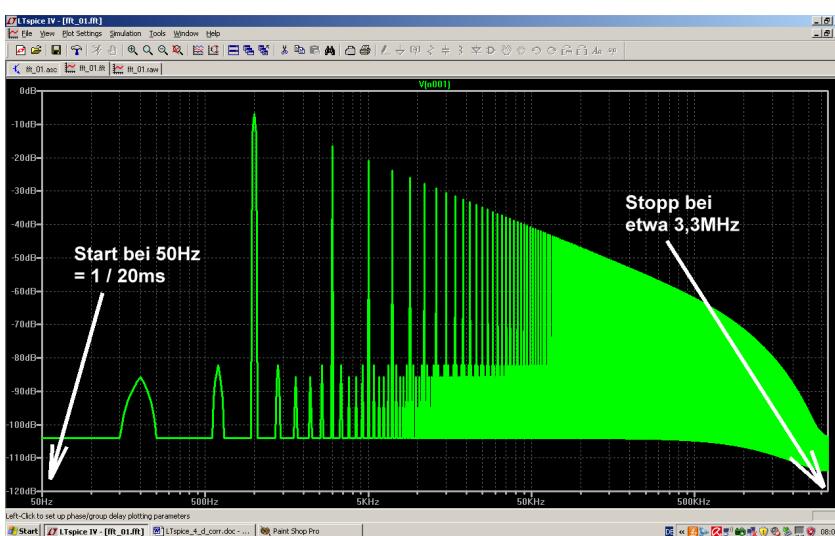


Dann aktivieren wir durch einen Mausklick die Anzeige der Eingangsspannung $V(n001)$.

Für die Berechnung des Spektrums stehen uns mindestens 200 000 echte Samples zur Verfügung. Also geben wir unbesorgt

131 072 Datenpunkte (= Samples)

vor und klicken dann OK.



Das ist der Erfolg. Die Startfrequenz und die Frequenzauflösung betragen:

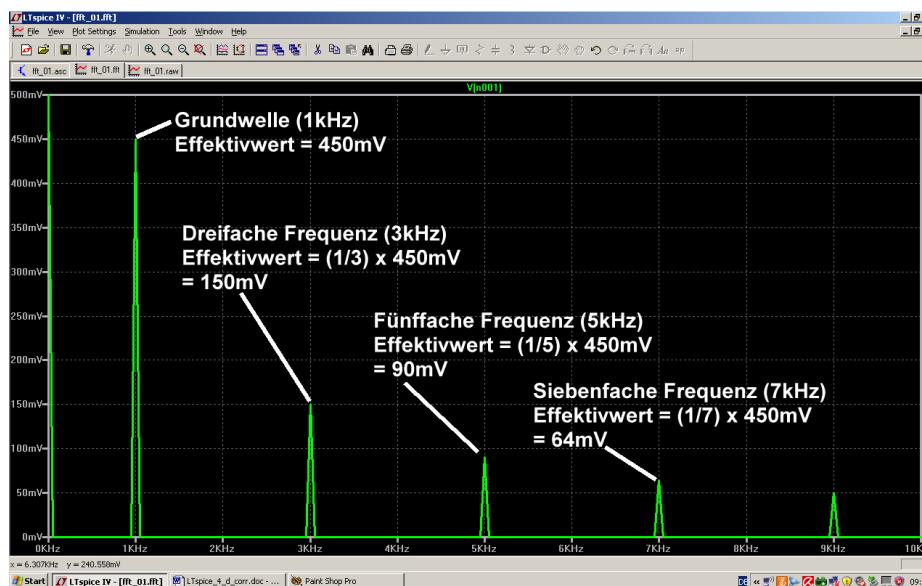
$$50\text{Hz} = 1/20\text{ms}$$

Eine höhere obere Frequenzgrenze beim Spektrum (und größeren Dynamikbereich) gibt es dagegen nur mit mehr Samples.

Doch jetzt wollen wir auf eine lineare Darstellung wechseln und darin kontrollieren, ob die Theorie recht hat. Sie behauptet:

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...10kHz mit einem Tick von 1kHz. Und Siehe da.....:

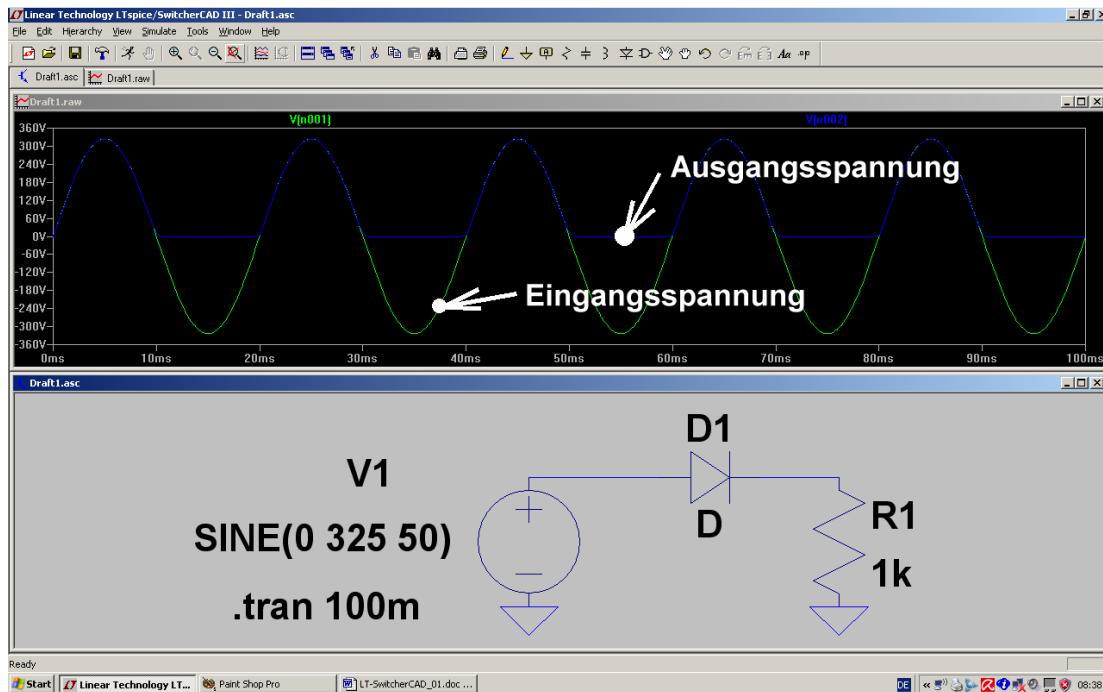
Übrigens:
Wer sich zusätzlich für die Phase interessiert, der klicke auf die

rechte senkrechte Achse und dort auf „Phase“.

6. Zweites Projekt: Gleichrichtung

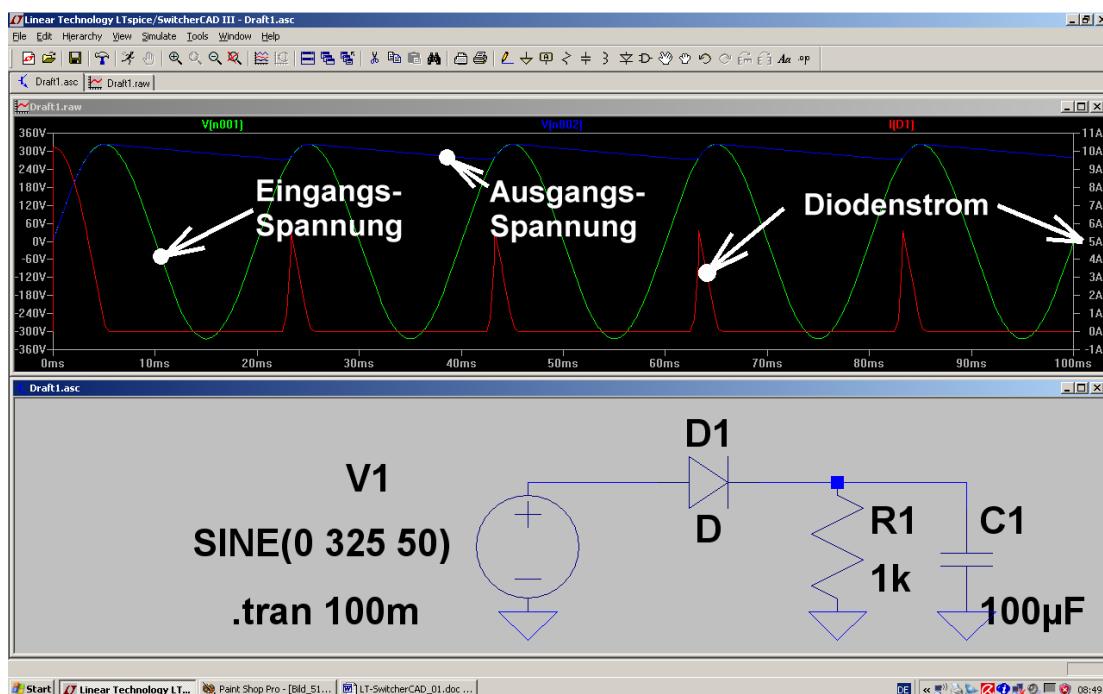
6.1. Einpuls-Gleichrichter ohne Trafo

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 dies 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).

Lösung:

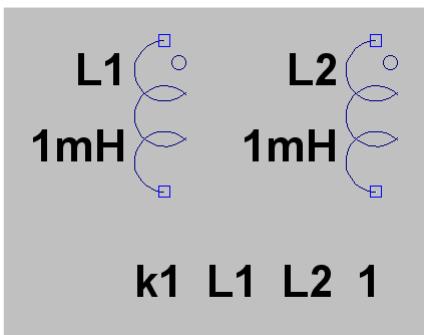


6.2. Eine wichtige Sache: ein Transformator für die Gleichrichterschaltungen

6.2.1. Die bequemste Lösung: simpler, idealer Trafo

Leider enthält die SwitcherCAD-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:

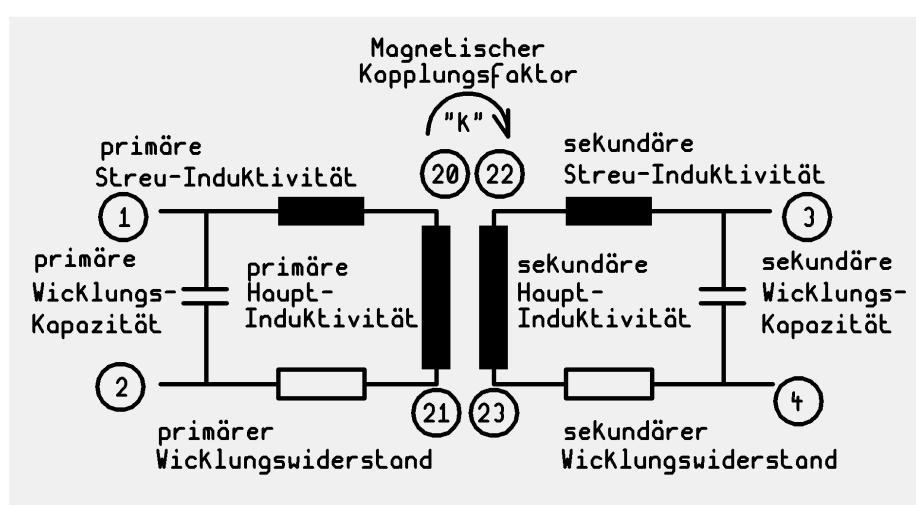
Eine direkte Eingabe des Übersetzungsverhältnisses ist bei SPICE nicht möglich, da immer nur mit den Bauteil-Eigenschaften (hier: Induktivitätswerte und magnetische Kopplung) simuliert wird. Also müssen wir das anders lösen und gehen auf die Induktivitäten los.
Dabei gilt folgendes:

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

und diese Formel stellen wir (wenn wir L1 vorgegeben haben und das Übersetzungsverhältnis kennen) einfach nach Lsek um. Ist also sehr einfach!

Nun soll als Nächstes mal 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).

6.2.2. 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:**

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 sieht das Ergebnis aus:

```
*  
* [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.lib** im Ordner

Programme / LTC / LTSpiceIV / lib / sub

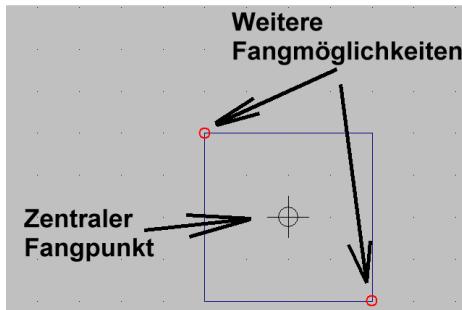
bei den schon vorhandenen übrigen SPICE-Modellen gespeichert. Bitte sorgfältig auf den Pfad UND die korrekte Endung achten...

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

6.2.3. Erzeugung eines passenden Symbols für diesen Transformator

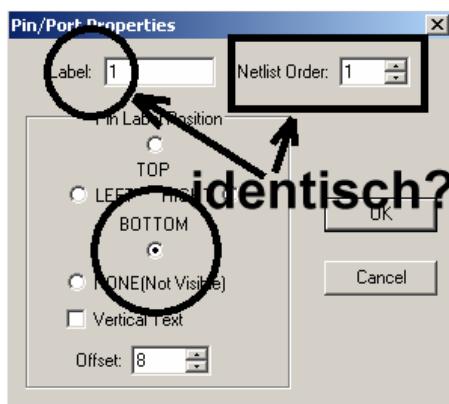
1. Schritt:

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.

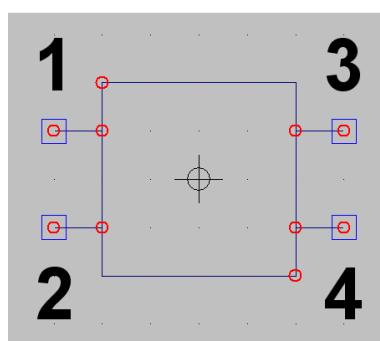


3. Schritt:

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.

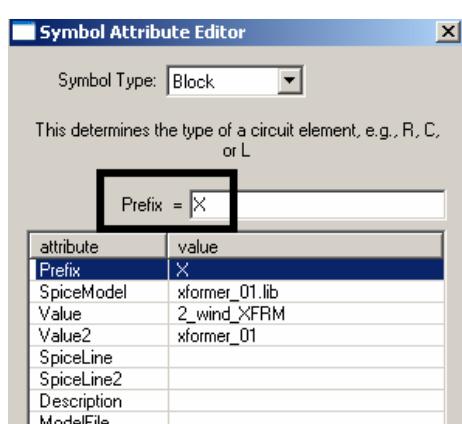
Bitte darauf achten, dass der Pin-Label und die Netlist-Order übereinstimmen!

Nach dem Klick auf OK hängt der Pin am Cursor und kann dann ca 5mm vom Symbol abgesetzt werden (Siehe nächstes Bild).



4. Schritt:

Nachdem wir alle 4 Pins erzeugt haben, öffnen wir nochmals „DRAW“ und zeichnen die Verbindungsleitungen von den Pins zum Symbol. So sollte am Ende das fertige Symbol aussehen.



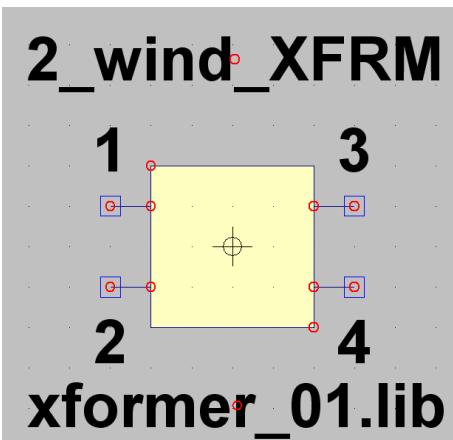
5.Schritt:

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 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.lib“ 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 „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.

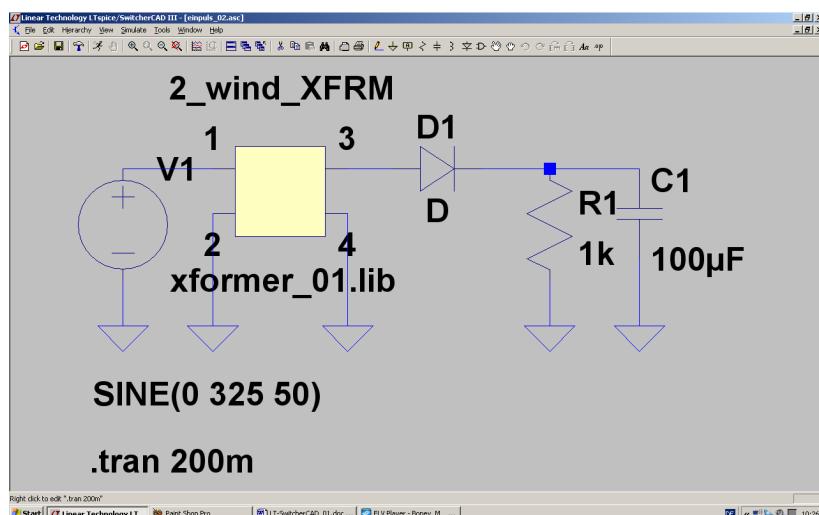
Tatsächlich alles geschafft!

6.3. Einpuls-Gleichrichter mit realistischem Trafo

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

Aufgabe:

Untersuchen Sie ein Netzteil mit Netztrafo, Gleichrichterdiode, Lastwiderstand $RL = 1\text{k}\Omega$ und Ladekondensator $C = 100\mu\text{F}$. 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 vom letzten Beispiel her. 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.

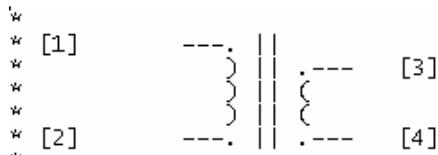
Dabei gilt folgendes:

$$\frac{L_{\text{Primär}}}{L_{\text{Sekundär}}} = (\ddot{u})^2$$

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

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

benutzen.



Für einen fast idealen Trafo kann man dann sowohl die Wicklungs-Widerstände, die Streu-Induktivitäten und die Wicklungs-Kapazitäten stark verkleinern.

Aber man darf sie nicht ganz auf Null setzen!

```

*.SUBCKT xformer_01 1 2 3 4
*
* Primary
*
Lleak1 1 20 1uH
Rpri1 20 21 1H
Rpri1 21 2 1
Cpri1 1 2 1pF

* Secondary
*
Lleak3 3 22 1uH
Lsec1 22 23 2.5mH
Rsec1 23 4 1
Csec1 3 4 1pF

K Lpri1 Lsec1 0.999
.ENDS

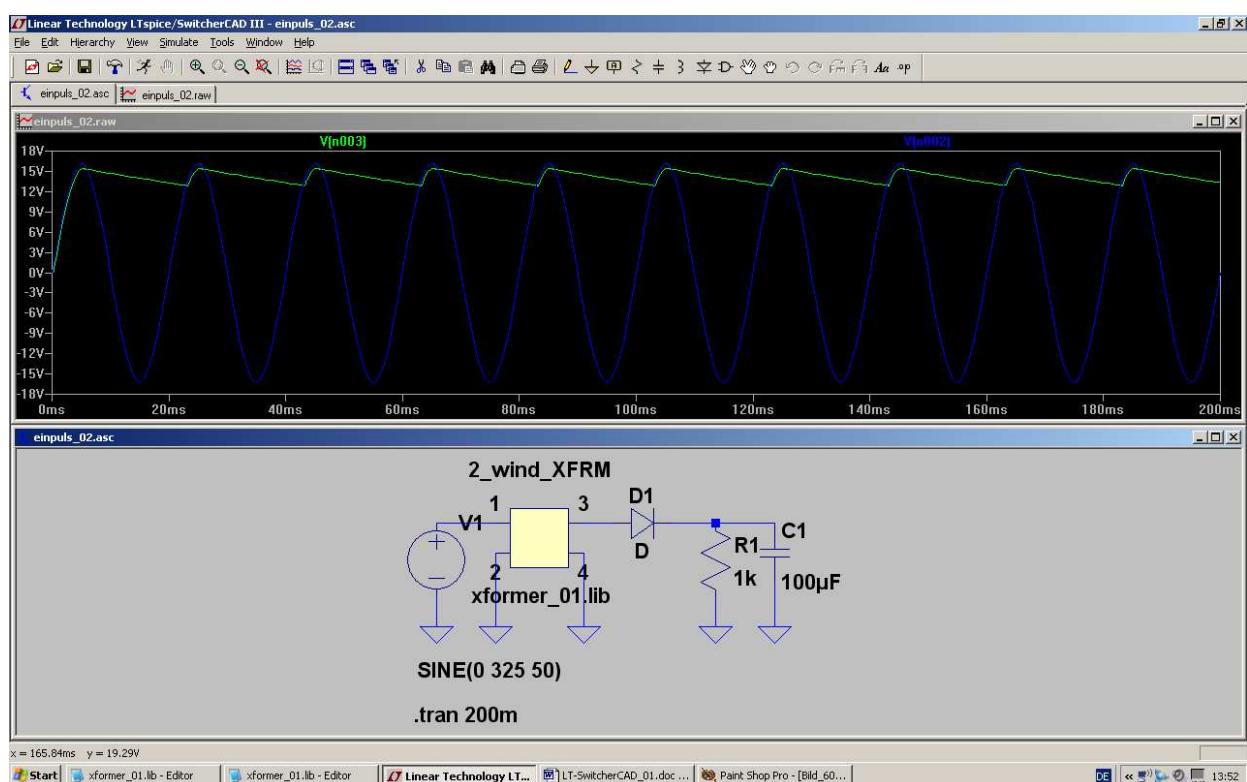
```



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

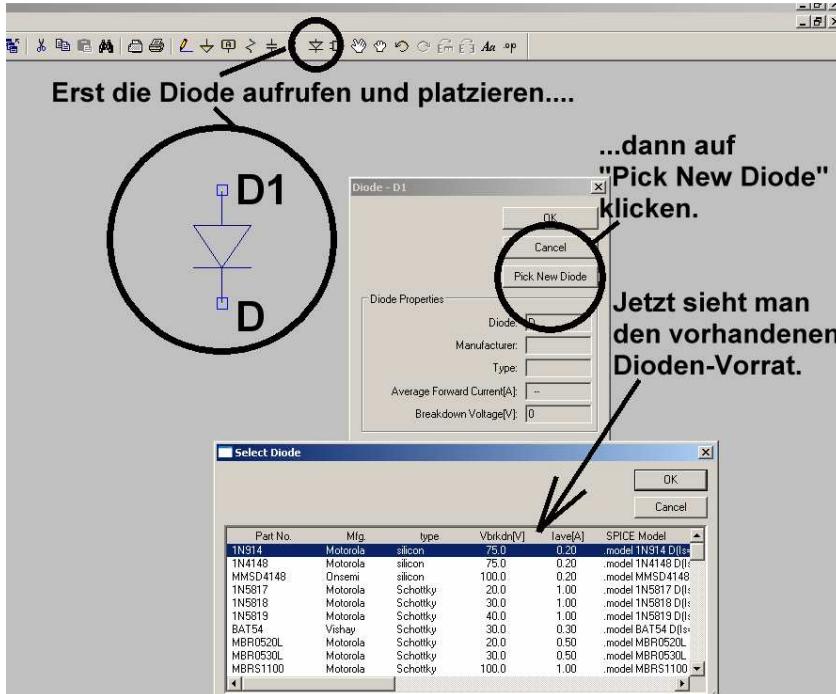
So sieht dann das „xformer_01.lib“-File aus, das wir anschließend wieder am korrekten Ort (= Ordner „sub“ im „lib“-Verzeichnis von LTSpiceIV) abspeichern müssen:

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



6.4. Verwendung der Diode 1N4007 in der Gleichrichterschaltung

Wir wollen nun anstelle der idealen Diode den bekannten Diodentyp „**1N4007**“ einsetzen. Leider ist dieses SPICE-Modell nicht in der mitgelieferten Bibliothek enthalten und deshalb üben wir auf diese Weise das Einbinden fremder Modelle.



Ein Tipp: wer wissen möchte, welche Diodentypen bereits beim Programm im Bauteilvorrat mitgeliefert werden, der verfährt nach der nebenstehenden Methode.

Aber jetzt wollen wir die Sache mit der 1N4004 anpacken.

1. Schritt:

Im Internet suchen wir (z. B. mit Google) nach der Library „**diode.lib**“. Sie enthält (als OrCad-Bibliothek) viele Netzgleichrichter.

Aber Vorsicht:

Diese Library kommt bei uns als HTML-File an und ist in dieser Form noch nicht zu gebrauchen!

Da bleibt nur ein Weg: erst den Inhalt mit **<STRG> + <a>** komplett markieren und mit **<STRG> + <c>** in die Zwischenablage kopieren. Dann den Text-Editor (z. B. Notepad) öffnen und die Zwischenablage mit **<STRG> + <v>** in ein neues Blatt einfügen.

Erst jetzt können wir das File als „**diode.lib**“ im Ordner

LTC / LTSpiceIV / lib / sub

speichern. (Bitte aber dabei auf die korrekte Endung achten und im Editor vorher beim File-Typ „**Alle Dateien**“ wählen -- sonst wird nämlich zusätzlich die Endung „**.txt**“ angehängt und dann findet unser Programm die Datei nicht.....)

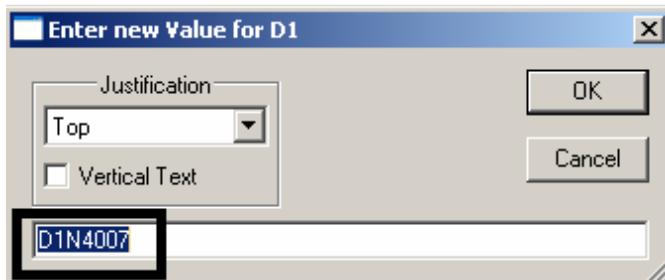
```
*$  
.model D1N4004 ako:D1N4001 D(Bv=600) ; use  
*$  
.model D1N4005 ako:D1N4001 D(Bv=900) ; use  
*$  
.model D1N4006 ako:D1N4001 D(Bv=1200) ; use  
*$  
.model D1N4007 ako:D1N4001 D(Bv=1500) ; use  
*$  
.model D1N4009 D(Is=544.7E-21 N=1 Rs=.1 Ikf=  
+ Vj=.75 Fc=.5 Isr=30.77n Nr=2  
*$
```

2. Schritt:

Mit dem Text-Editor (z. b. Notepad) öffnen wir nochmals das File „**diode.lib**“ und suchen nach „**1N4007**“. Dabei stellen wir fest, dass dieses Modell die Bezeichnung

D1N4007

aufweist!



3. Schritt:

Nun fahren wir in unserem Schaltplan mit dem Cursor auf die untere Angabe „D“ beim Diodensymbol und klicken mit der rechten Maustaste darauf.

Jetzt können wir als neuen Wert „D1N4007“ eintragen.

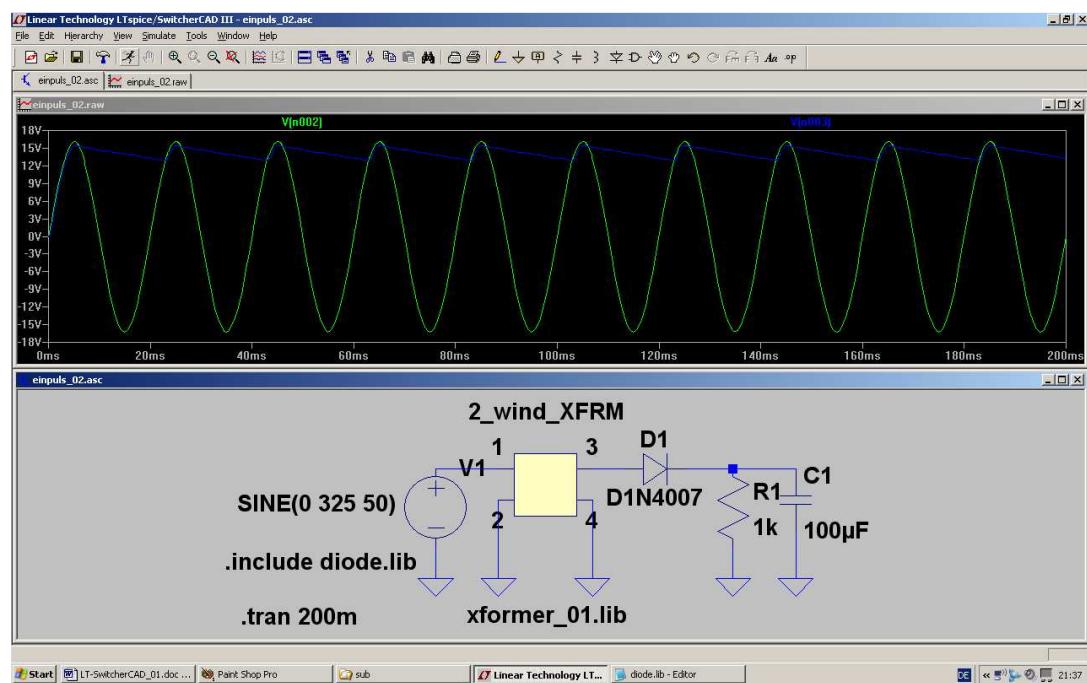
4. Schritt:

Aber das ist noch nicht alles. Damit die Einbindung und Zuweisung richtig funktioniert, müssen wir „Edit“ und „SPICE Directive“ aufrufen. Wir tippen ein:

.include diode.lib

und setzen diese Zeile irgendwo in unserem Schaltplan ab.

5. Schritt:



Jetzt läuft die Simulation korrekt mit dem neuen Diodenmodell.

6.5. Zweipuls-Gleichrichter mit realistischem Trafo

Hier ist ein neues Bauteil erforderlich: ein Transformator mit zwei getrennten Sekundärwicklungen. Und wir wollen wieder nicht den einfachen breiten, sondern den steinigen Weg beim Trafomodell wählen...

```

* --- [3]
*   } | .--- [4]
*   } | .--- [5]
* [1] ---. | | .--- [6]
*   } |
*   } |
* [2] ---. | | .--- [6]
*   } |
*   } |
* --- [3]
* --- [4]
* --- [5]
* --- [6]

.SUBCKT xformer_02 1 2 3 4 5 6
* Primary
*Lleak1 1 20 1uH
Lpri1 20 21 1H
Rpri1 21 2 1
Cpri1 1 2 1p
* Secondary 1
*Lleak3 3 22 1uH
Lsec1 22 23 2.5mH
Rsec1 23 4 1
Csec1 3 4 1p
* Secondary 2
*Lleak5 5 24 1uH
Lsec2 24 25 2.5mH
Rsec2 25 6 1
Csec2 5 6 1p
K Lpri1 Lsec1 Lsec2 0.999
.ENDS

```

1. Schritt:

Also gehen wir in den Texteditor, rufen unser File „xformer_01.lib“ auf und ändern es ab.

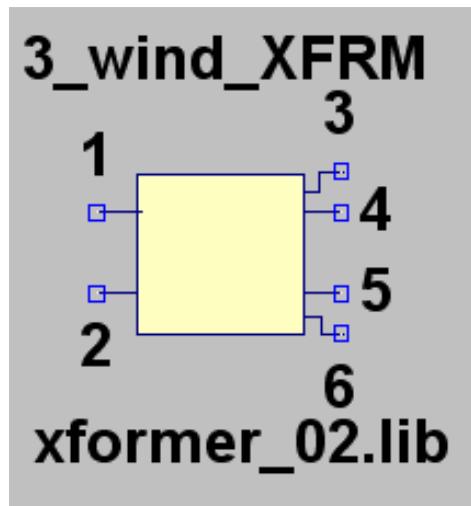
Wir sehen gleich wieder ein Übersetzungsverhältnis von $\bar{U} = 20:1$ vor und tragen die zu einer Primär-Induktivität von 1H gehörenden Sekundär-Induktivitäten von je $1H / 400 = 2,5mH$ in die entsprechenden Zeilen ein.

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

Bitte nicht vergessen:

Wir haben es nun mit **drei Induktivitäten zu tun, die miteinander verkoppelt sind** und müssen das im File angeben.

Stimmt alles, dann speichern wir dieses neue File als „**xformer_02.lib**“ wieder im „sub“-Ordner der Library.



2. Schritt:

Auch ein neues Symbol ist nötig. Die Erstellungs-Prozedur läuft exakt so ab wie im vorigen Beispiel (Kapitel 5.2.2.) und am Ende muss der Übertrager so aussehen.

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

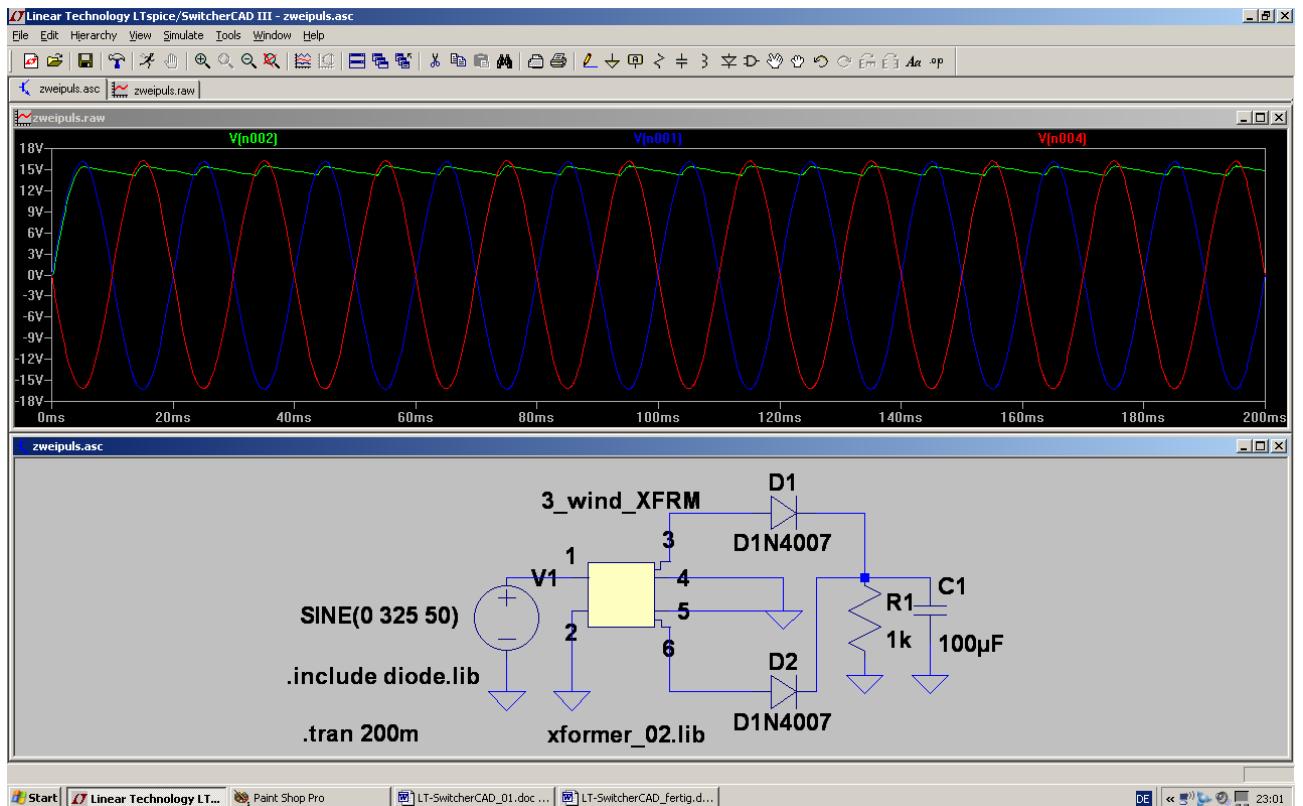
Warnung:

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....

3. Schritt:

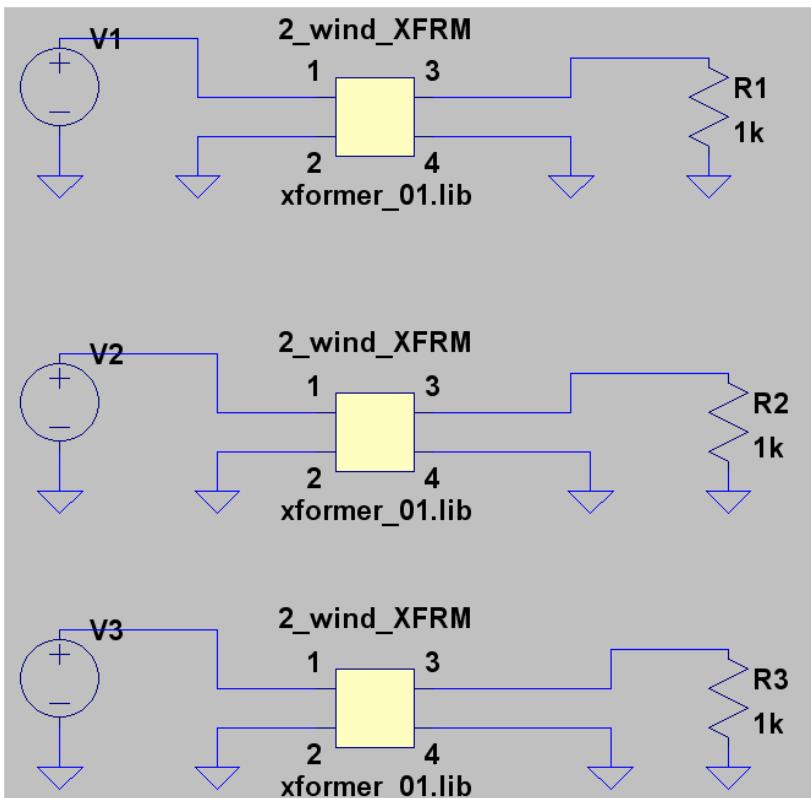
Nun wird die „Zweipuls-“ (= Mittelpunkts-) Schaltung unter Verwendung dieses Übertragers sowie 2 Dioden vom Typ 1N4007 gezeichnet und alles simuliert.

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



7. Drittes Projekt: Drehstrom

7.1. Programmierung eines Drehstromsystems



Wir wollen uns mit drei Spannungsquellen, drei realistischen Trenntransformatoren und drei Lastwiderständen die Sache mit dem Drehstrom etwas näher ansehen.

Beim Transformator verwenden wir die schon bekannte Datei

„xformer_01.lib“,
aber wir machen beide Hauptinduktivitäten gleich und erzielen so ein Übersetzungsverhältnis von $ü = 1:1$.

Diese Bauteile werden geholt und in der nebenstehenden Form verdrahtet.

Independent Voltage Source - V2

Functions

- (none)
- PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)
- SINE(Voffset Vamp Freq Td Theta Phi Ncycles)
- EXP(V1 V2 Td1 Tau1 Td2 Tau2)
- SFFM(Voffset Vamp Fcar MDI Fsig)
- PWL(t1 v1 t2 v2...)
- PWL FILE:

DC offset[V]:	0
Amplitude[V]:	325
Freq[Hz]:	50
Tdelay[s]:	6.66666ms
Theta[1/s]:	0
Phi[deg]:	0
Ncycles:	

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:

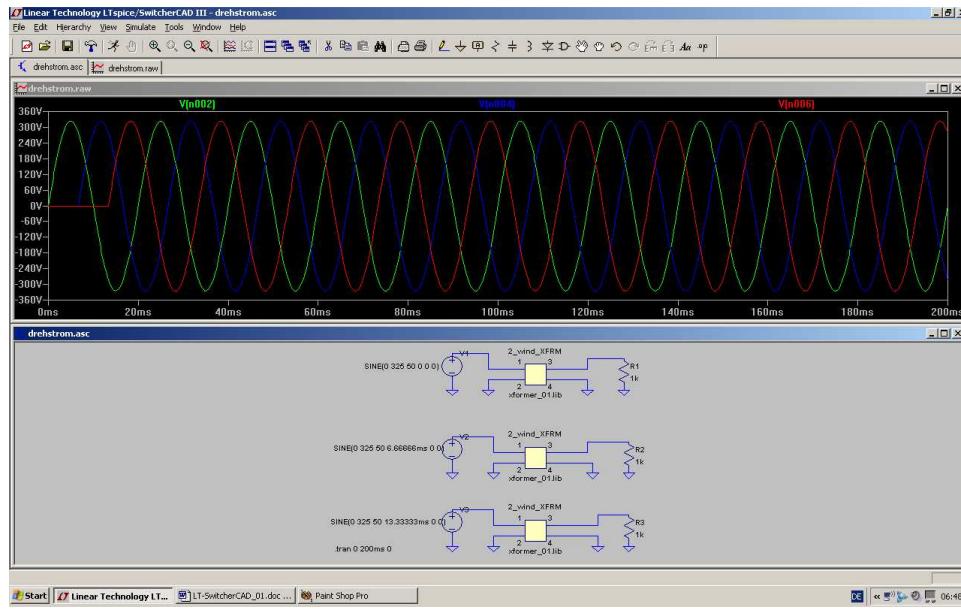
Die beim Drehstrom erforderliche Phasenverschiebung von 120 Grad zwischen den einzelnen Kurven erreichen wir durch **drei unterschiedliche Startverzögerungen** (= Tdelay)!

Bei V1 beträgt sie Null Millisekunden

Bei V2 beträgt sie (siehe Bild) 6,66666 ms

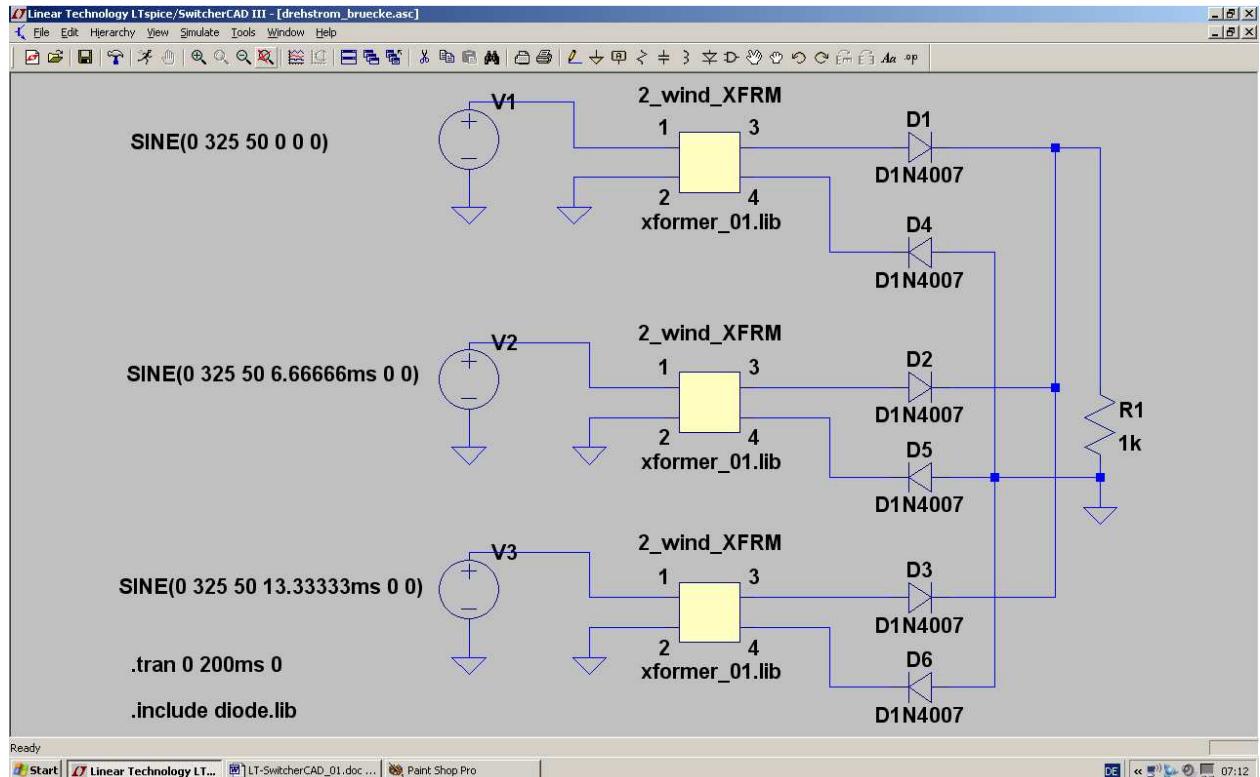
Bei V3 beträgt sie 13,33333ms

Programmieren wir nun noch eine Simulationsdauer von 200 ms , dann sieht das Simulationsergebnis so aus:

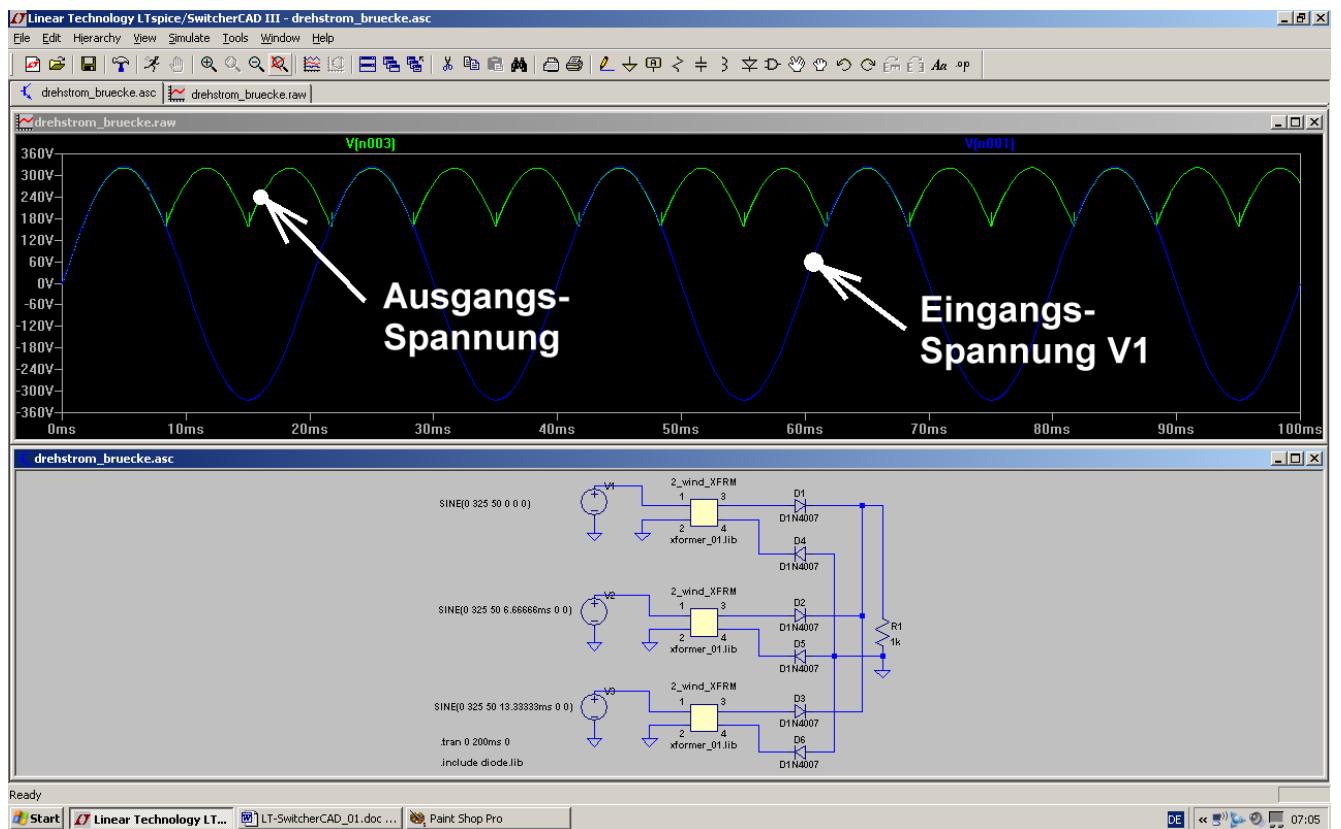


7.2. Drehstrom-Lichtmaschine im Auto

Wir speichern die vorige Schaltung als neues Projekt und geben ihr einen anderen Namen. Dann ändern wir die Schaltung ab und brauchen dazu zusätzlich **6 Dioden vom Typ 1N4007**. Zwei der drei Widerstände entfallen und dann können wir die Schaltung neu verdrahten.



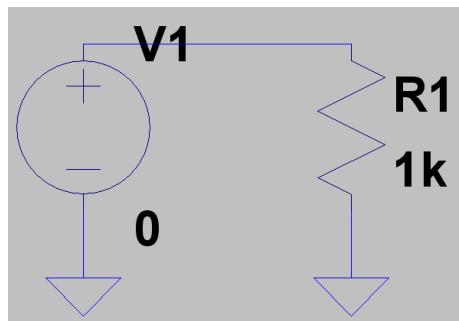
So sieht das Ergebnis der Simulation aus, wenn wir nur die Eingangsspannung V1 sowie die Ausgangsspannung am Lastwiderstand darstellen lassen:



8. Viertes Projekt: Darstellung von Bauteil-Kennlinien

8.1. Ohm'scher Widerstand

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

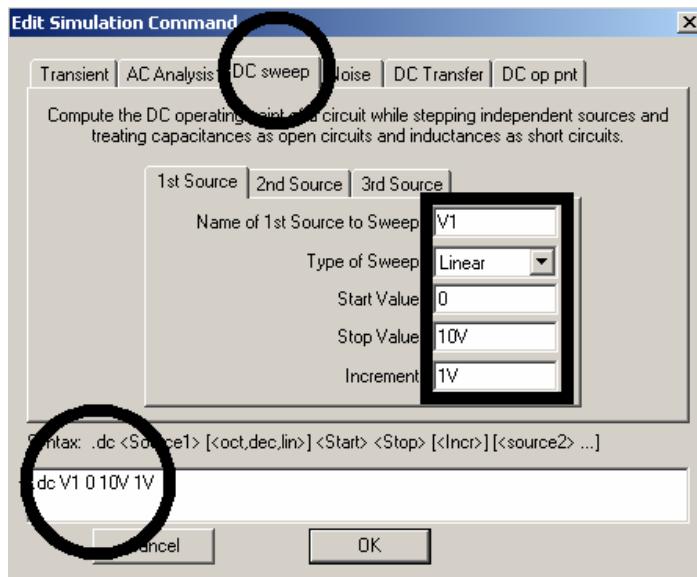


Dazu öffnen wir ein neues Projekt („New Schematic“) und zeichnen uns die nebenstehende Schaltung.

Beim Widerstand „R1“ wählen wir einen Wert von $1\text{k}\Omega$.

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.

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

Startwert = Null Volt

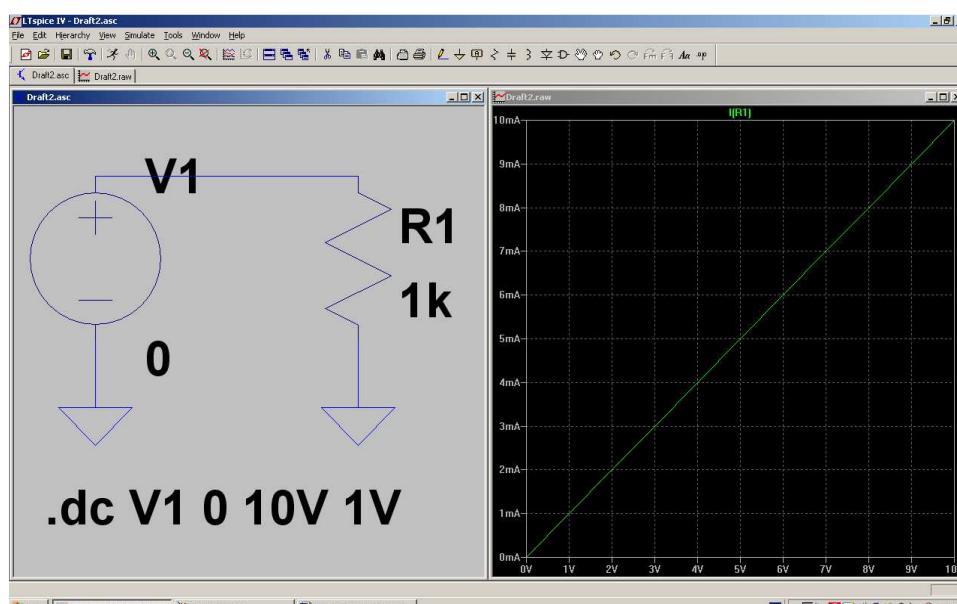
Endwert = 10 Volt

Schrittweite = 1V

Linearer Sweep

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

„.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...

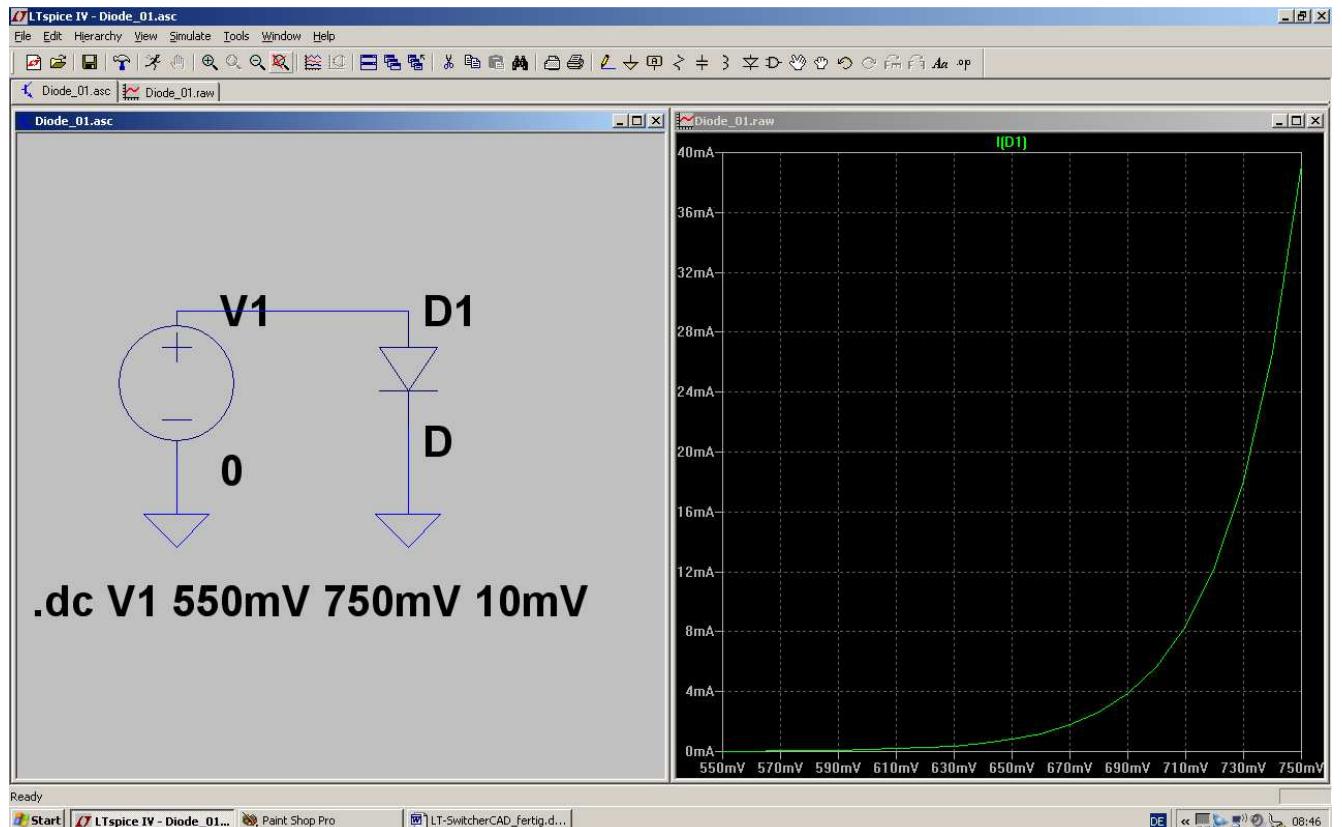
8.2. Diode

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 noch das Simulation Command umändern in

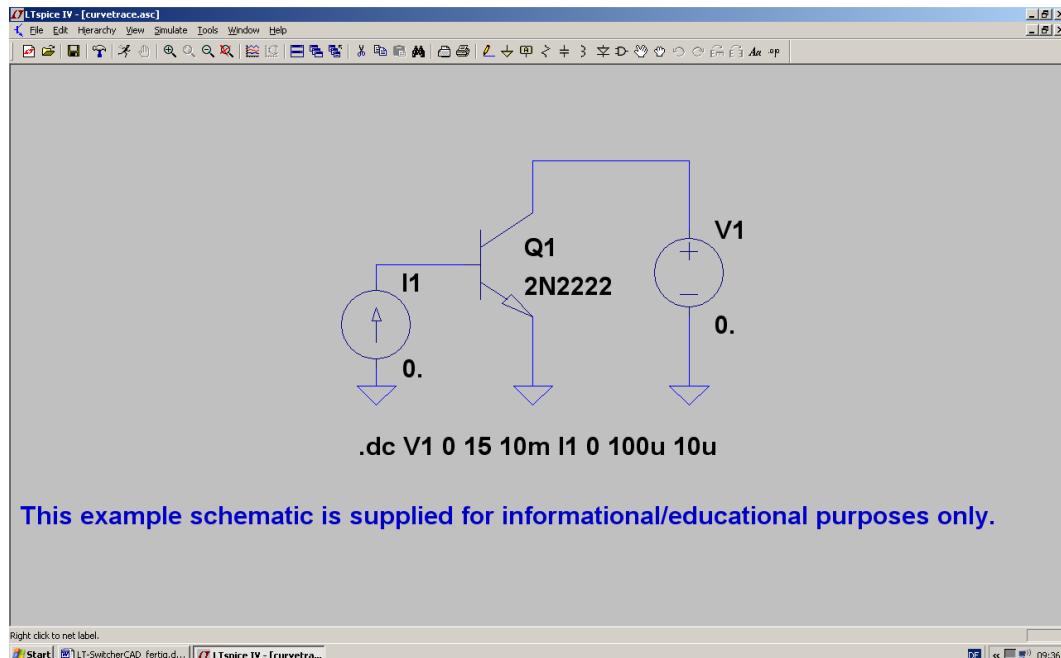
.dc V1 550mV 750mV 10mV

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



8.3. NPN-Transistor

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



Man erkennt:

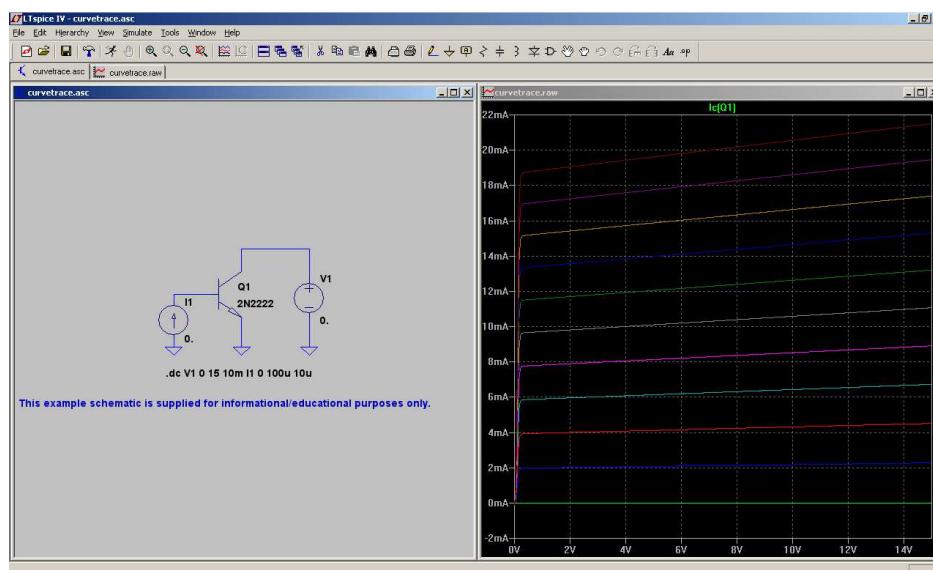
- Am Kollektor des Transistors wird die Gleichspannungsquelle V1 angelegt. Sie liefert zunächst Null Volt.
- Eine Konstantstromquelle I1 speist die Basis des Transistors. Auch ihr Startwert ist „Null“

Das Simulation Command

.dc V1 0 15 10mV I1 0 100u 10u

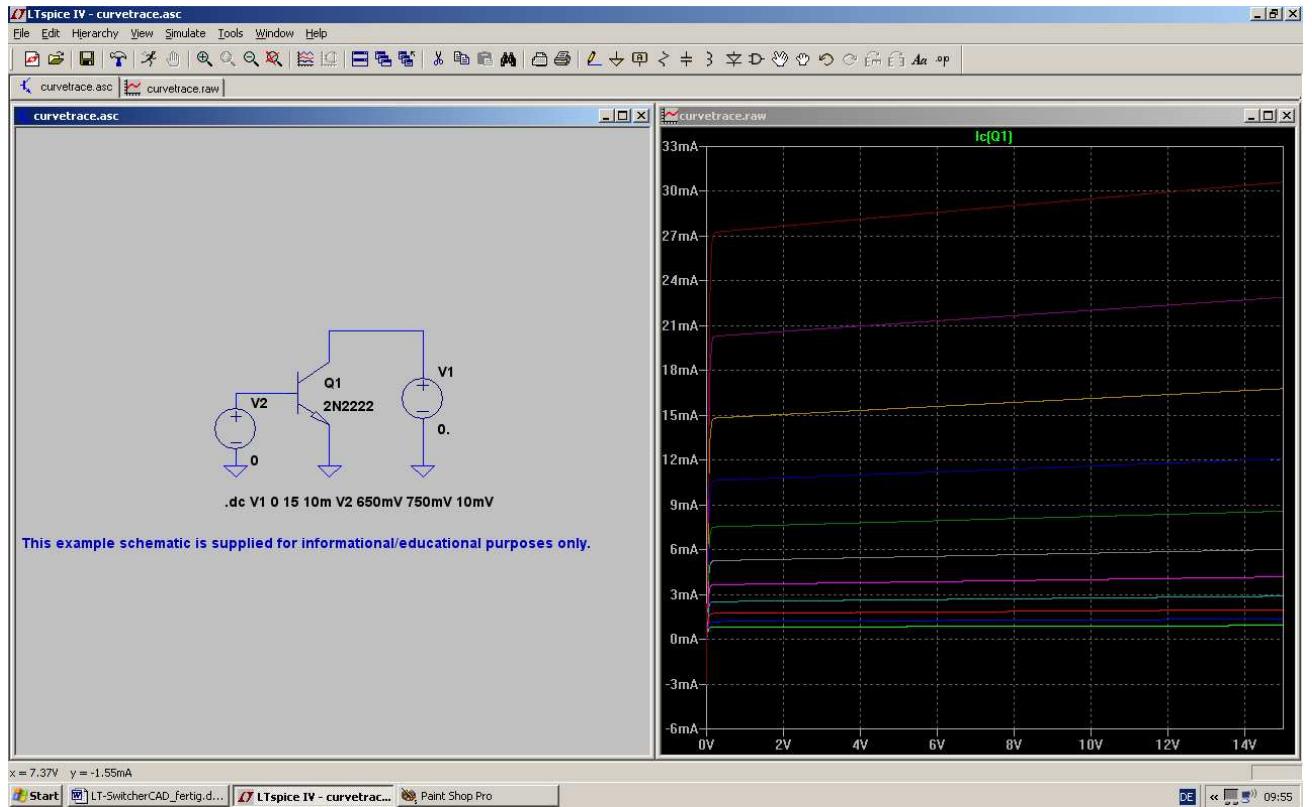
besagt:

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 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 verfasst:

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

Die Angaben für die Kollektorspannung V_1 bleiben darin unverändert. Die Basis-Spannungsquelle V_2 wird in Schritten von 10mV von 650mV auf 750mV geändert.

Ergebnis: Siehe oben...

8.4. N-Kanal-Sperrsicht-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

BF245B

Nach ihm sucht man allerdings im mitgelieferten Vorrat vergeblich. (Wissen Sie noch, wie das ging? Erst den FET als „njf“ holen und absetzen. Dann ein rechter Mausklick auf das Schaltzeichen und unter „Pick new JFET“ nachsehen)

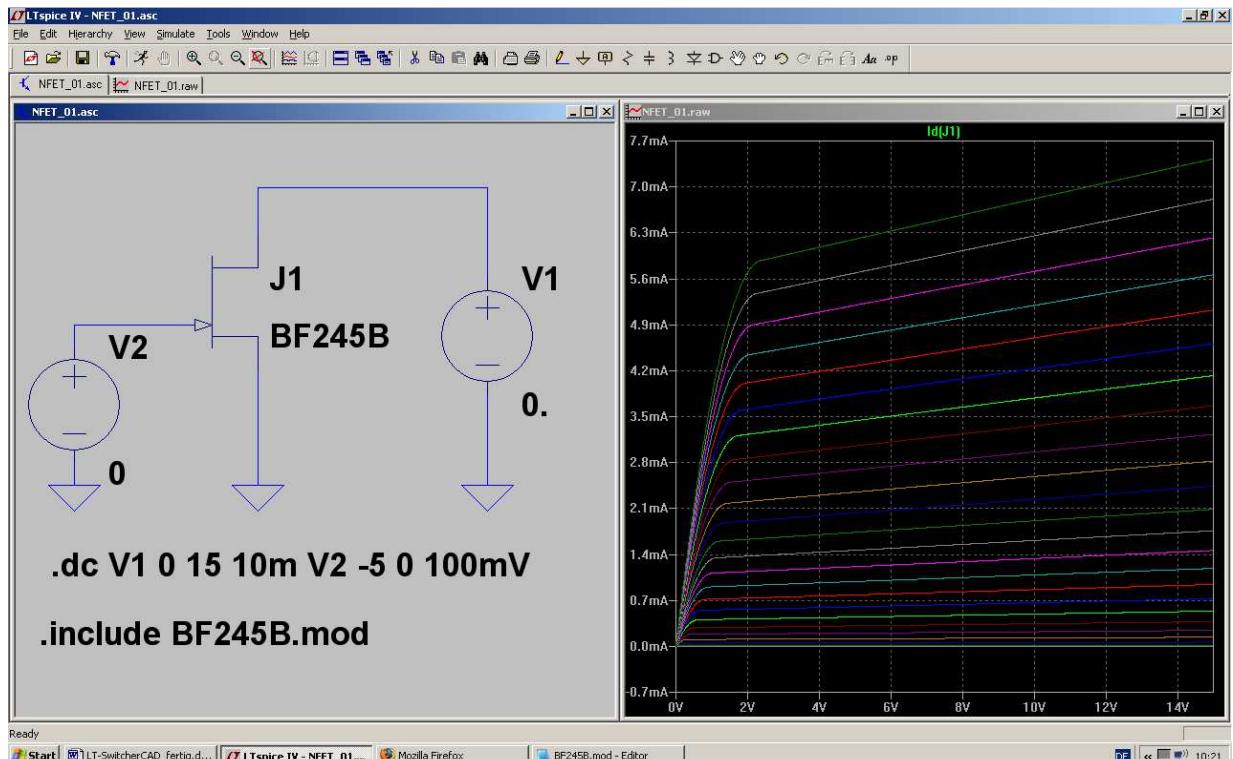
Also geht es ins Internet mit der Such-Eingabe

bf245b spice model.

Das Ergebnis wird als Text in die Zwischenablage und dann in ein neues Blatt des Texteditors kopiert. Von dort aus kann es endgültig als

BF245B.mod

in unserer **Bibliothek im Ordner „sub“** abgespeichert werden (...Bitte aber auf die korrekte Endung achten....) Nun sehen wir uns mal an, was noch zu erledigen ist:



Erst wird der FET (= „njf“) in die Schaltung eingebaut und dann mit der rechten Maustaste auf die Modellangabe geklickt (= Zeile „NJP“ rechts unten beim Symbol). Die Modellbezeichnung wird auf **BF245B** abgeändert und dann (über „Edit“) eine **zusätzliche SPICE-Direktive** geschrieben. Sie lautet:

.include BF245B.mod

und damit wird das aus dem Internet stammende Spice-Modell des BF245B dem Simulationsteil zur Verfügung gestellt.

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

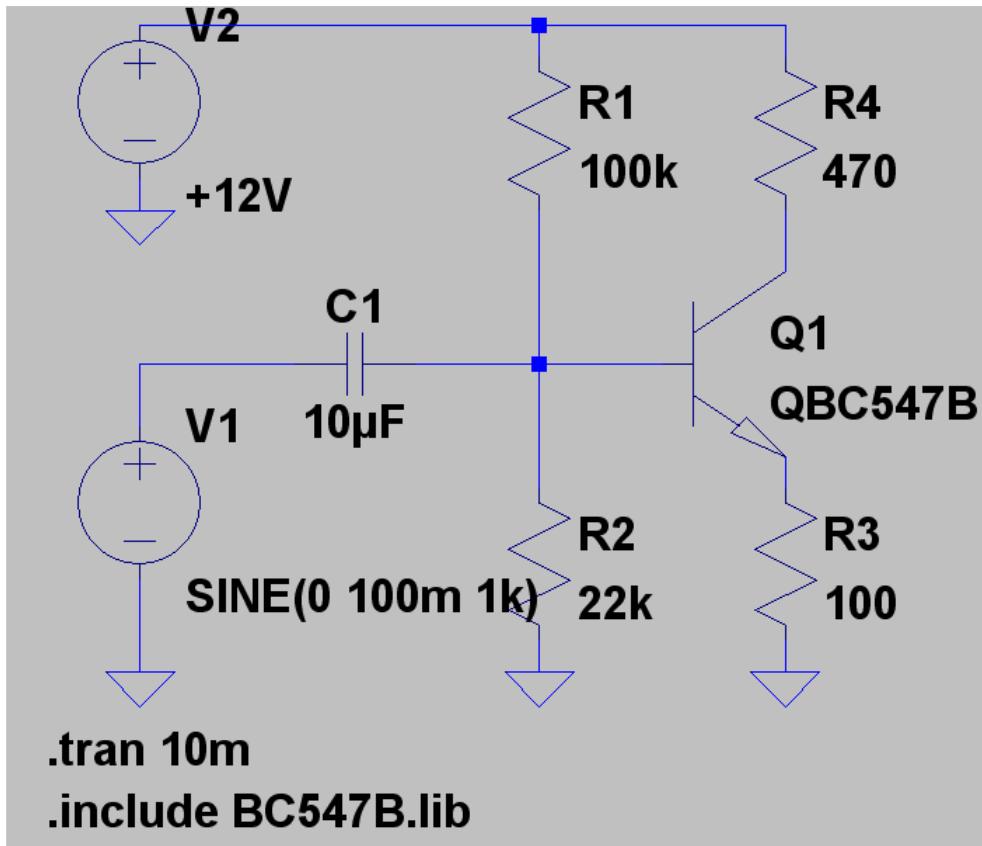
-5V bis 0V in Schritten von 100mV

einstellt, dann sollte alles klappen.

9. Fünftes Projekt: Schaltungen mit Transistoren

9.1. Einstufiger Verstärker

9.1.1. Ansteuerung mit einem Sinus-Signal



Anleitung:

1. Schritt:

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

2. Schritt:

Den Widerständen und dem Kondensator werden die korrekten Werte zugewiesen.

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....10 Millisekunden vorgesehen.

4. Schritt:

Im Internet suchen wir nach „**BC547B.lib**“. Der Inhalt wird markiert, in die Zwischenablage kopiert, in ein neues Blatt eines Texteditors eingefügt und schließlich als „**BC547B.lib**“ im Ordner

LTC / LTSpiceIV / lib / sub

gespeichert

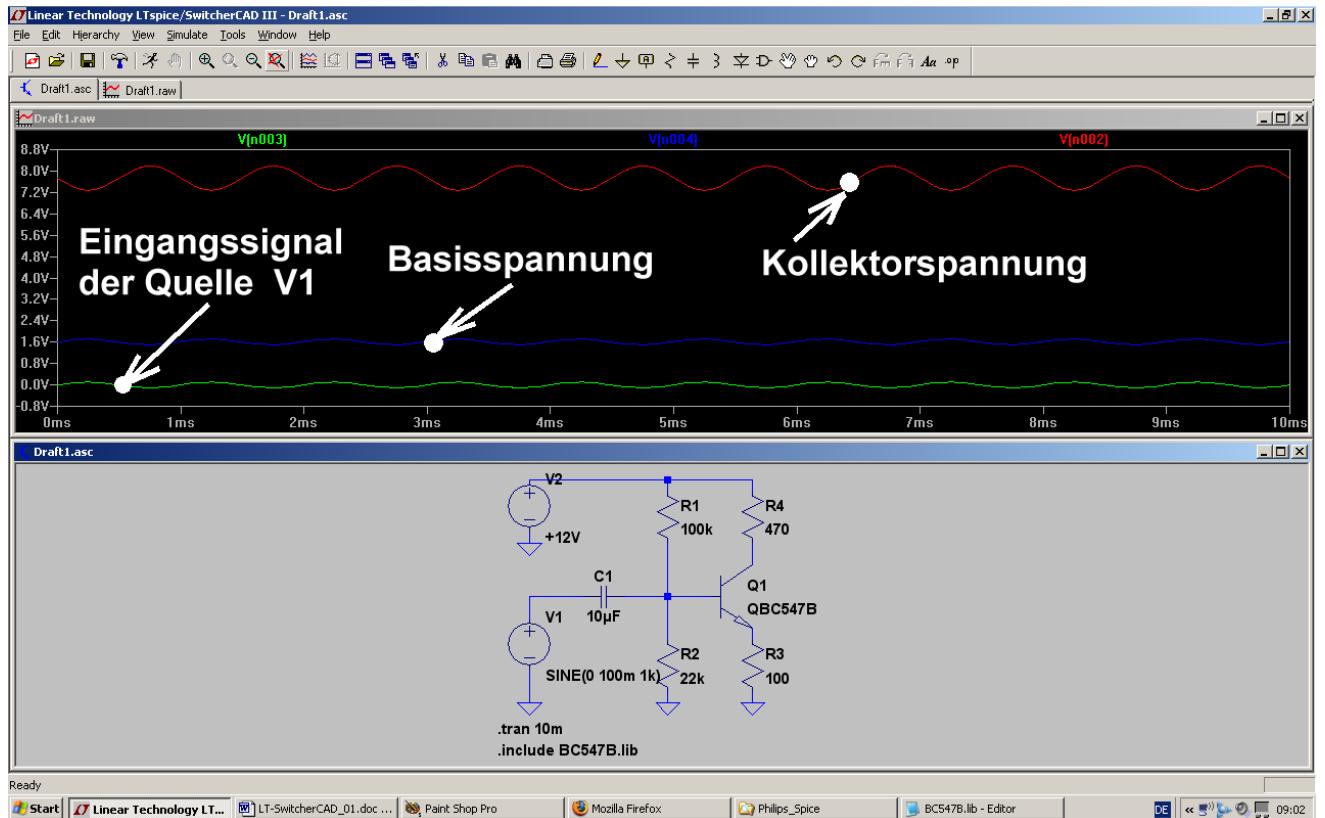
5. Schritt:

Wir stellen den Cursor auf die untere Angabe „**NPN**“ beim Transistorsymbol, klicken rechts, tragen die nun gültige Bezeichnung „**QBC547B**“ ein und schreiben noch das Spice-Command **.include BC547B.lib**. Fertig!

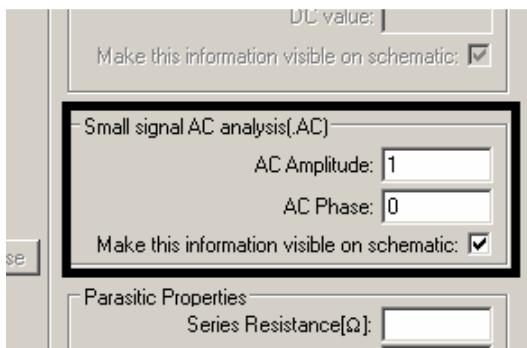
Beim Simulationsergebnis schauen wir uns die Eingangsspannung, die Basisspannung des Transistors und seine Kollektorspannung an. Besonders die Phasenumkehr im Transistor können wir nun sehr schön sehen. Außerdem lässt sich nun die Behauptung der Theorie, das die Verstärkung der Stufe etwa den Wert

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

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

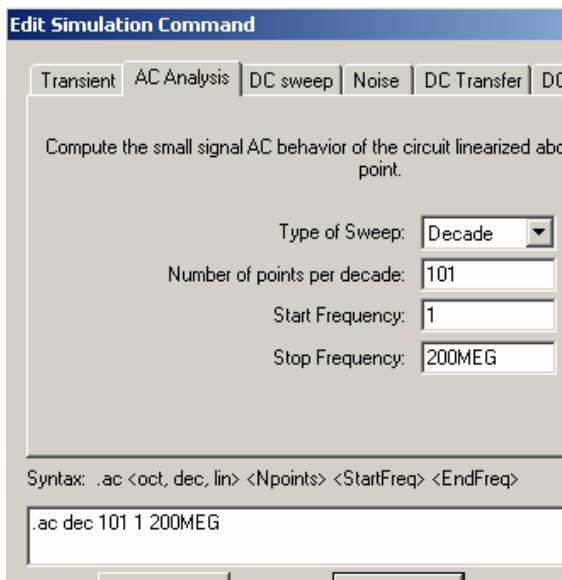


9.1.2. Simulation des Frequenzganges („AC-Sweep“)



1. Schritt:

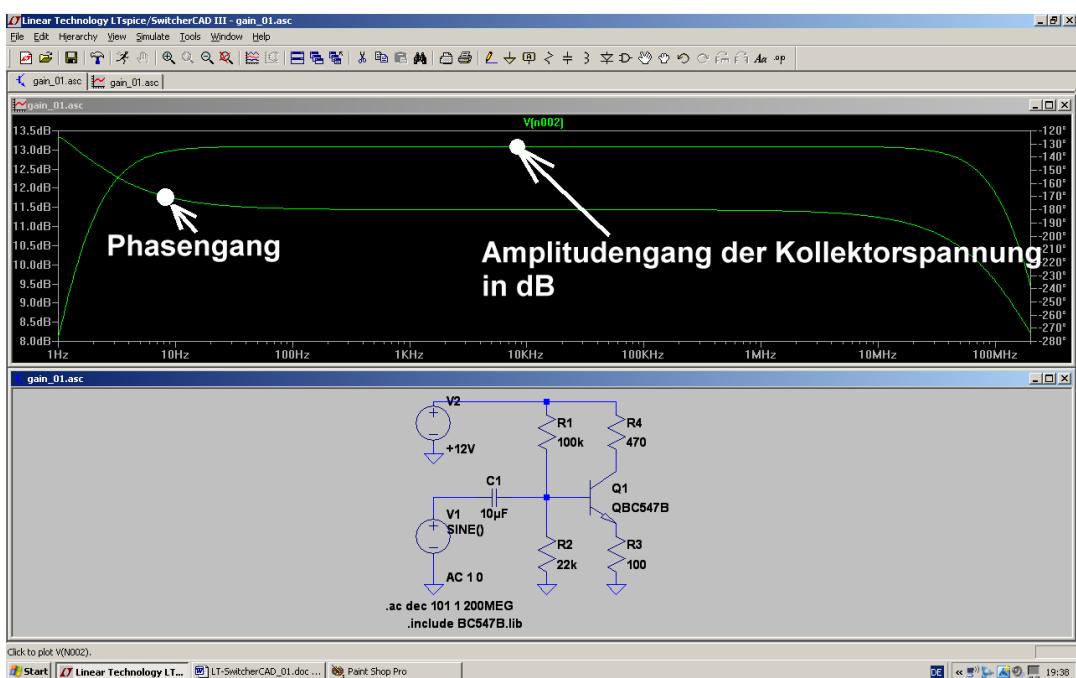
Wir stellen die Spannungsquelle V1 auf „Sweep-Betrieb“ um.



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 101 Punkte pro Dekade.

So sollte das Simulationsergebnis schließlich aussehen.



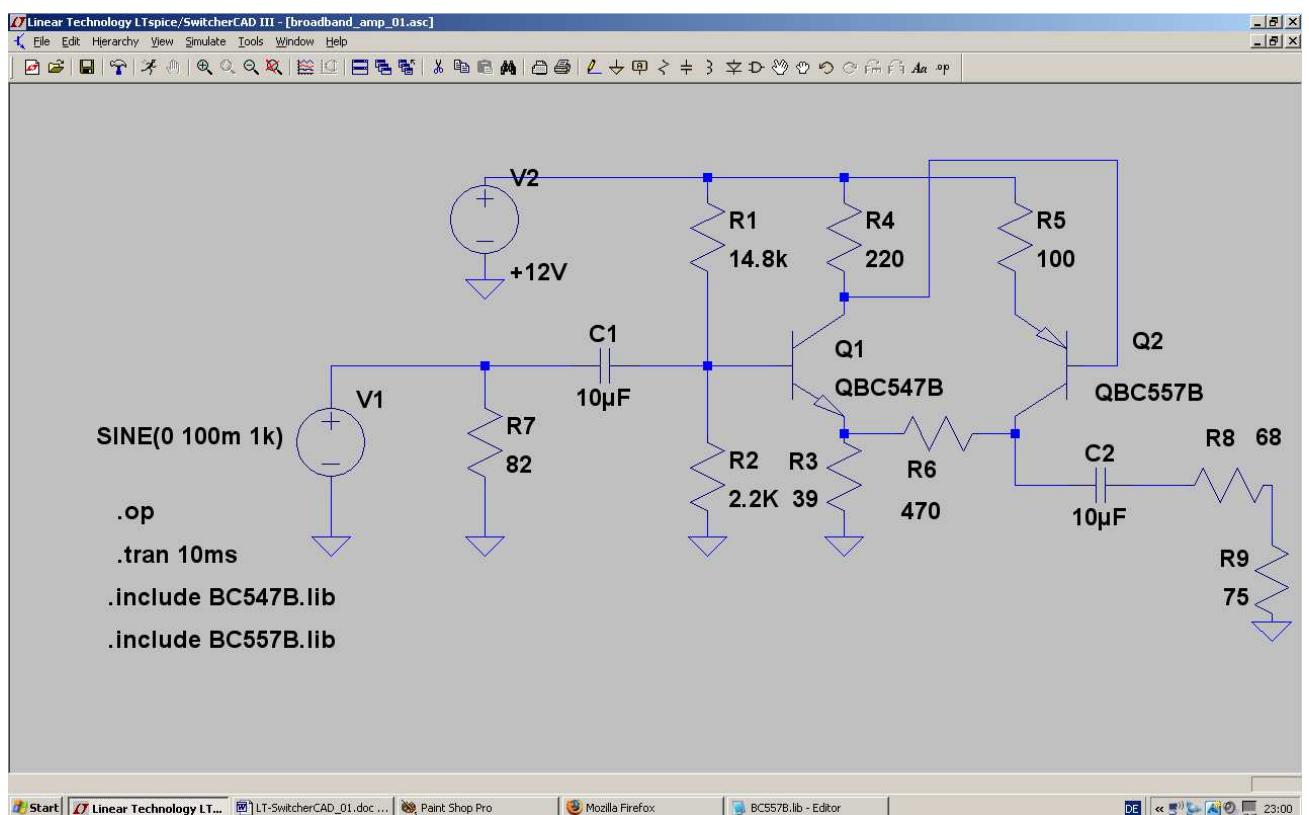
9.2. Zweistufiger gegengekoppelter Breitbandverstärker

9.2.1. 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.

9.2.2. Simulations-Schaltung und Simulations-Vorgaben



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

Bei den Widerständen $R1 = 14,8\text{k}$ und $R2 = 2,2\text{k}$ dürfen wir nur folgende Eingabeform **OHNE KOMMA IN DER WERTANGABE** benützen, da sonst mit falschen Werten -- ohne Warnung! -- simuliert wird!

14.8k und **2.2k**

Für den **npn-Transistor BC557B** ist natürlich wieder eine Google-Suche im Internet mit anschließender Speicherprozedur erforderlich ist (= HTML-File markieren, Inhalt in das Clipboard kopieren, Texteditor öffnen und Clipboard hineinkopieren, schließlich das File mit korrekter Endung im korrekten Ordner ablegen.....)

Bei beiden Transistoren ist natürlich jeweils die korrekte Namensänderung samt zugehöriger Spice-Direktive im Schaltplan erforderlich.

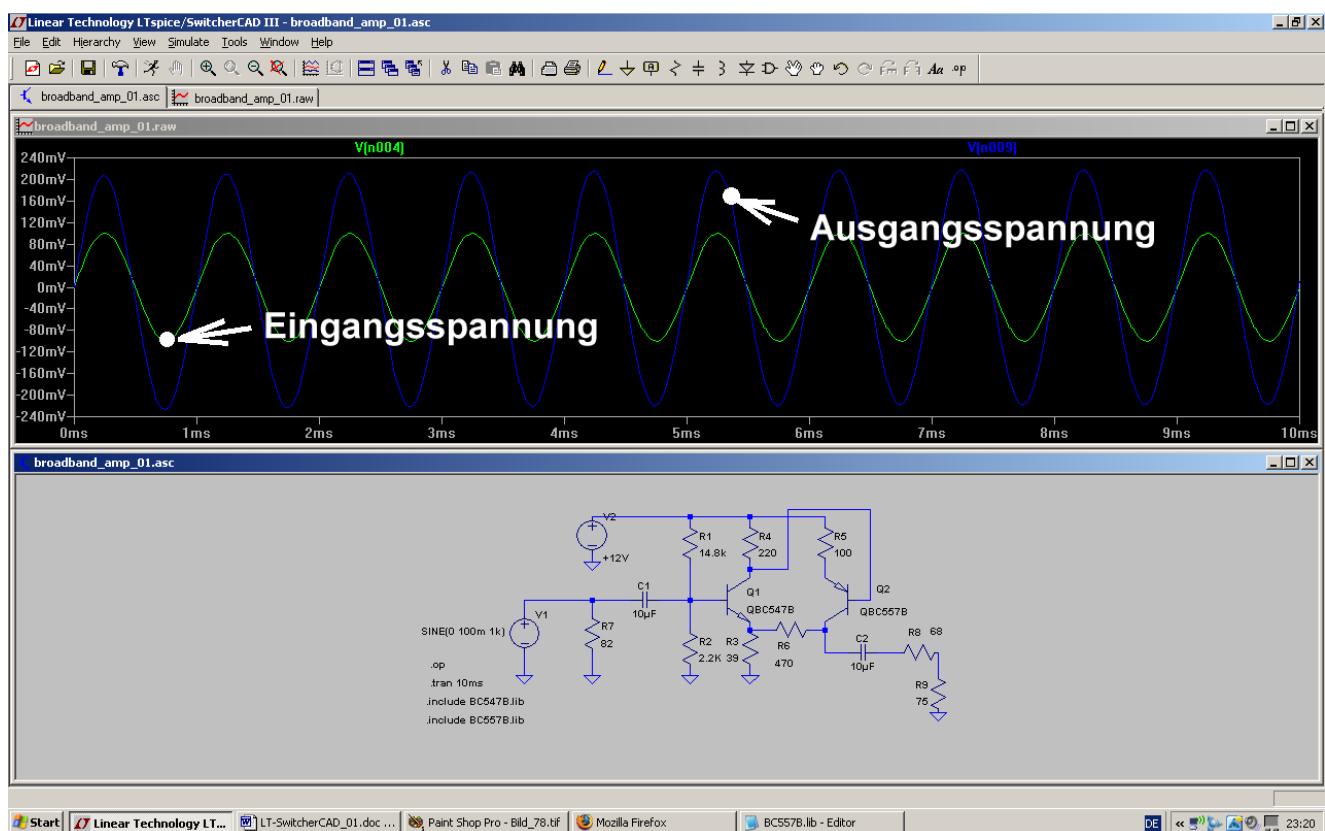
Angesteuert wird die Schaltung im ersten Durchgang mit einer 1kHz-Sinusspannung, die einen Spitzenwert von 100mV aufweist. Simuliert wird von 0...10Millisekunden.

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

Sie sorgt dafür, dass uns 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)

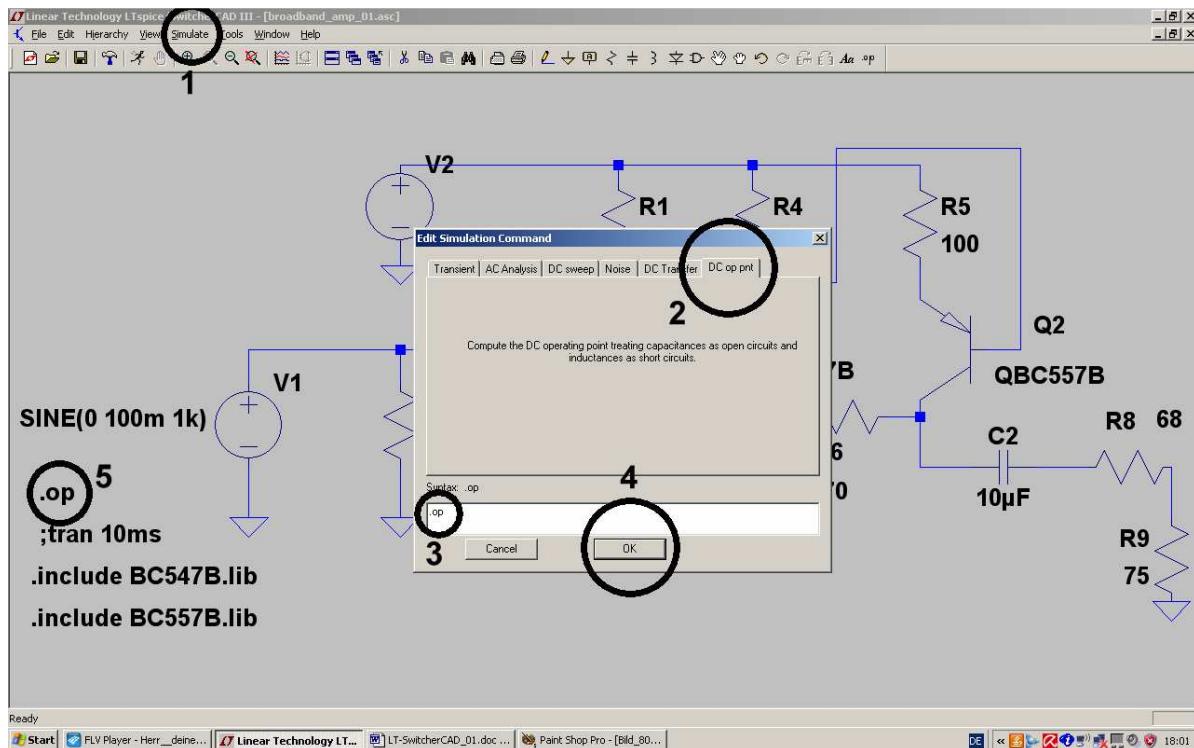
9.2.3. Simulation in der Time Domain (= im Zeitbereich)



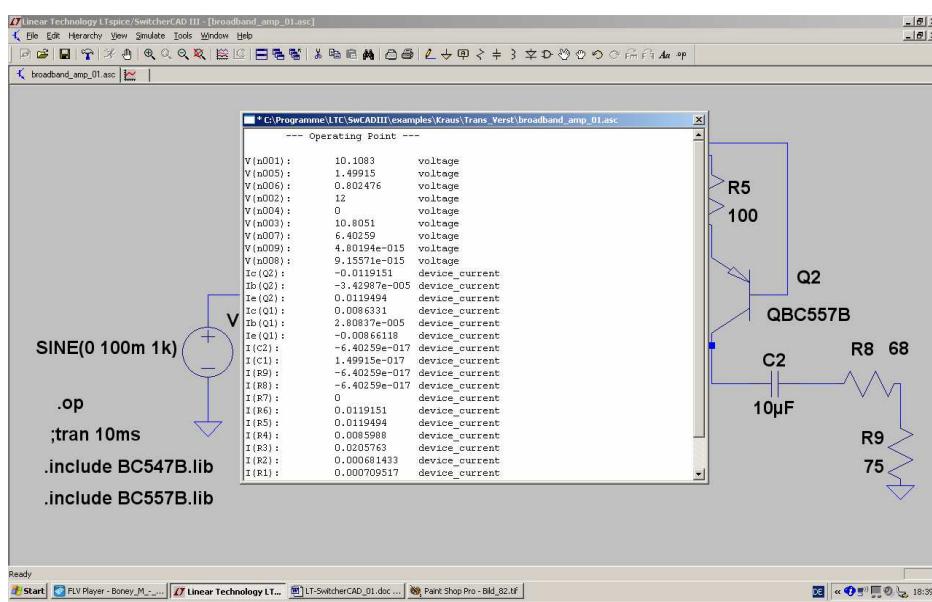
Man sieht, dass tatsächlich die Ausgangsspannung etwa doppelt so groß ist wie die Eingangsspannung (was einer Spannungsverstärkung von 6dB entspricht). Und zwei Emitterstufen bringen doppelte Phasenumkehr -- also muss das Ausgangssignal wieder gleichphasig mit dem Eingangssignal sein.....

9.2.4. DC-Bias (= Gleichstrom-Analyse)

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



- 1): Das Menü „Simulation“ wird geöffnet und darin „Edit Simulation Command“ angeklickt.
- 2): Dann wechseln wir auf die letzte Karteikarte „DC op point“
- 3). Im Eingabefenster muss nun „„.op““ stehen.
- 4): Das bestätigen wir mit OK und prüfen, ob
- 5) das Kommando „.op“ korrekt am Cursor hängt und im Simulationsschaltbild abgesetzt werden kann. (**Wichtige Kontrolle: vor dem Kommando „trans 10ms“ muss nun ein Strickpunkt stehen**).



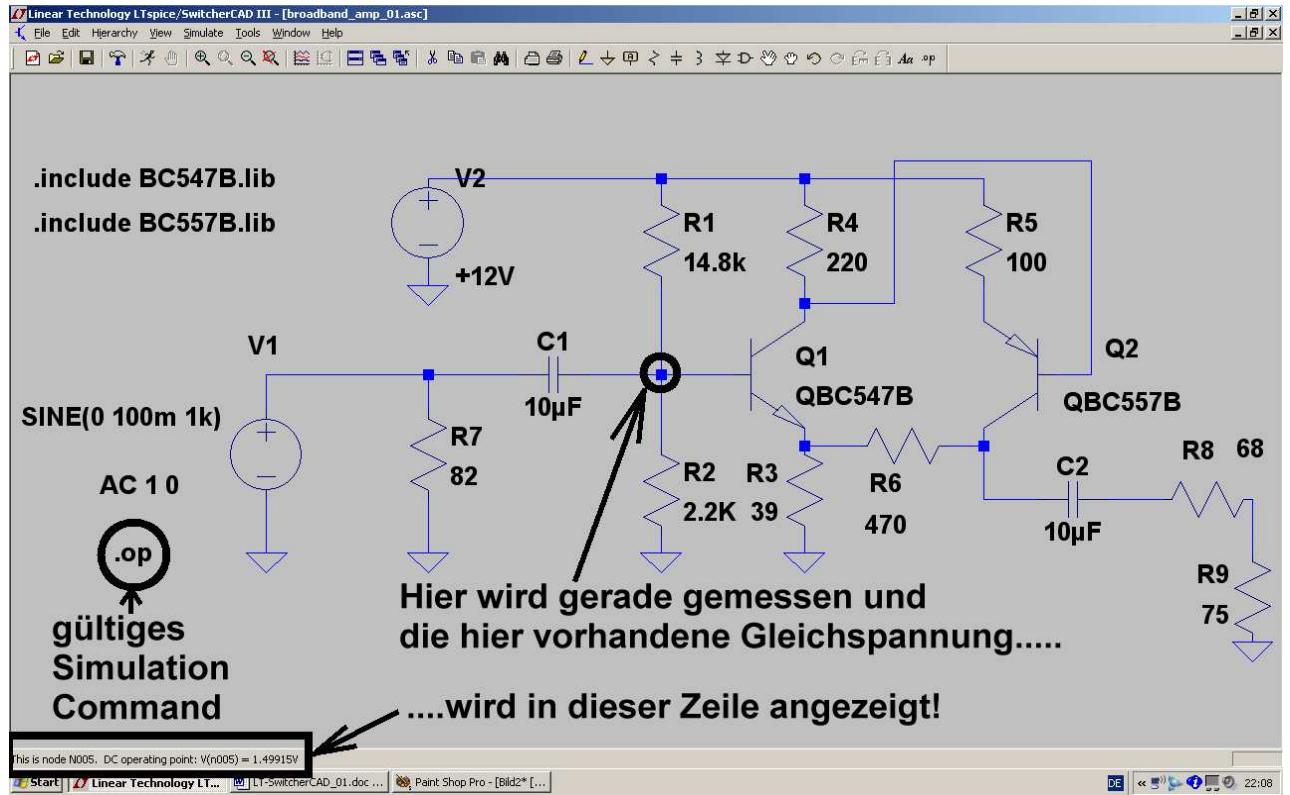
Jetzt dürfen wir wieder mal auf das „rennende Männchen“ (= Simulationsbutton) drücken und auf das Ergebnis warten.

So sieht es aus.

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

Da gibt es nämlich noch eine weitere hübsche Möglichkeit:

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



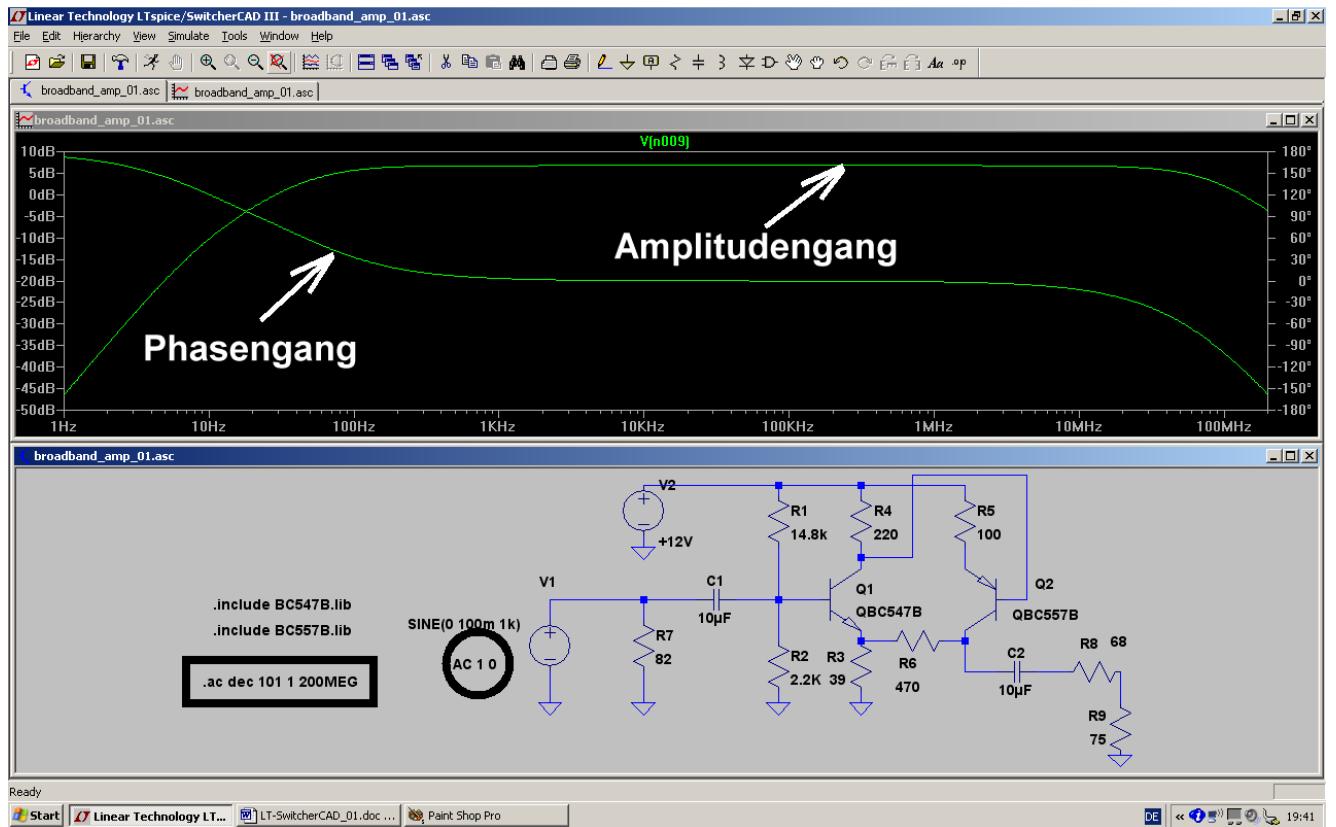
Wichtig:

Möchte man nun wieder zur Transientensimulation (= Ansteuerung mit einer Sinusspannung) zurück, dann muss vor „op“ ein Punkt und umgekehrt vor „trans....“ ein Strichpunkt sitzen! Am schnellsten geht das wieder über „Simulation / Edit Simulation Command / Karteikarte „Transient“ und dort einem Klick auf OK.

9.2.5. 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 101 Punkten pro Dekade erstellen.



9.3. 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.

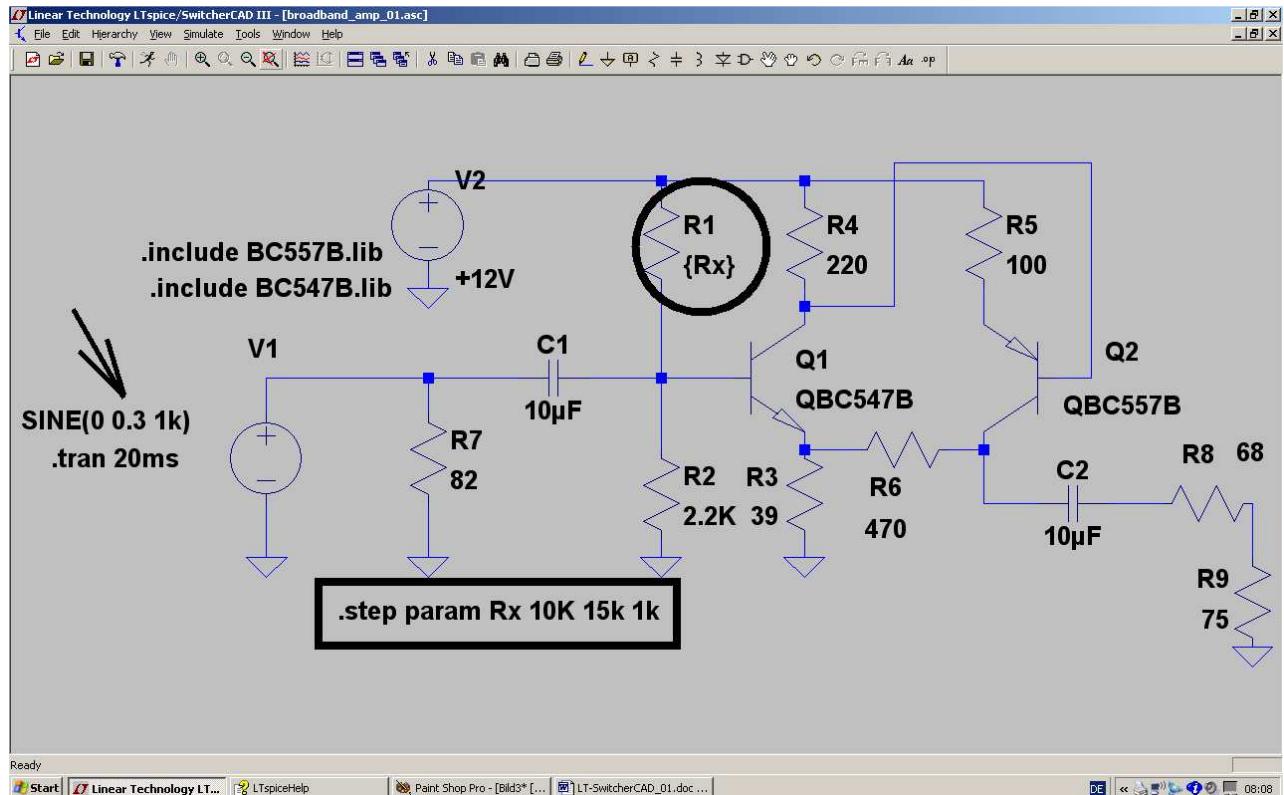
Wir wollen uns daraus das Kommando

.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 7.2.3.** her und lassen uns den oberen Basis-Spannungsteilerwiderstand R1 in 5 Schritten von 10k bis 15k variieren. Die Ergebnisse können dann direkt verglichen werden und so hat man sehr schnell die korrekte Potentiometer-Einstellung.

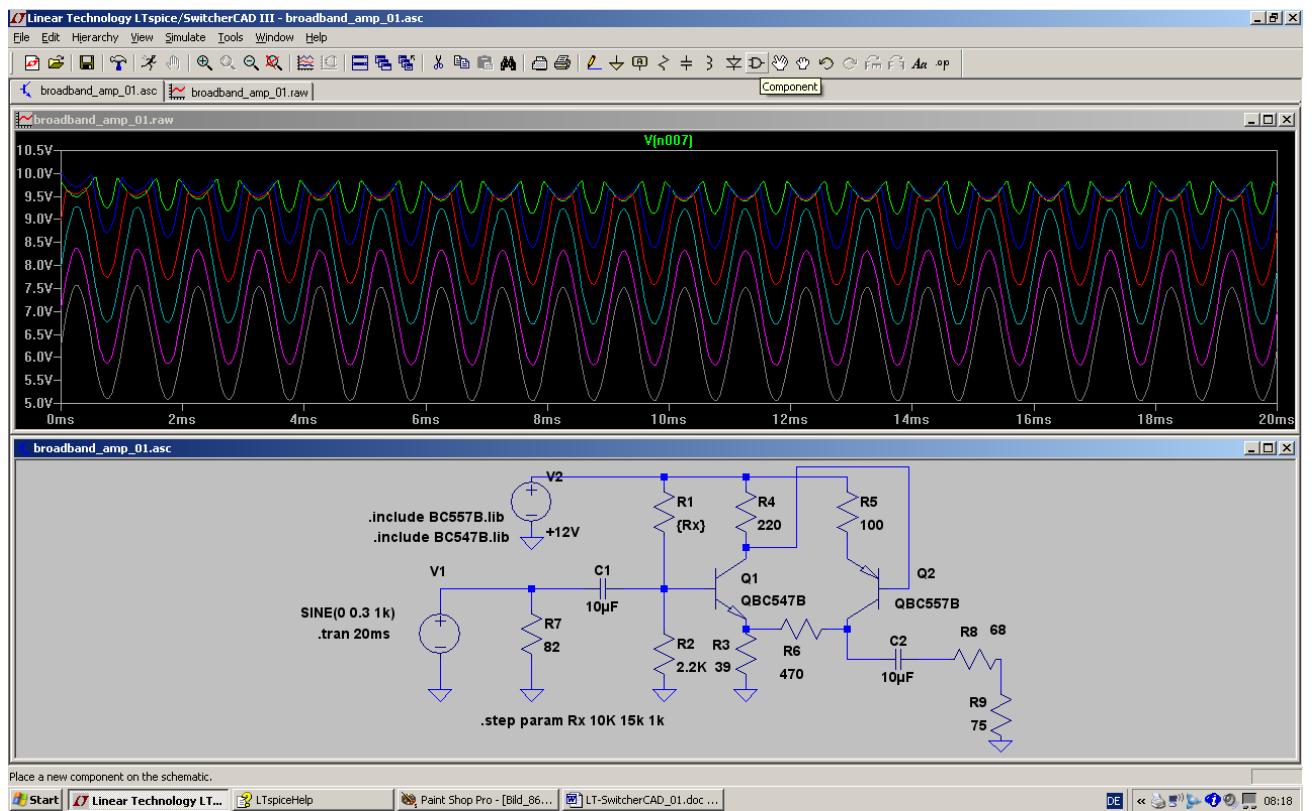


Bitte genau hinschauen:

- Die Amplitude der Eingangsspannung wurde auf 0,3V erhöht
- Beim Widerstand R1 wurde die Wertangabe durch den Parameter **{Rx}** ersetzt
- Schließlich ist noch die Spice-Direktive **.step param Rx 10k 15k 1k** erforderlich.
Dadurch wird R1 in Schritten von 1 Kilo-Ohm von 10k bis 15k 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 an den Kollektor von Transistor Q2 tippen, 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 (= unterste Kurve) erforderlich ist, um eine mittlere Kollektorgleichspannung von ca. 6,5V zu erhalten.

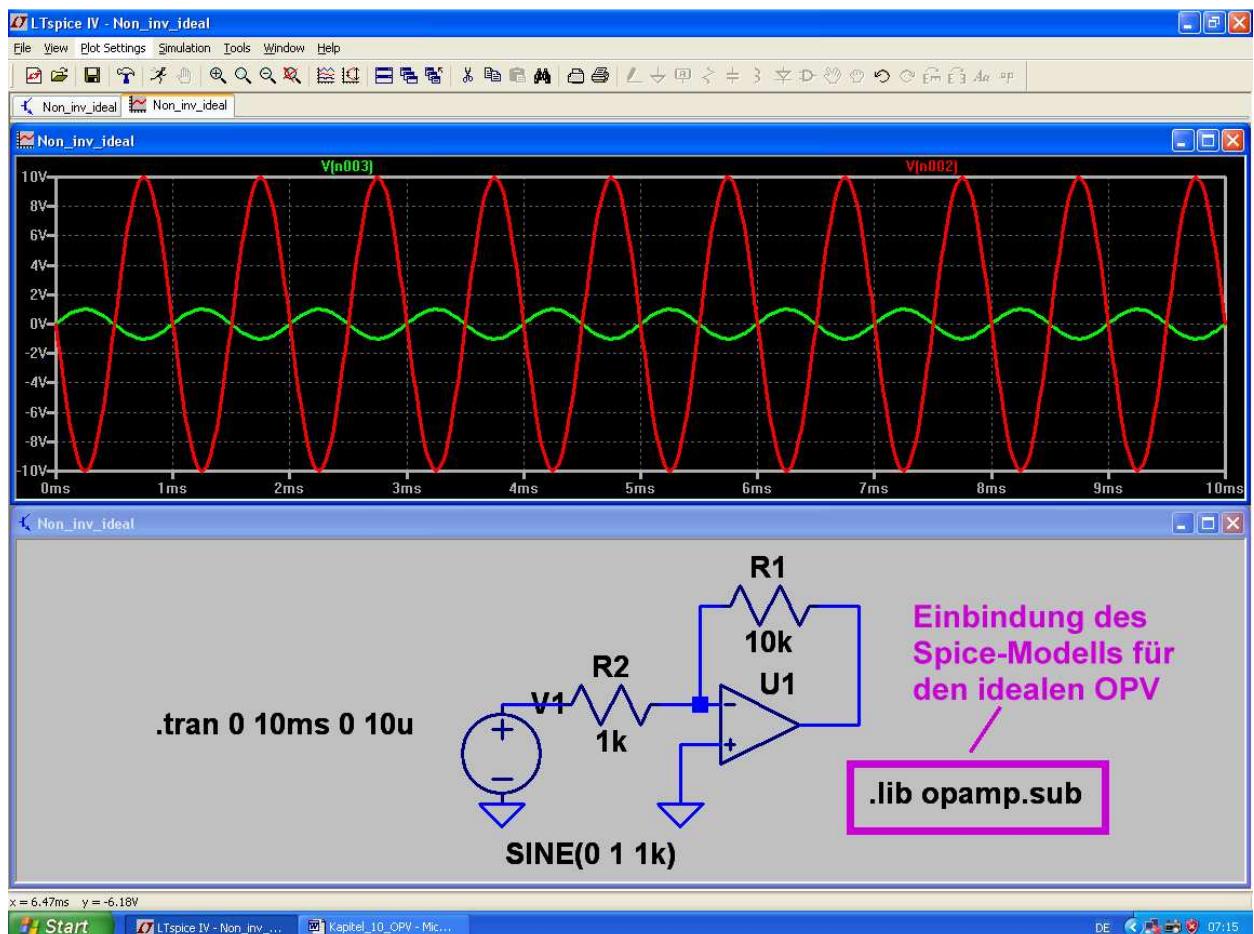


10. Sechstes Projekt: OPV-Schaltungen

10.1. Einstieg: Umkehrender Verstärker

10.1.1. Verwendung des einfachsten OPV-Modells „opamp“

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“ und starten damit ein neues Projekt, nämlich einen umkehrenden Verstärker mit 10facher Verstärkung. Gespeist wird der Eingang mit einer Sinusspannung (Spitzenwert = 11V / Frequenz = 1kHz). Allerdings darf man die **Spice-Direktive zur Einbindung der Modelldatei** nicht vergessen:

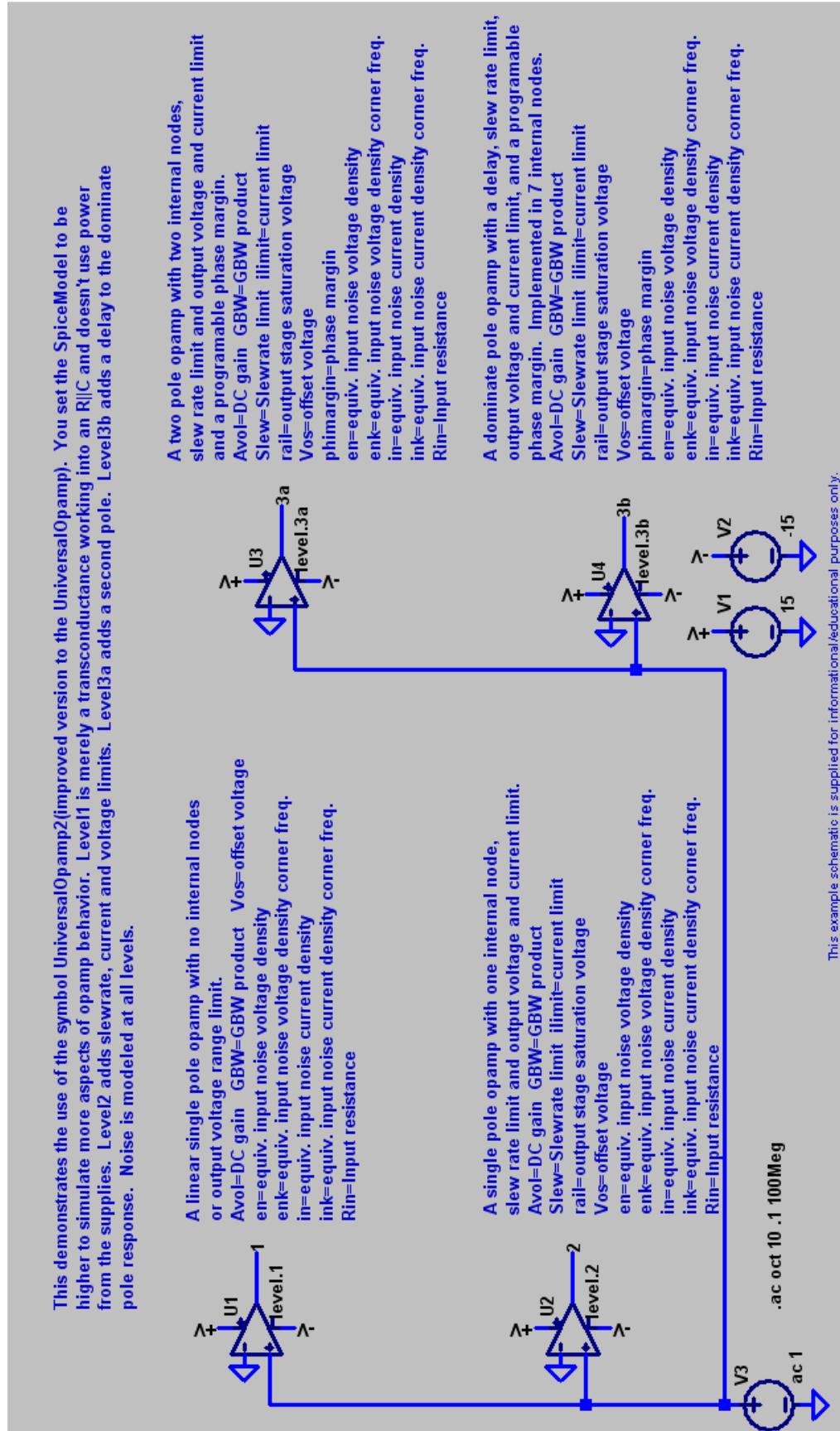


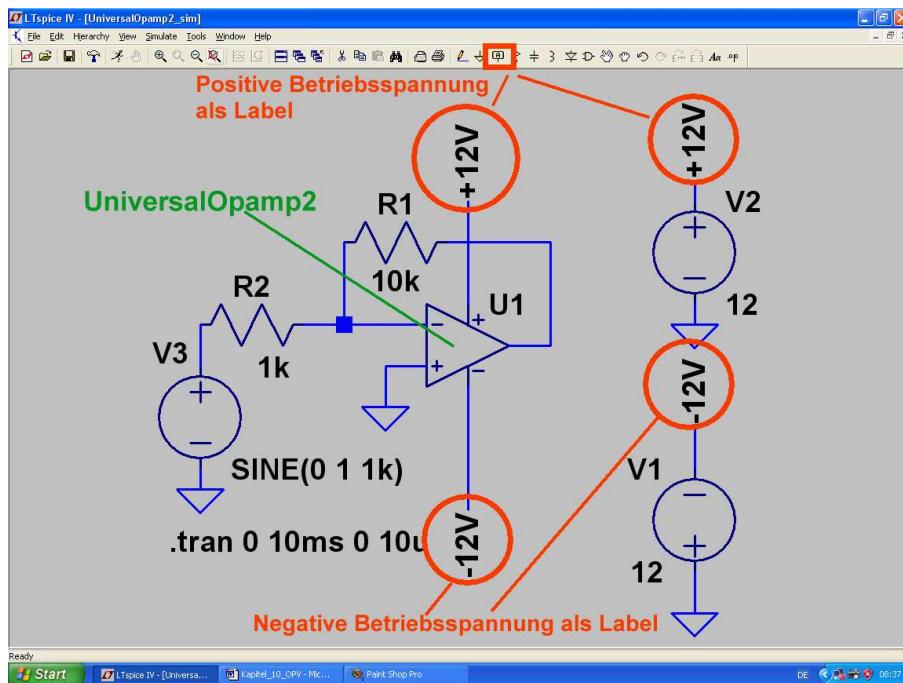
Hinweise:

- Die Ergebniskurven im oberen Diagramm wurden mit **dicken Linien** dargestellt. Dazu klickt man auf den „**Button mit dem Hammer**“ in der Menüleiste, wählt die Karteikarte „**Waveforms**“ und aktiviert darin die erste Zeile „**Plot data with thick lines**“ mit einem Häkchen.
- Möchte man die **Farbe** einer Kurve verändern, dann klickt man mit der **linken Maustaste** auf die Bezeichnung dieser Kurve (oberhalb des Diagramms). Im auftauchenden Menü wird eine Palette mit 16 Farben zur Auswahl angeboten.
- Möchte man das **Schaltbild mit etwas kräftigeren Linien** zeichnen, dann klickt man wieder auf den „**Button mit dem Hammer**“. Auf der Karteikarte „**Drafting Options**“ muss man dann die dritte Zeile von unten („**Draft with thick lines**“) aktivieren.

10.1.2. Einsatz des Universalen OPVs und die Verwendung von Labels

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 vier verschiedene Qualitätsstufen und die sind im mitgelieferten Beispiel „UniversalOpamp2.asc“ genau beschrieben:

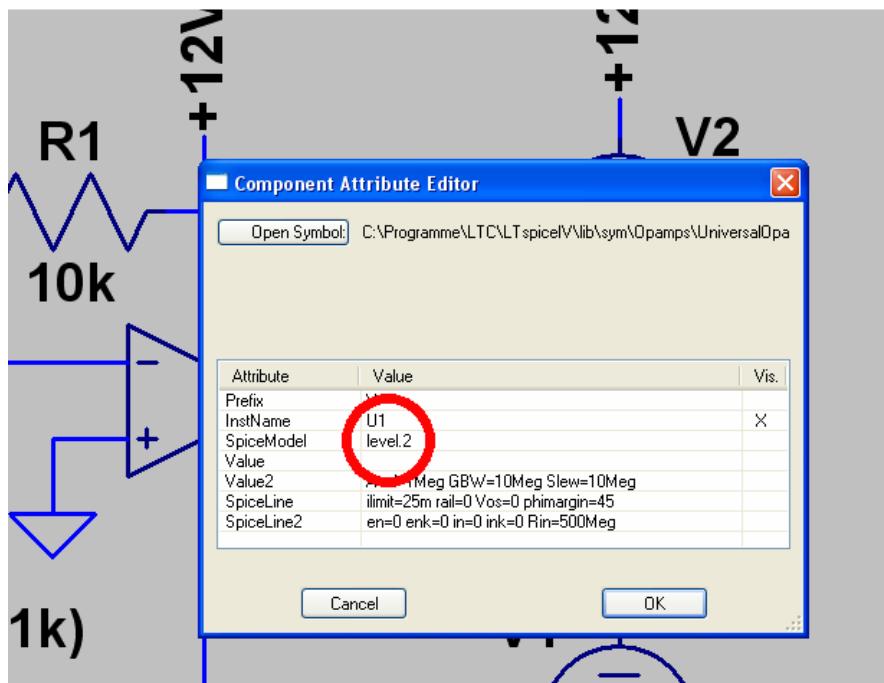




So sieht die Schaltung aus, wenn man in der vorigen Schaltung einen „UniversalOpamp2“ einsetzt. Dabei wurden die beiden Speisespannungsquellen mit „Labels“ versehen, um das Schaltbild übersichtlicher zu gestalten.

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 $f = 1\text{kHz}$.

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 Pulsfrequenz (z. B. $f = 100\text{kHz}$) ansteuert, zeigen sich die Unterschiede.

10.1.3. Verwendung eines OPVs vom Typ „TL072“ und Erstellung eines passenden Spice-Symbols

Beim TL072 handelt es sich um einen sogenannten „Wald- und Wiesentyp“: oft verwendet, leicht beschaffbar und nicht teuer. Da jedoch bei LTSpice nur die Modelle der hauseigenen Typen mitgeliefert werden, wollen wir uns mal Schritt für Schritt ansehen, wie man den TL072 für den Einsatz in diesem Programm vorbereitet.

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.

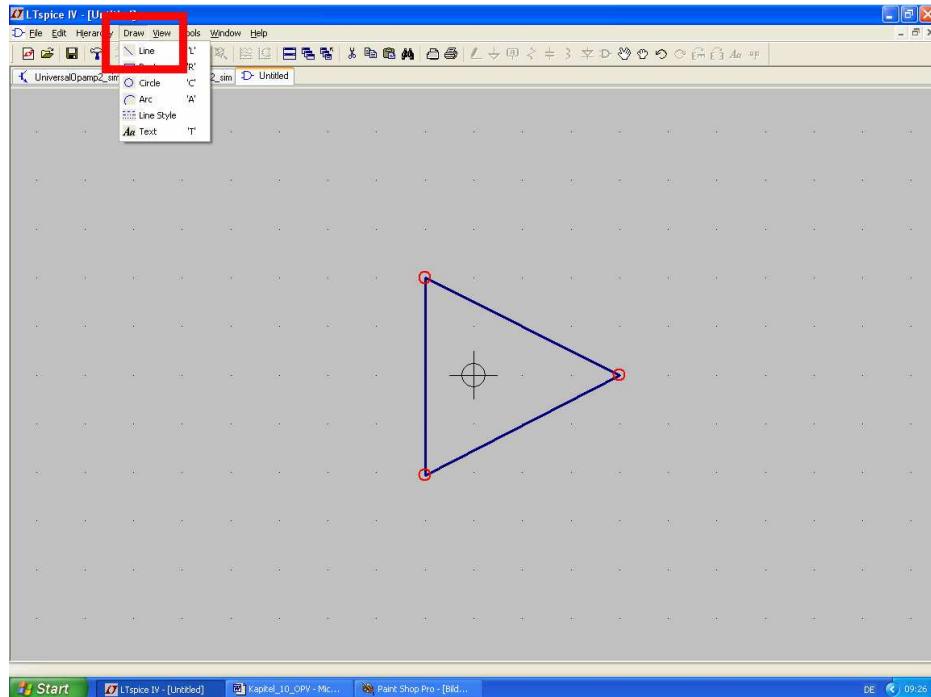
2. Schritt:

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) 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 „**LTS defense / 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 .
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
```

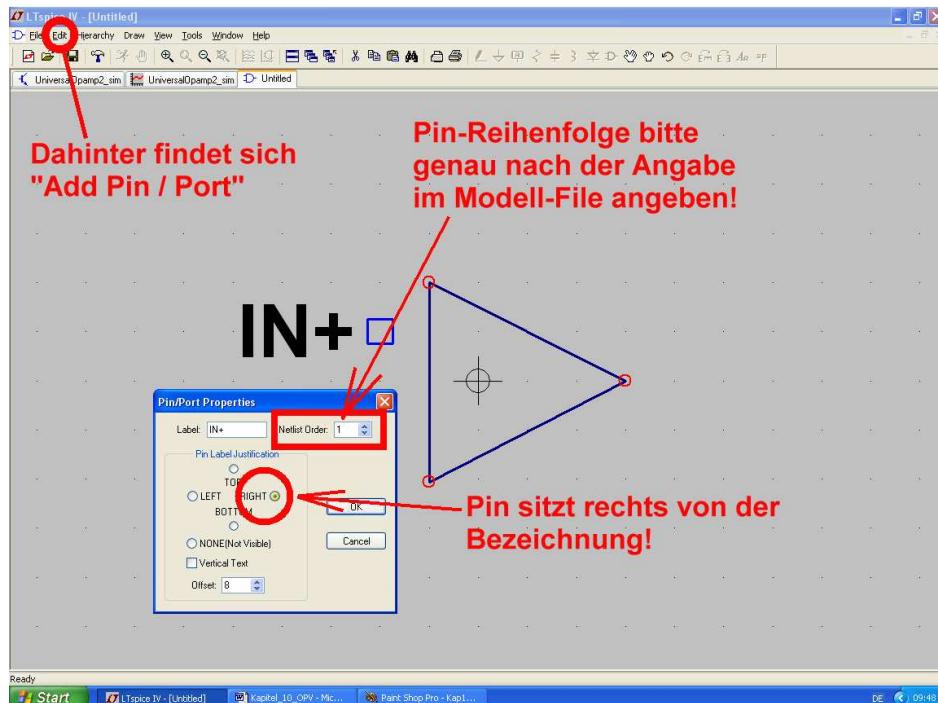
3. Schritt:

LTS spice 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:

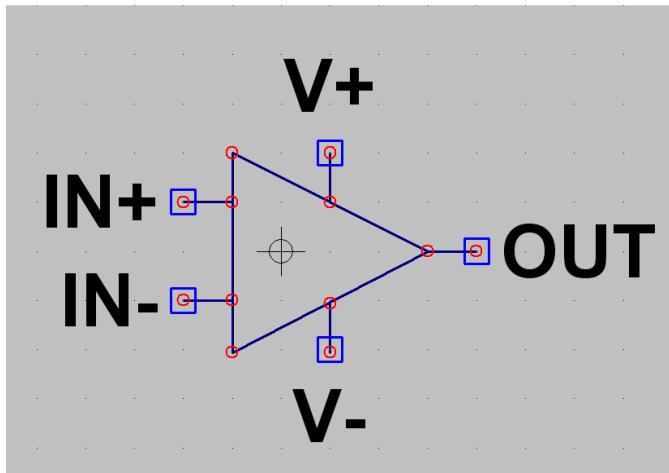
Erstens muss die richtige Pin-Nummer gemäß den Angaben im Modell-File („TL072.sub“) eingetragen werden.

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.

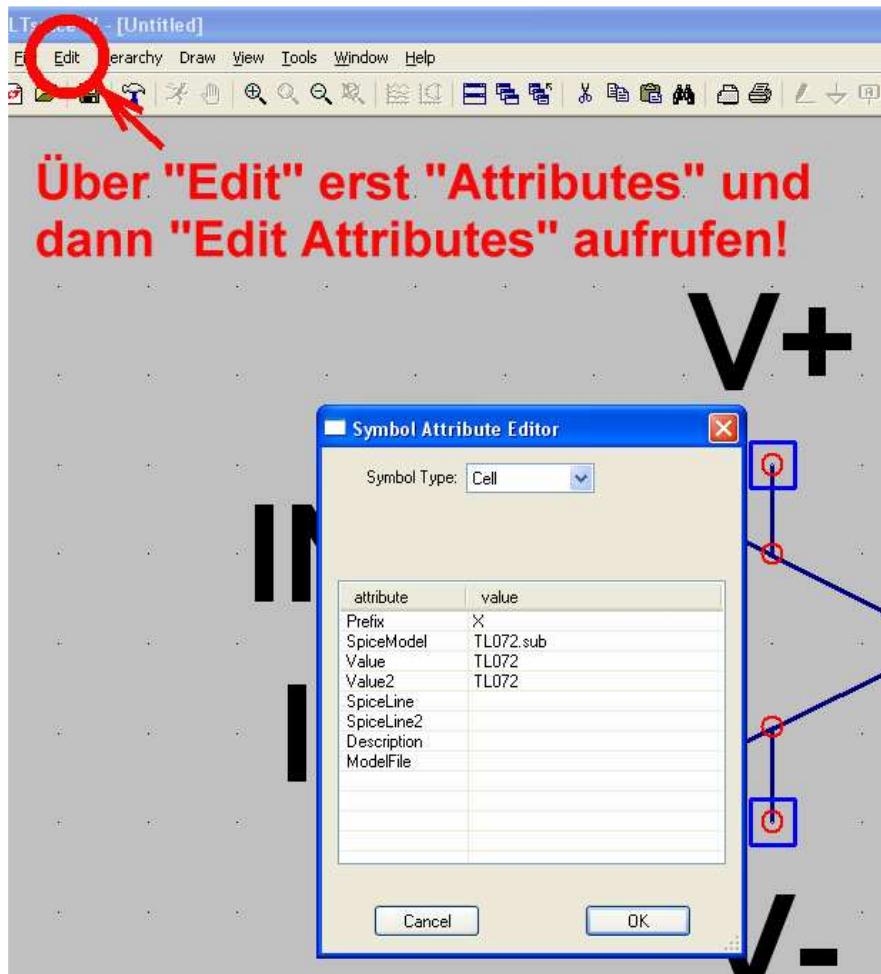
Zweitens muss man angeben, wo sich der Pin räumlich gegenüber der Bezeichnung befinden soll (hier: rechts).

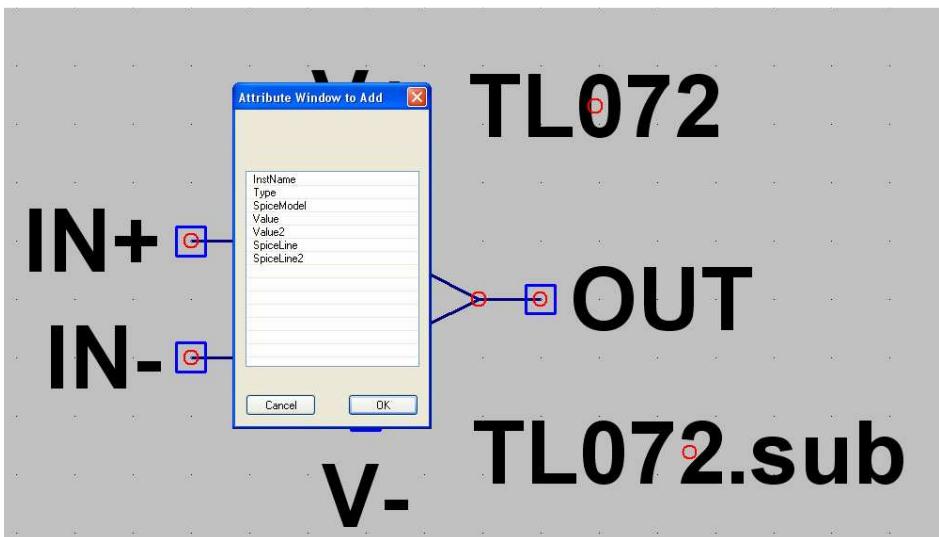


Sind alle Pins platziert, dann sorgt man im Anschluss wieder mit „Draw Lines“ für **Verbindungsleitungen** von jedem Pin zum Schaltzeichen.

Das ist das Ziel...

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

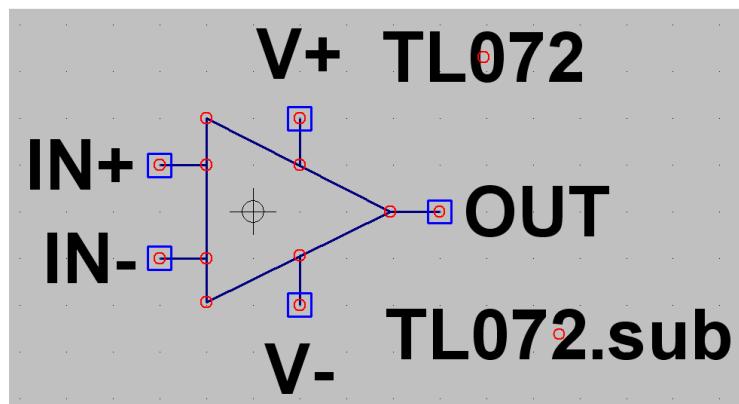




Nun wiederholt sich das Spiel: über „Edit“ und „Attributes“ kommen wir nun an „Attribute Window“ heran.

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.

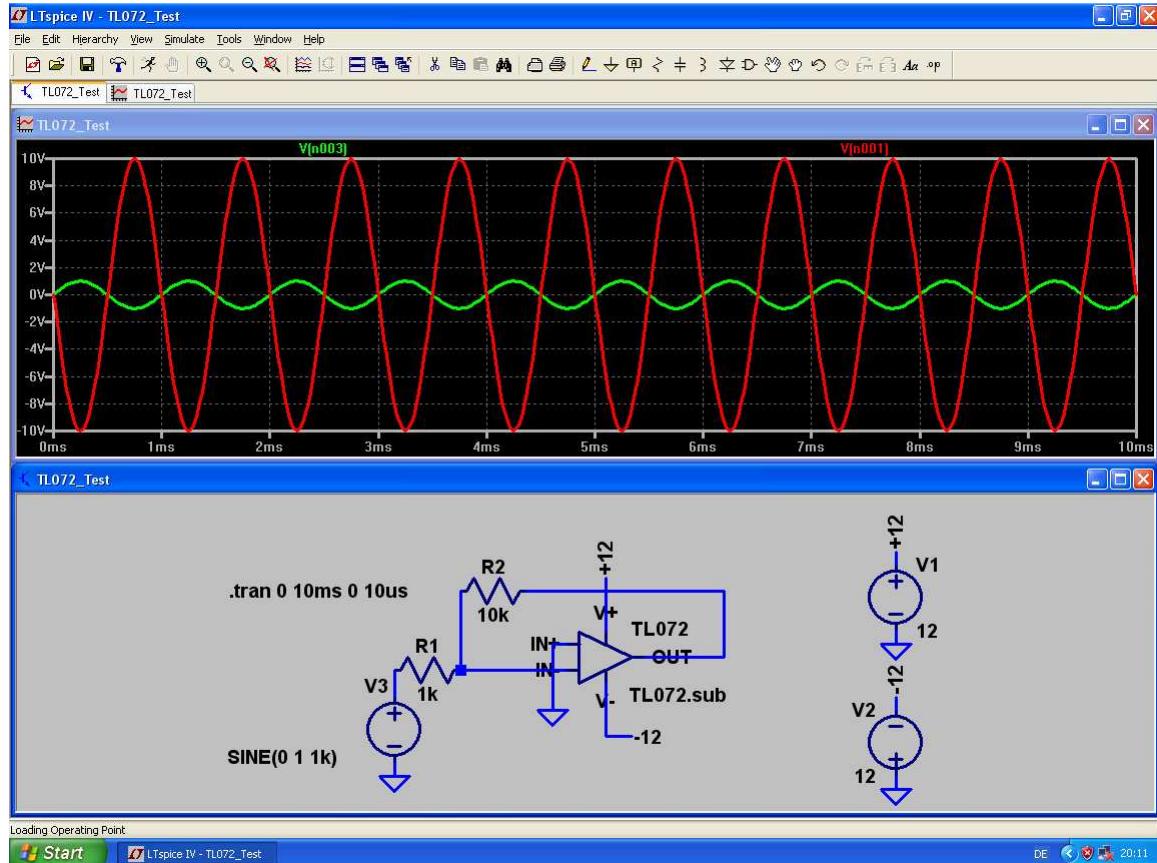
Achtung:
Es muss im Pfad

LTspice / lib / sym / Opamps

unter dem Namen „TL072“ abgelegt werden.

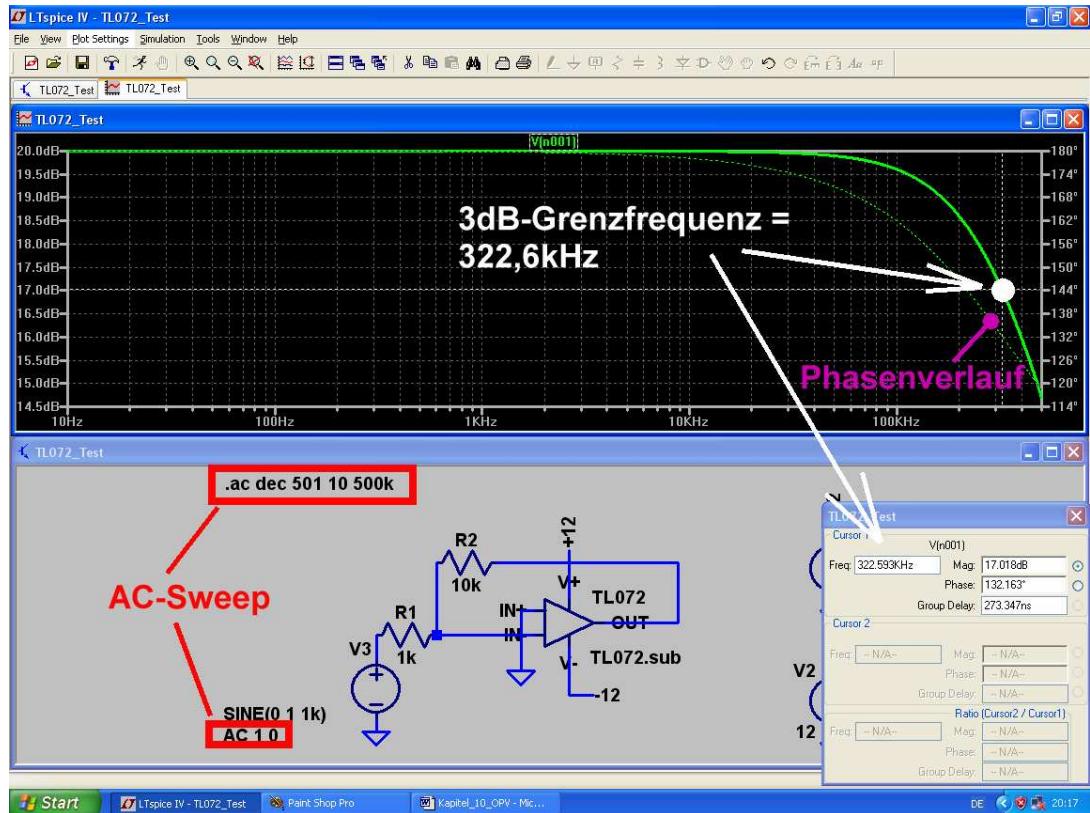
Fertig!

Nun ist natürlich unser invertierender Verstärker mit 10facher Verstärkung auch kein Problem mehr:



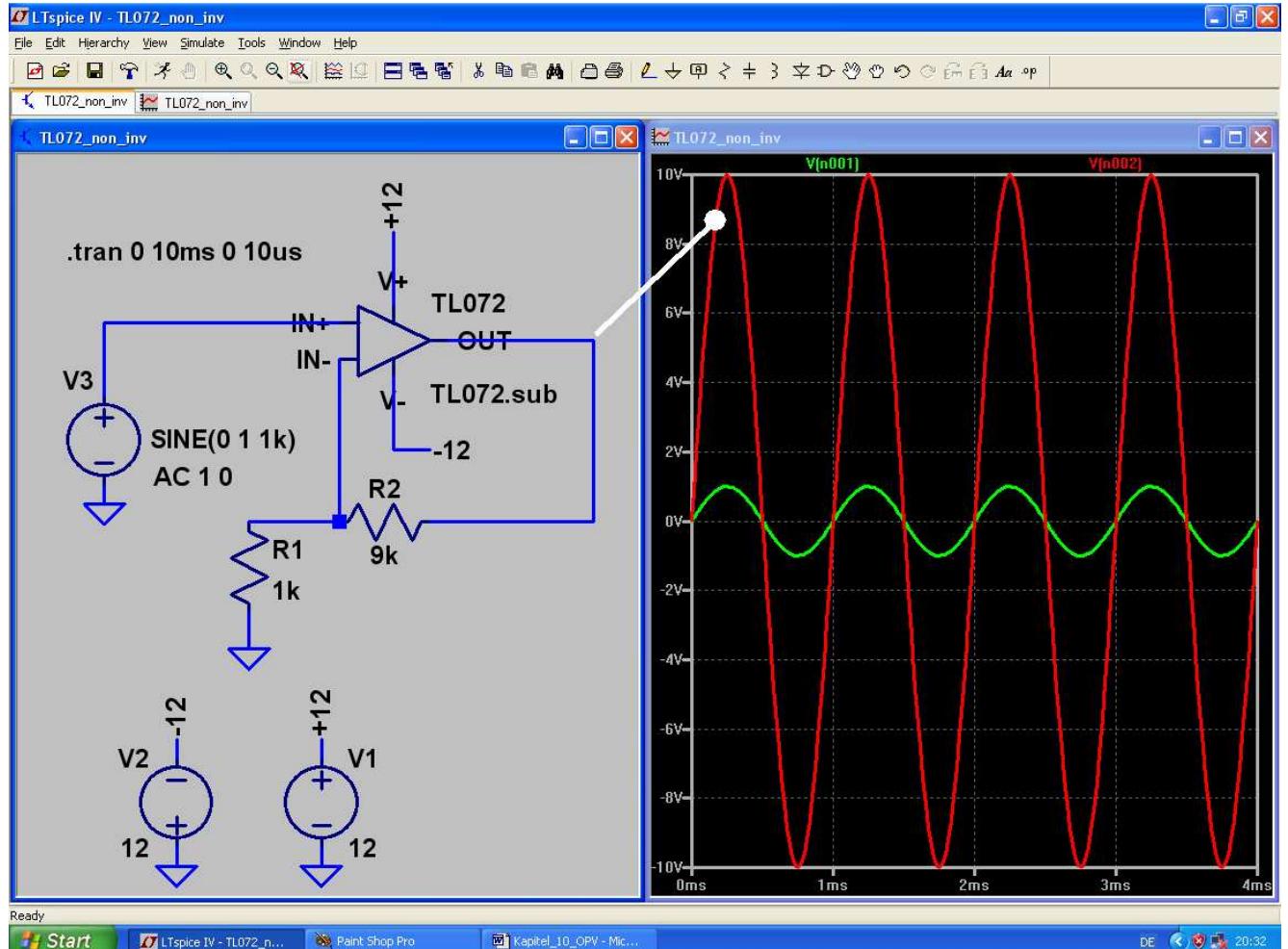
Interessant dürfte dagegen ein AC-Sweep sein, der das Verhalten bei höheren Frequenzen bzw. die obere Grenzfrequenz zeigt. Bitte mal simulieren!

Ergebnis:



10.2. Nicht umkehrender Verstärker mit TL072

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

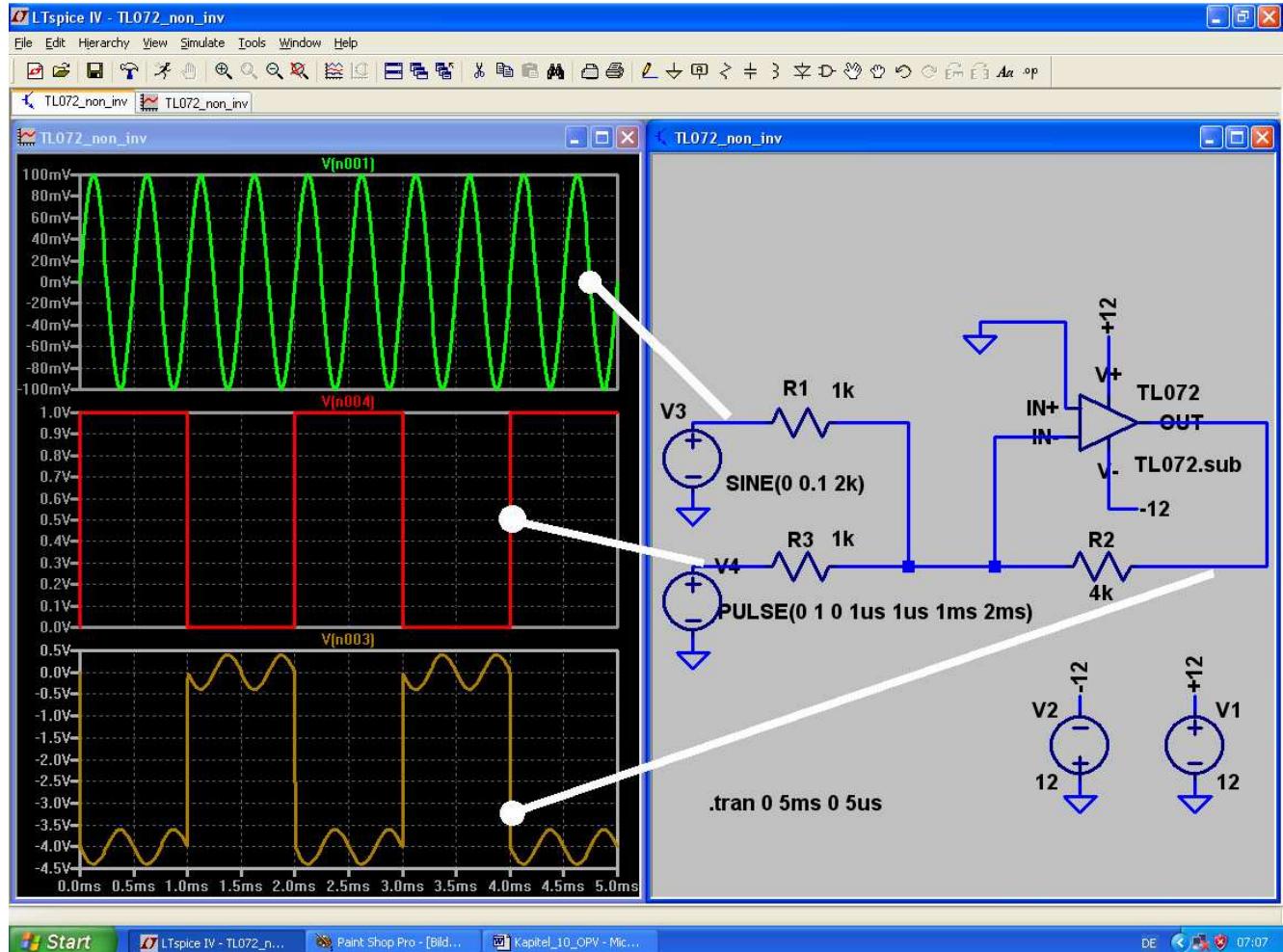


10.3. Analog-Addierer mit TL072

Mit OPVs lassen sich bekanntlich analoge Signale sehr leicht zusammenaddieren. Wir wollen das mit einem Sinus ($f = 2\text{kHz}$, Spitzenwert = 0,1V) und einem Rechtecksignal ($f = 500\text{Hz}$, $U_{\min} = 0\text{V}$, $U_{\max} = 1\text{V}$) ausprobieren und fordern zusätzlich:

- eine 4fache Verstärkung und
- einen Eingangswiderstand von $1\text{k}\Omega$ bei jedem Eingangskanal.

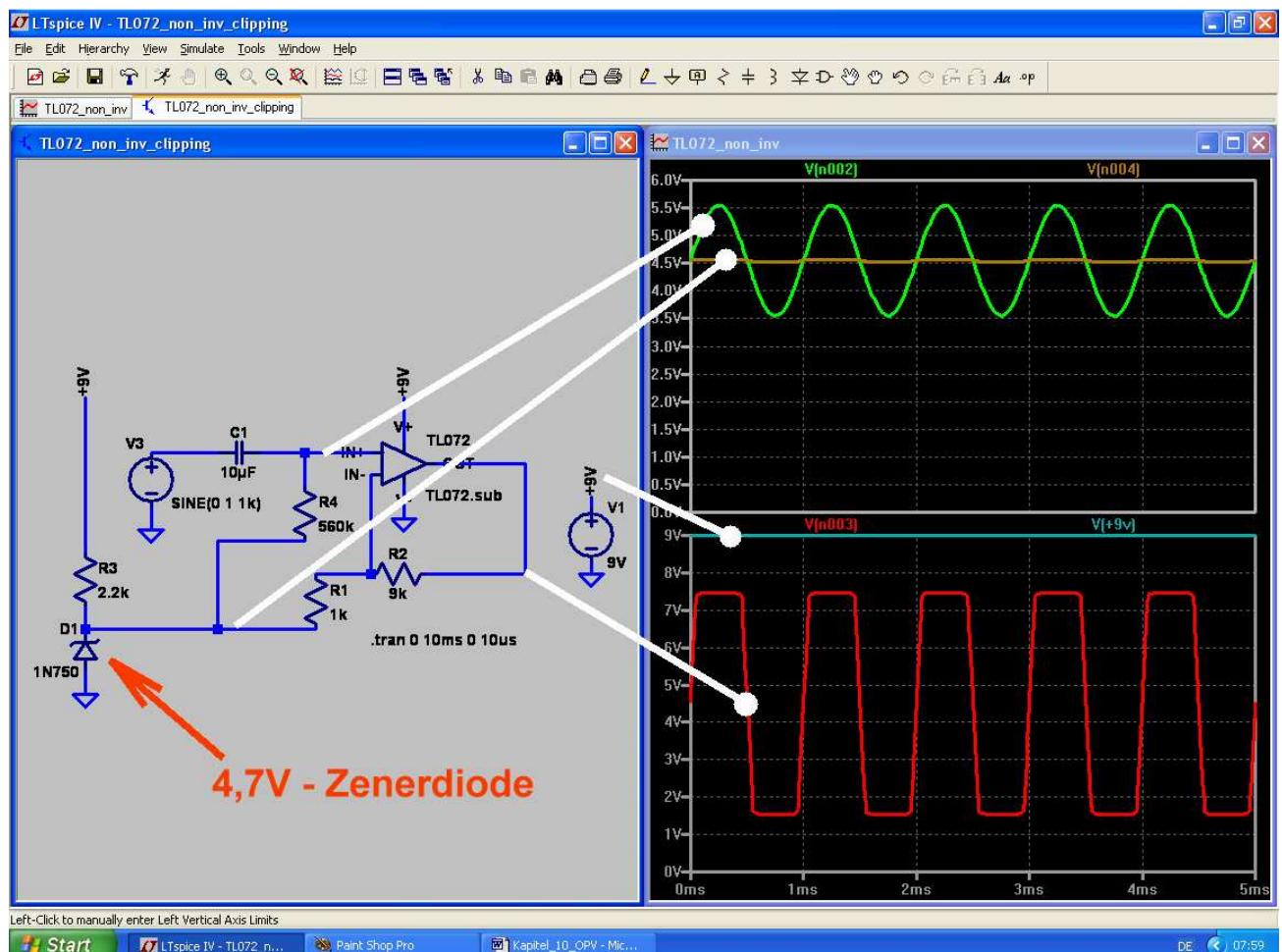
Dazu dient eine umkehrende Schaltung, bei der der Invertierende Eingang des OPVs einen „Virtuellen Nullpunkt“ bildet.



10.4. Probleme beim Betrieb mit nur einer Betriebsspannung

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

- Der „V+“ - Anschluss des OPV wird an diese **Versorgungsspannung** angeschlossen.
- Der „V-“ - Anschluss wird mit **Masse** verbunden.
- Das Gleichspannungspotential an den Eingängen des OPVs** muss dann auf **halbe Betriebsspannung** gebracht werden.
- Beim 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 unverzerrte Ausgangsspannung durch eine Simulation ermitteln.



Analyse der Schaltung und Auswertung der Ergebnisse:

Mit der Zenerdiode 1N750 wird eine „Vorspannung“ mit $U_b / 2 = +4,5V$ aus der Vorsorgungsspannung von +9V erzeugt. Sie wird über den Widerstand $R_4 = 560k$ 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 R_1 und R_2 .

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.

11. Siebtes Projekt: DC-DC-Konverter

Dieses Projekt beginnt gleich mit einem Hinweis und einer **Warnung**:

Solche Schaltungen nutzen zum Teil recht verwickelte Vorgänge in Induktivitäten aus. **Das übliche einfache Modell einer Induktivität wird deshalb in der Simulationspraxis dieser Sache nicht gerecht und das Ergebnis ist entweder Nonsense 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 geheimgehalten, 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 beladen, dass diese Effekte stark gedämpft werden und / oder genügend lang simulieren.

11.1. Bereitstellung des Power-MOSFETs „IRFZ44N“

Für die folgende Untersuchung der verschiedenen Grundschaltungen benötigen wir einen passenden elektronischen Schalter. Hier haben sich Power-MOSFETs durchgesetzt und die Firma IRF (= International Rectifiers) ist ein bekannter und führender Hersteller. Selbstverständlich stellt sie für alle Produkte SPICE-Modelle bereit, aber die Produktpalette ist ungeheuer groß.

Wir benutzen für unsere Simulationen einen „Standardtyp“ mit dem Namen IRFZ44N und benötigen dafür ein neues Symbol in unserer Software.

Die folgende Selbstbau-Aktion ist unvermeidbar, weil das Modell nur als „Subcircuit“ erhältlich ist und wir deshalb selbst antreten müssen.....

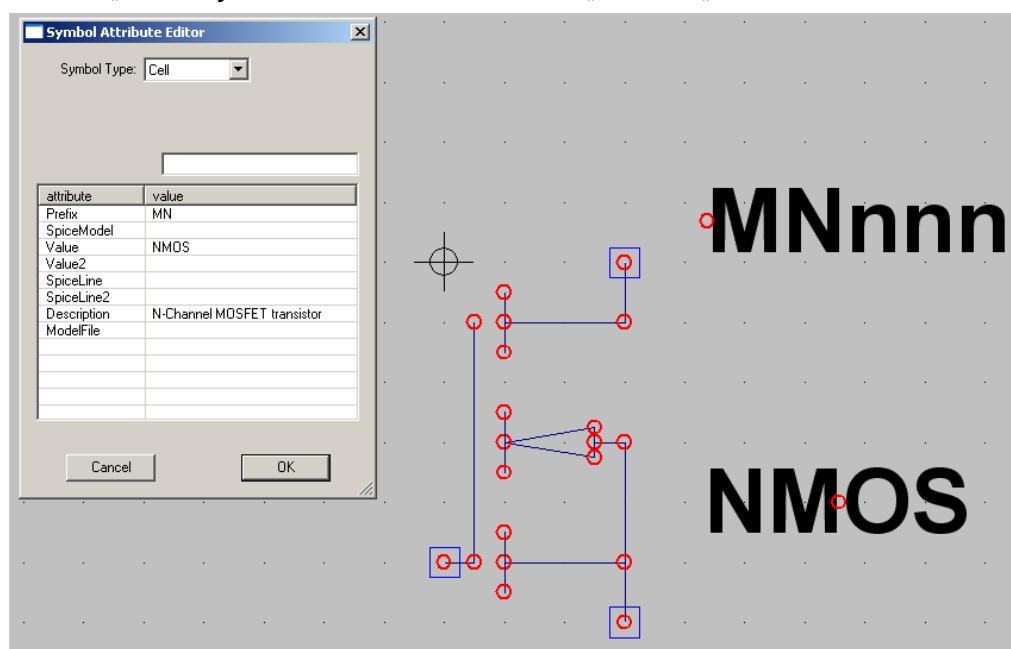
Also wieder mal:

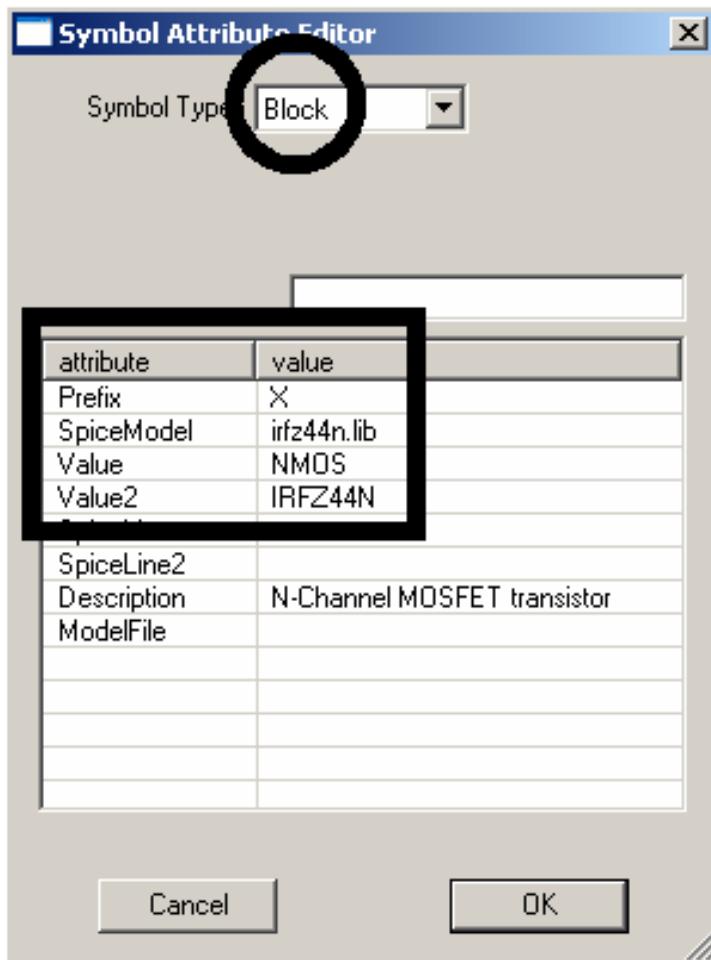
1. Schritt:

Mit der Suchmaschine (Google o. ä.) holen wir unter dem Begriff „**irfz44n spice model**“ direkt die Datei „**irfz44n.spi**“. Deren Inhalt wird markiert, in ein neues Blatt eines Editors kopiert und als „**irfz44n.lib**“ im Ordner „**lib / sub**“ des SwitcherCAD-Programms gespeichert.

2. Schritt:

Unter „**File**“ wählen wir „**New Symbol**“ und gehen dann nochmals in „**File**“. Jetzt öffnen wir im Ordner „**lib / sym**“ die Datei „**nmos.asy**“. Anschließend machen wir über „**Edit**“ und „**Edit Attributes**“ die Attribute-Liste sichtbar:



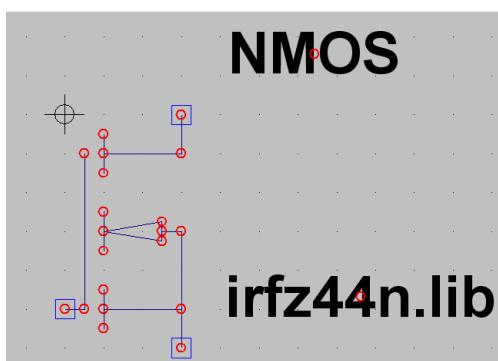


3. Schritt:

Und darin muss massiv umgebaut werden.

Bitte alle Einträge nach nebenstehendem Vorbild sorgfältig vornehmen und prüfen!!

(Besonders in der ersten Zeile „Prefix“ muss unbedingt von MN = model auf „X“, also **subcircuit**, umgestellt werden!)



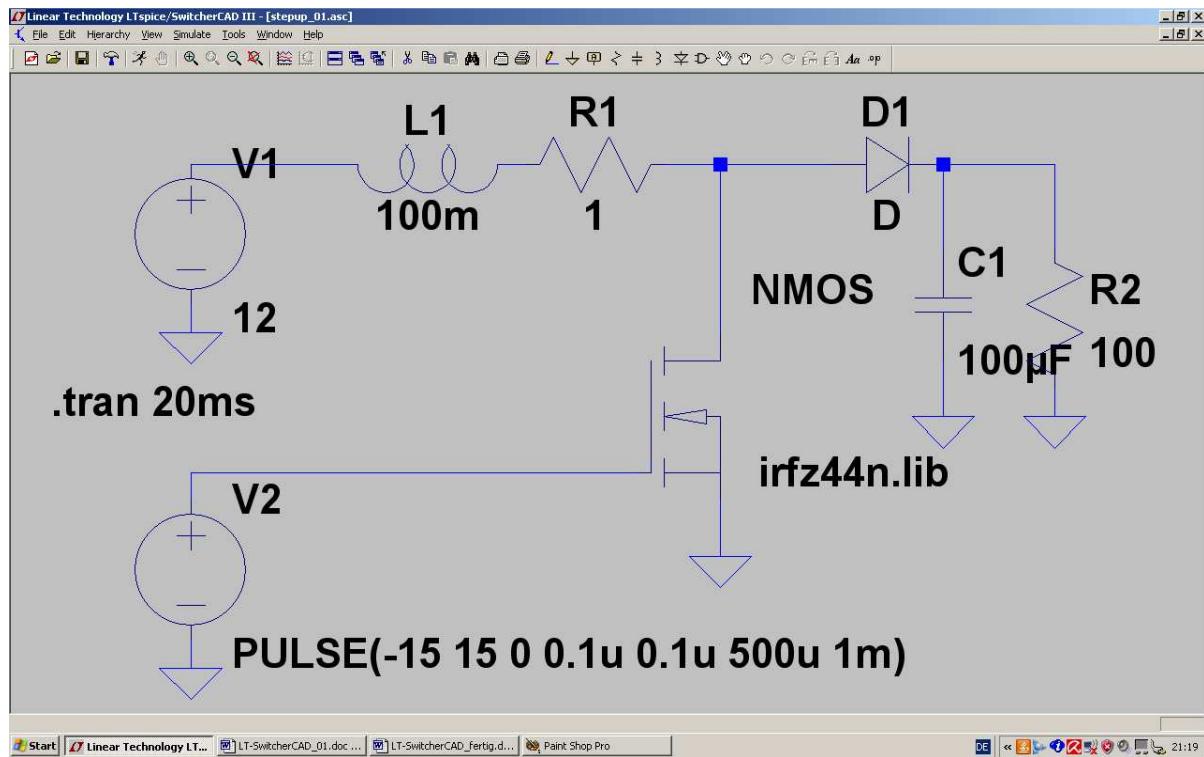
4. Schritt:

Nun gehen wir nochmals in „Edit“, öffnen diesmal aber „Attribute Window“. Wir klicken „Value“ an und platzieren die auftauchende Bezeichnung „NMOS“ oberhalb unseres Symbols. Nach der Bestätigung mit OK wiederholen wir die Prozedur, aber diesmal klicken wir auf „SPICE File“. Die auftauchende Angabe „irfz44n.lib“ wird unter das Symbol gesetzt.

Erst jetzt dürfen wir das fertige Symbol speichern -- am besten in einem extra angelegten eigenen Ordner „NMOS-FETs“ in „lib / sym“.

Das war's und jetzt geht es an die Schaltungen.

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



Die Schaltung besteht zunächst aus der Gleichspannungsquelle V_1 mit 12 V, an die unsere Induktivität mit $L = 100\text{mH}$ (Fachausdruck: „Speicherdrossel“) angeschlossen wird. Es folgt der geforderte Reihenwiderstand, 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

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

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 rückartig 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 höher auf. 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.

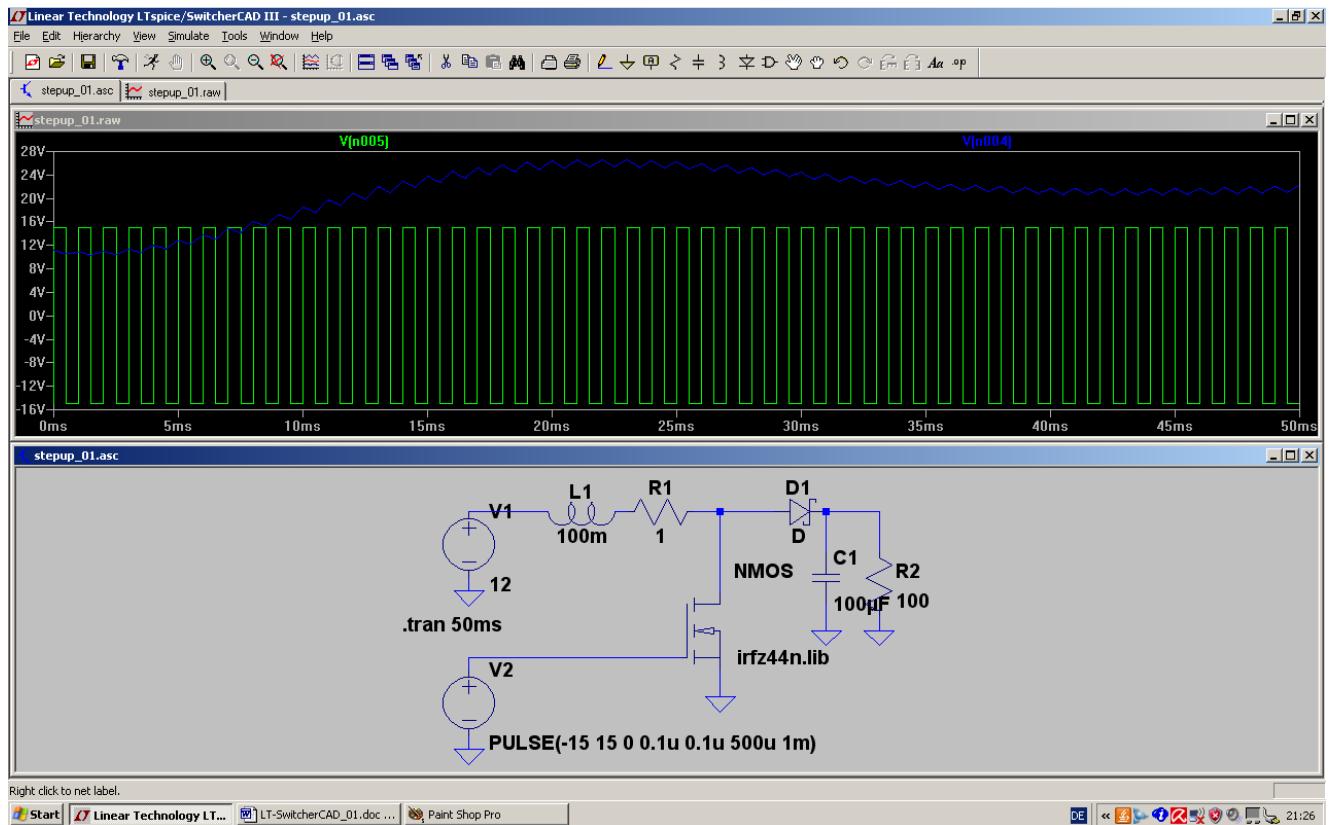
Folge: die Ausgangs-Gleichspannung wird deutlich höher als die Betriebsspannung!

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.

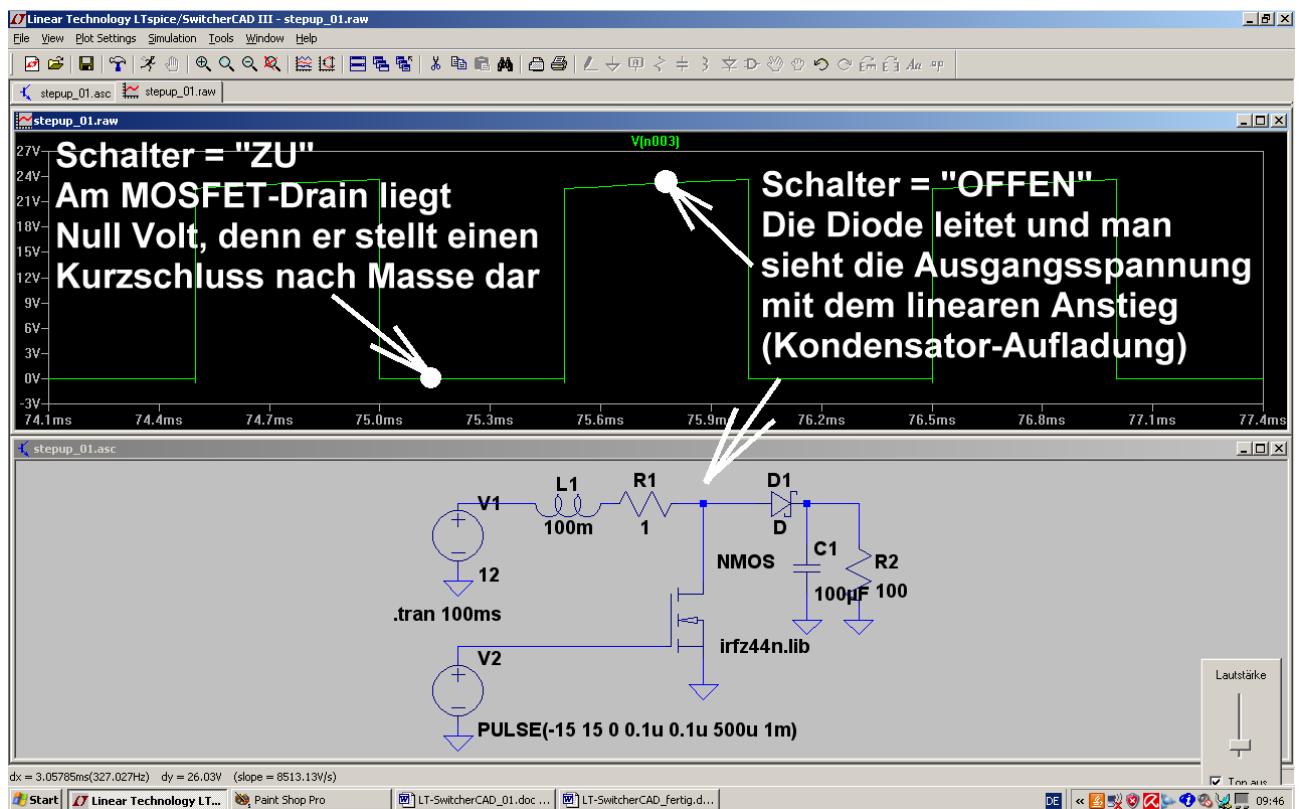
Eingeschaltet wird er mit +15 V, gesperrt dagegen mit -15 V.

Als Simulationszeitraum wählen wir 0.....50ms.



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

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

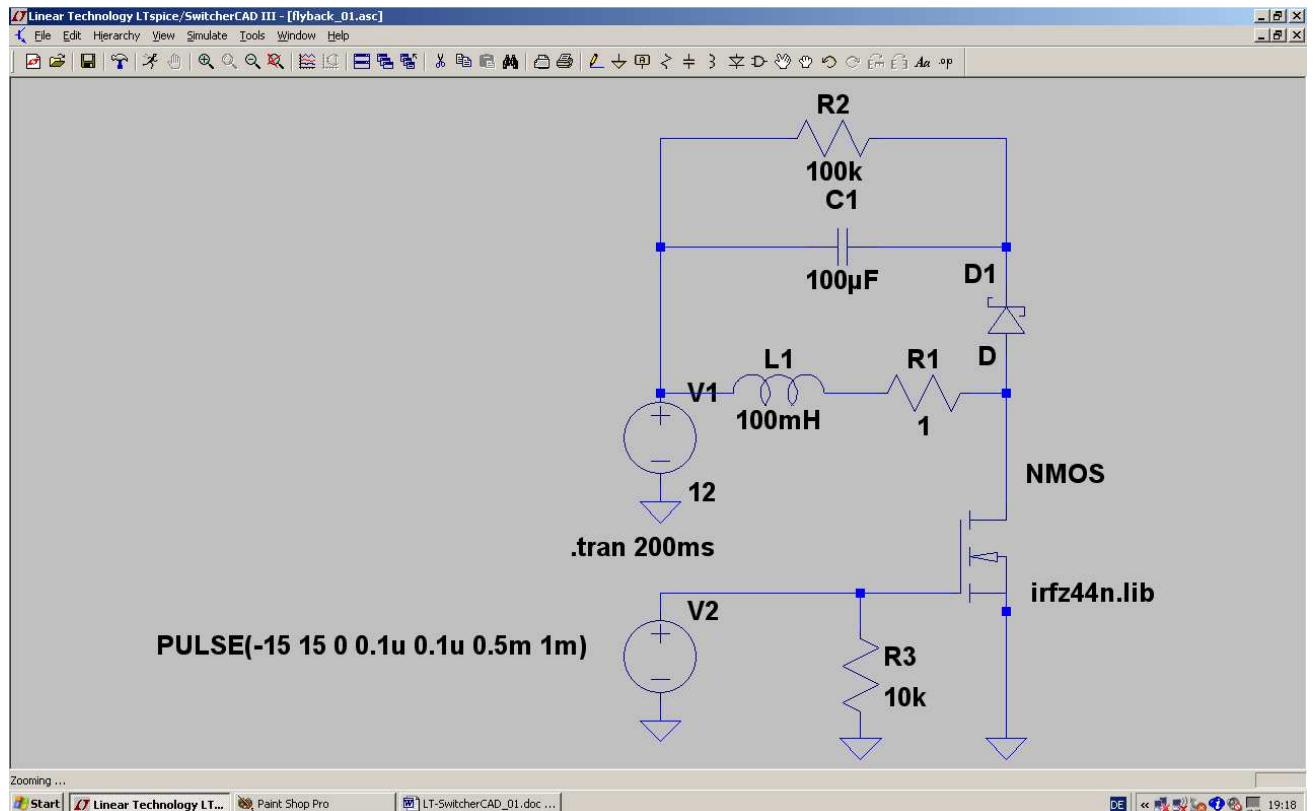


Wer Spaß daran hat, kann sich nun auch die Gate-Spannung auf den Schirm holen und / oder die Simulation für andere Werte der Ansteuerfrequenz, der Induktivität oder des Ladekondensators wiederholen.

Hinweis:

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

11.3. 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:

- 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.
- 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 irgend etwas nachgibt! **Zur Demonstration dieses Effektes wurde der Lastwiderstand auf 100kΩ erhöht.**
- Genau hinsehen: **der Pluspol der Kondensatorspannung (= Ausgangsspannung) ist an der Katode** der verwendeten Schottky-Diode, außerdem ist diese Ausgangsspannung im Normalfall immer deutlich höher als die Betriebsspannung!

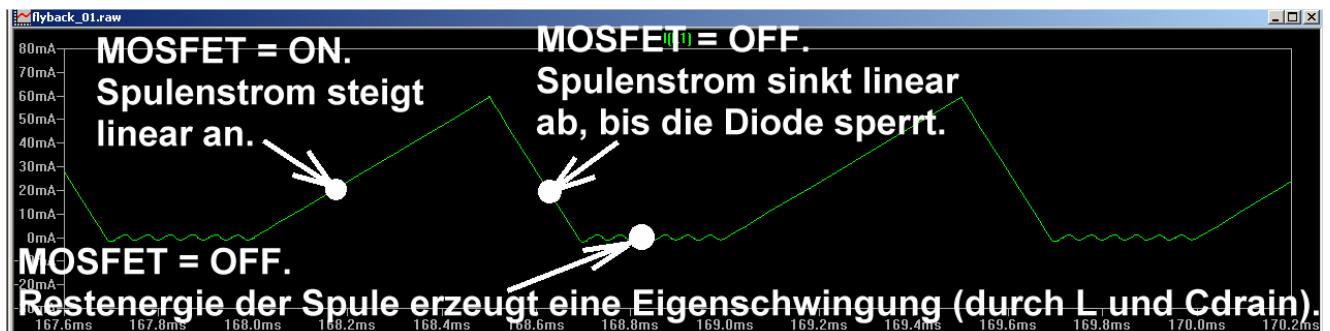
Hinweis:

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

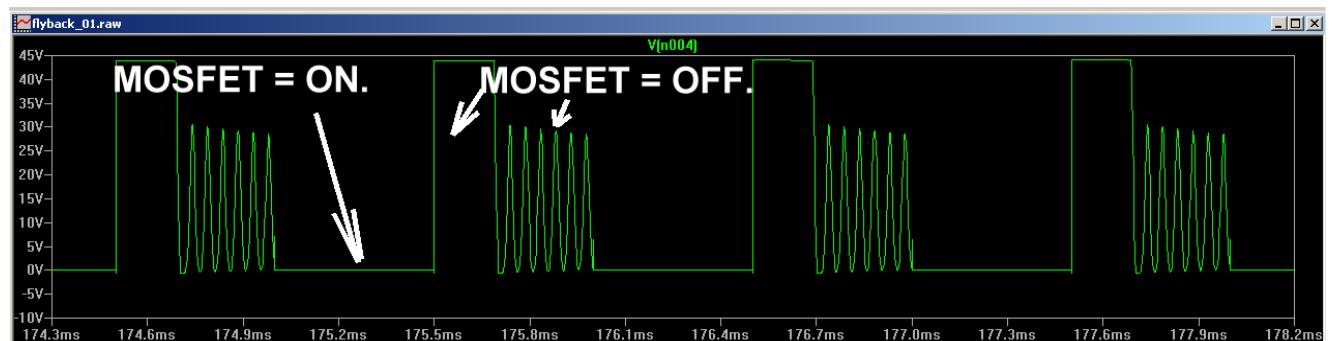
Die obige Schaltung wird nun gezeichnet und simuliert. Als Ansteuersignal dient wieder eine symmetrische Rechteckspannung mit +15 V Maximalwert, -15 V Minimalwert und einer Frequenz von 1 kHz. Da die Schwelldruckspannung der Diode nicht Null Volt beträgt, endet der Aufladevorgang des Ausgangskondensators 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!

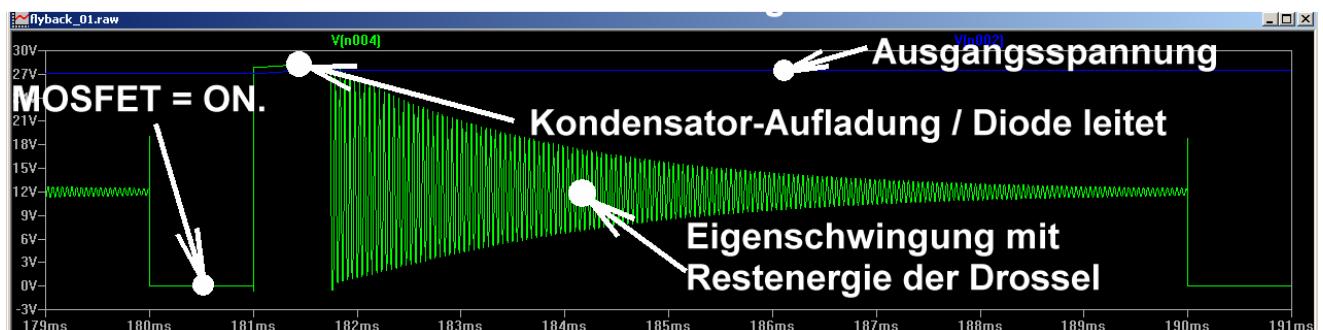
Schauen wir doch mal über die Simulation des Spulenstroms nach, ob das alles stimmt:



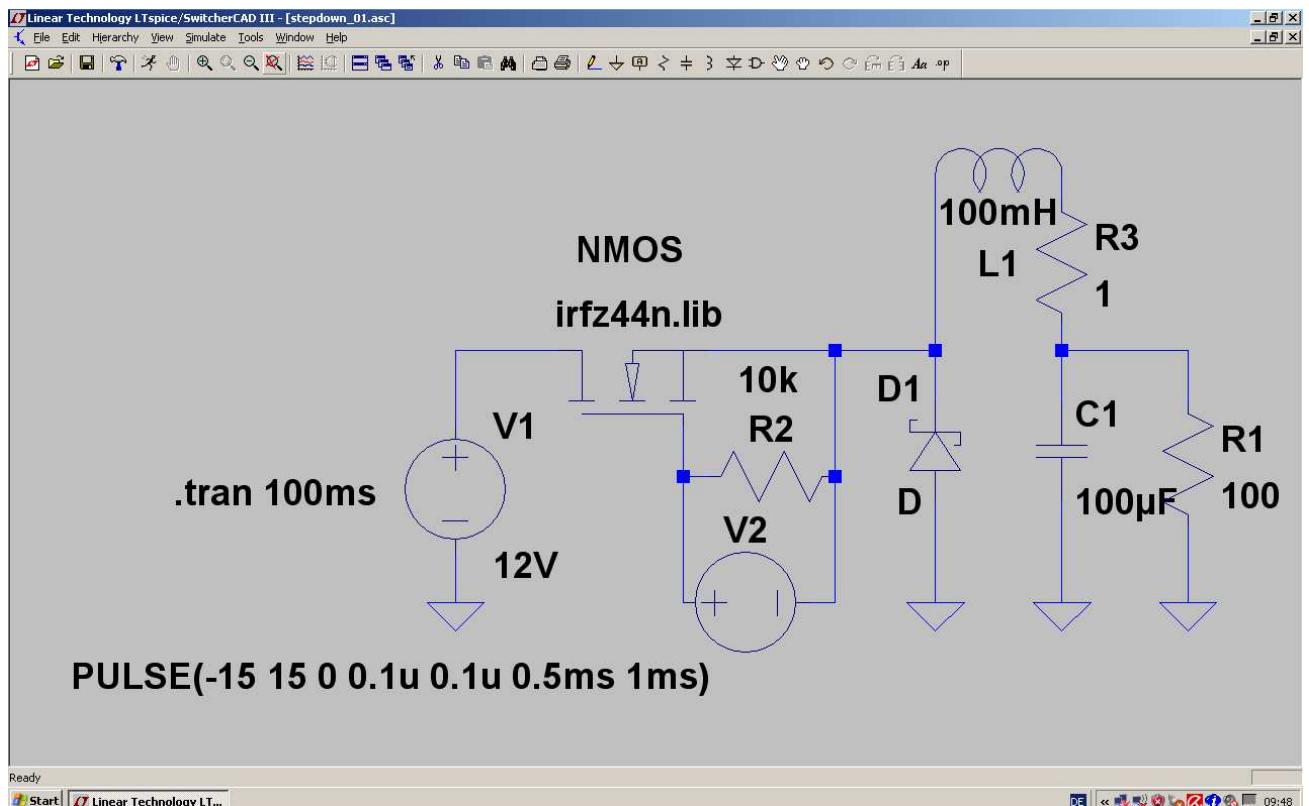
Genau so schön muss man das an der Drainspannung des MOSFETS sehen:



Zur besseren Verdeutlichung hier nochmals eine Simulation mit einer Pulslänge von 1ms und einer Periodendauer von 10ms:



11.4. 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:

- 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.
- 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:

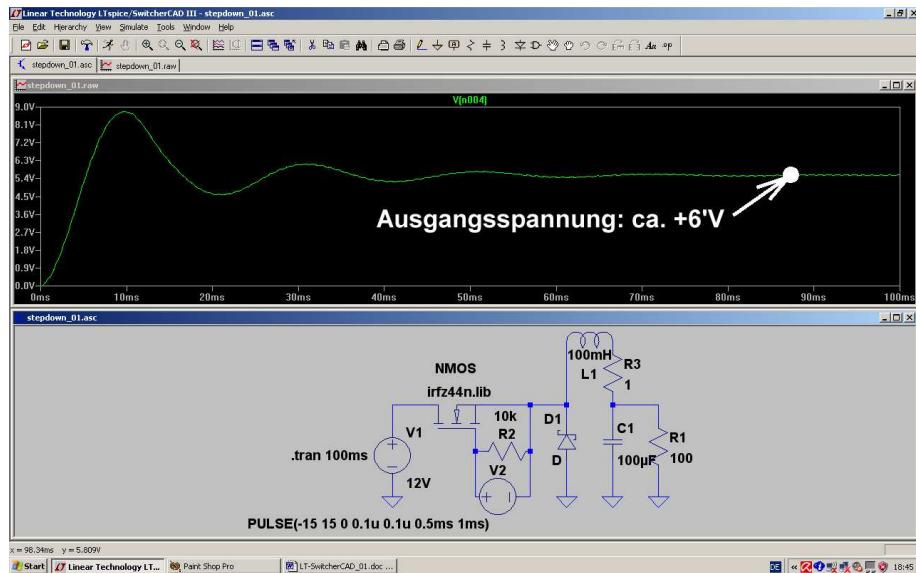
- Die Regelung oder Stabilisierung der Ausgangsspannung erfolgt bei dieser Schaltung immer über das Tastverhältnis (= Verhältnis von FET-Einschaltzeit zur Periodendauer)!
- Die Ausgangsgleichspannung ist bei dieser Schaltung IMMER kleiner als die Eingangsspannung, ihren Wert erhält man, wenn man die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

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

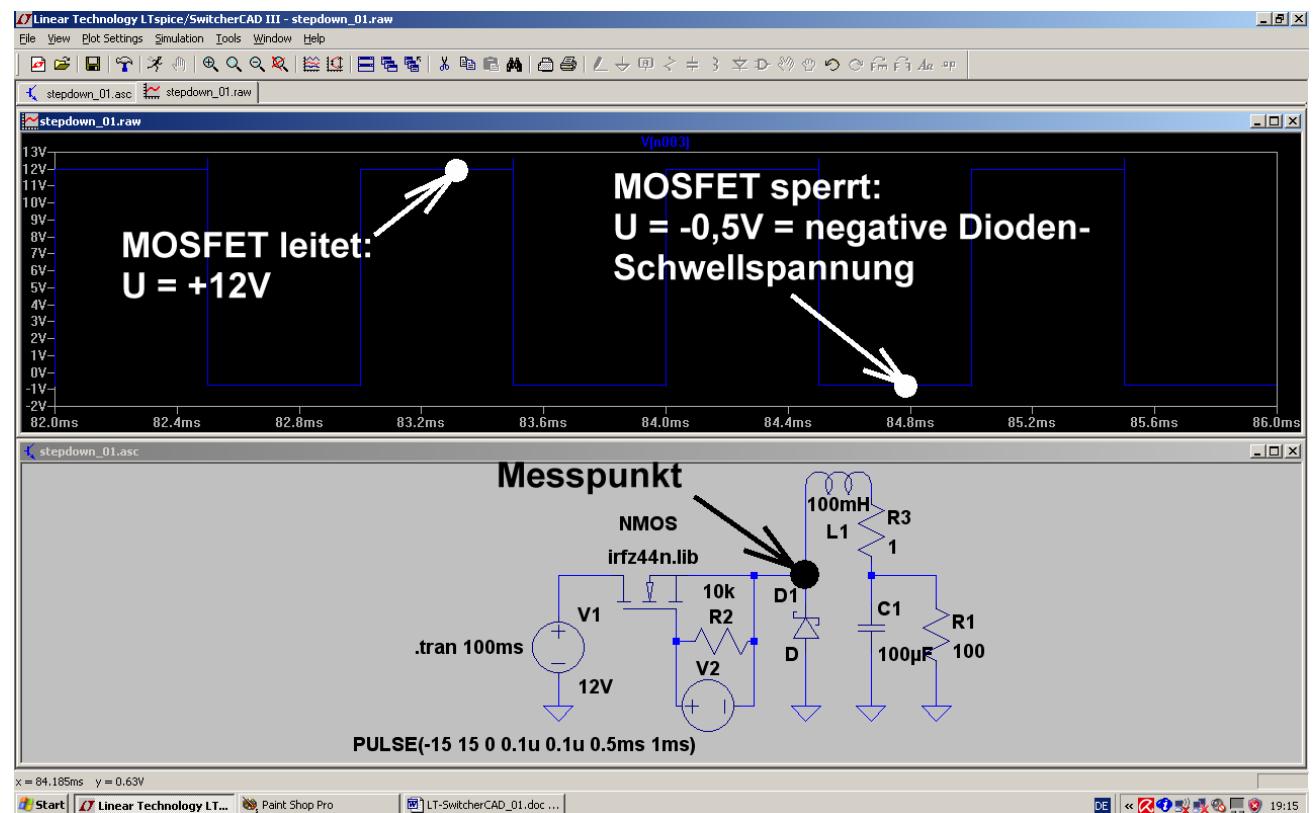
zu erwarten.

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:



12. Achtes Projekt: Phasenanschnitt-Steuerung mit Thyristor

12.1. 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

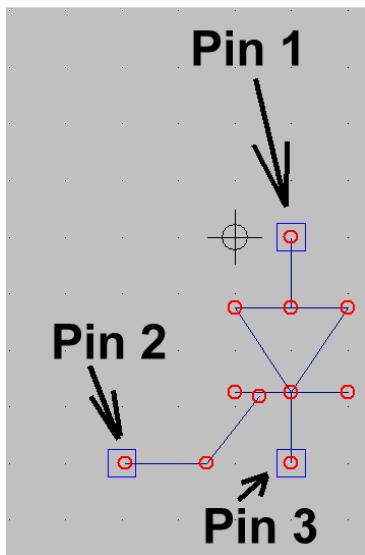
thyristr.lib

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

Aber Vorsicht:

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 eingefügt. Das kann dann endgültig als „thyristr.lib“ am angegebenen Ort gespeichert werden.

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:

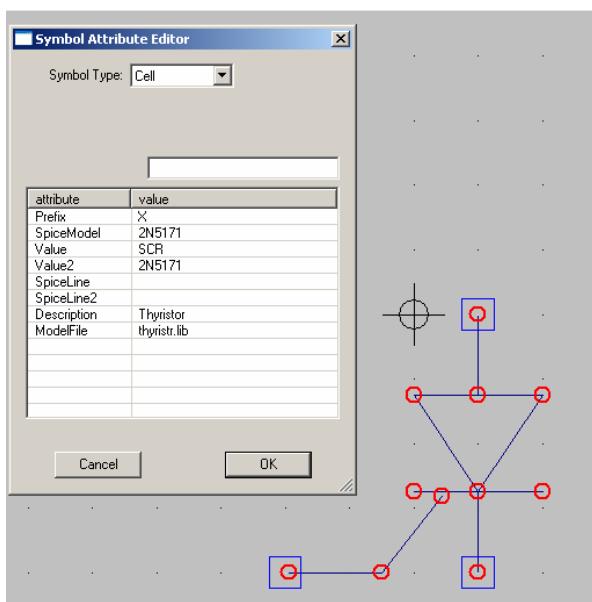
Entsprechend dem SPICE-Modell müssen wir streng die Zuordnung

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.

Präfix X

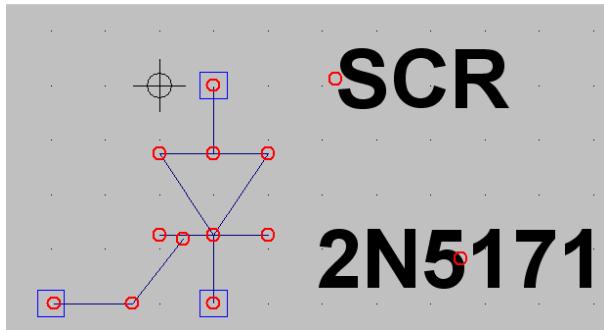
SpiceModel 2N5171

Value SCR

Value 2 2N5171

Description Thyristor

ModelFile thyristr.lib

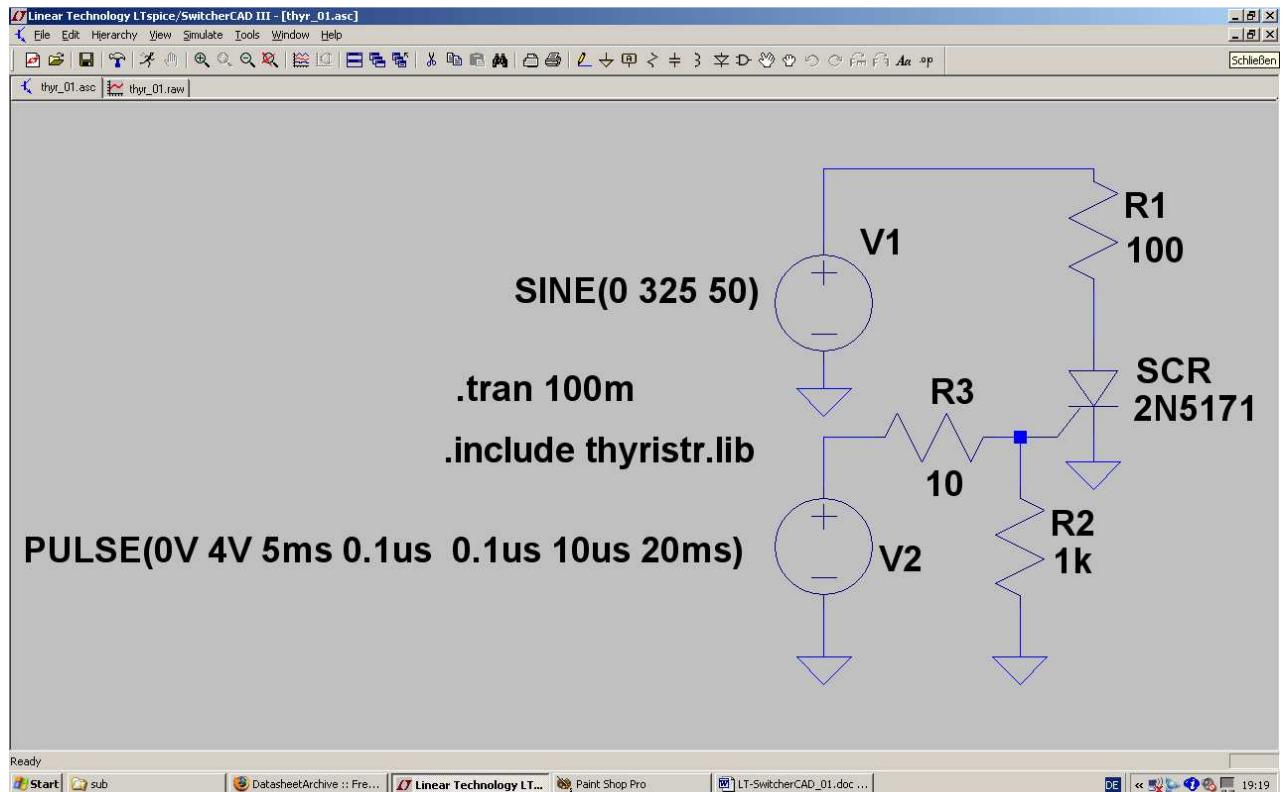


3. Schritt:

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.

12.2. Schalten von Ohm'schen Lasten



Schaltungsbeschreibung:

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

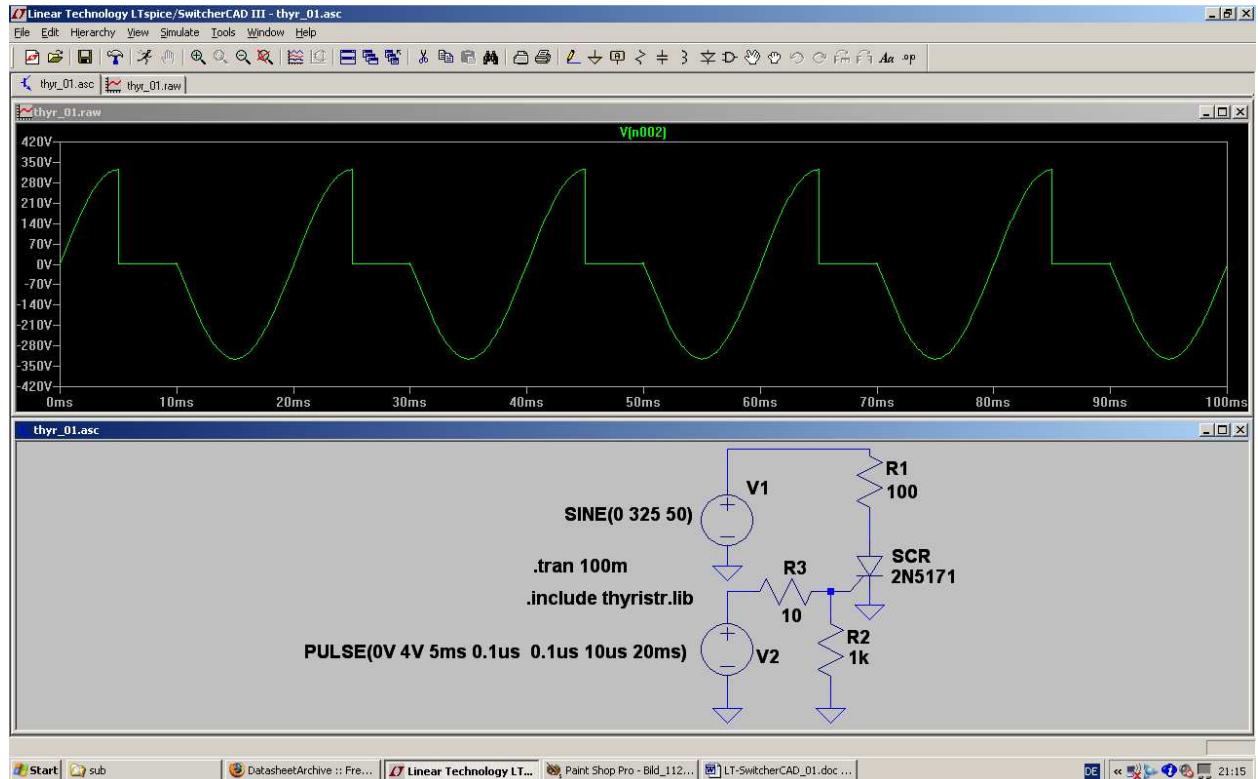
Erläuterungen:

- eine Simulationszeit von 0....100ms wird durch die SPICE-Direktive „.tran 100m“ festgelegt.
- Mit „.include thyristr.lib“ wird die Datei mit den Thyristor-SPICE-Modellen bereitgestellt.
- Die Pulsspannung wird durch folgendes Attribut der Spannung V2 erzeugt:

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

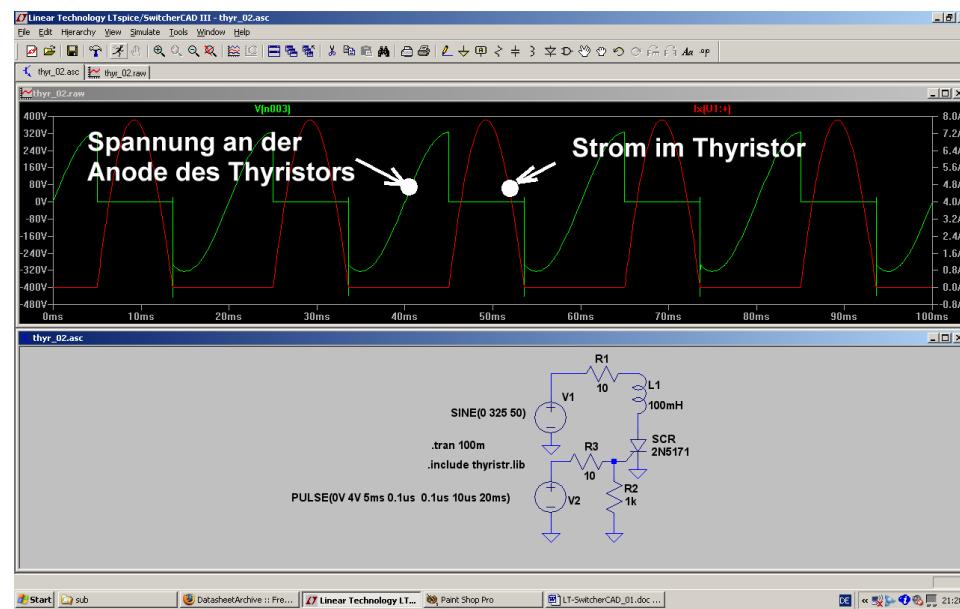
Das ergibt eine Minimalspannung von Null Volt, eine Maximalamplitude von +4V, eine Startverzögerung von 5ms, eine Anstiegs- und Abfallzeit von je 100ns, eine Pulslänge von 10 Mikrosekunden und eine Periodendauer von 20ms

Und so sieht die Spannung an der Anode des Thyristors aus:



12.3. 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 los werden möchte!) dagegen aufzukommen und 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.

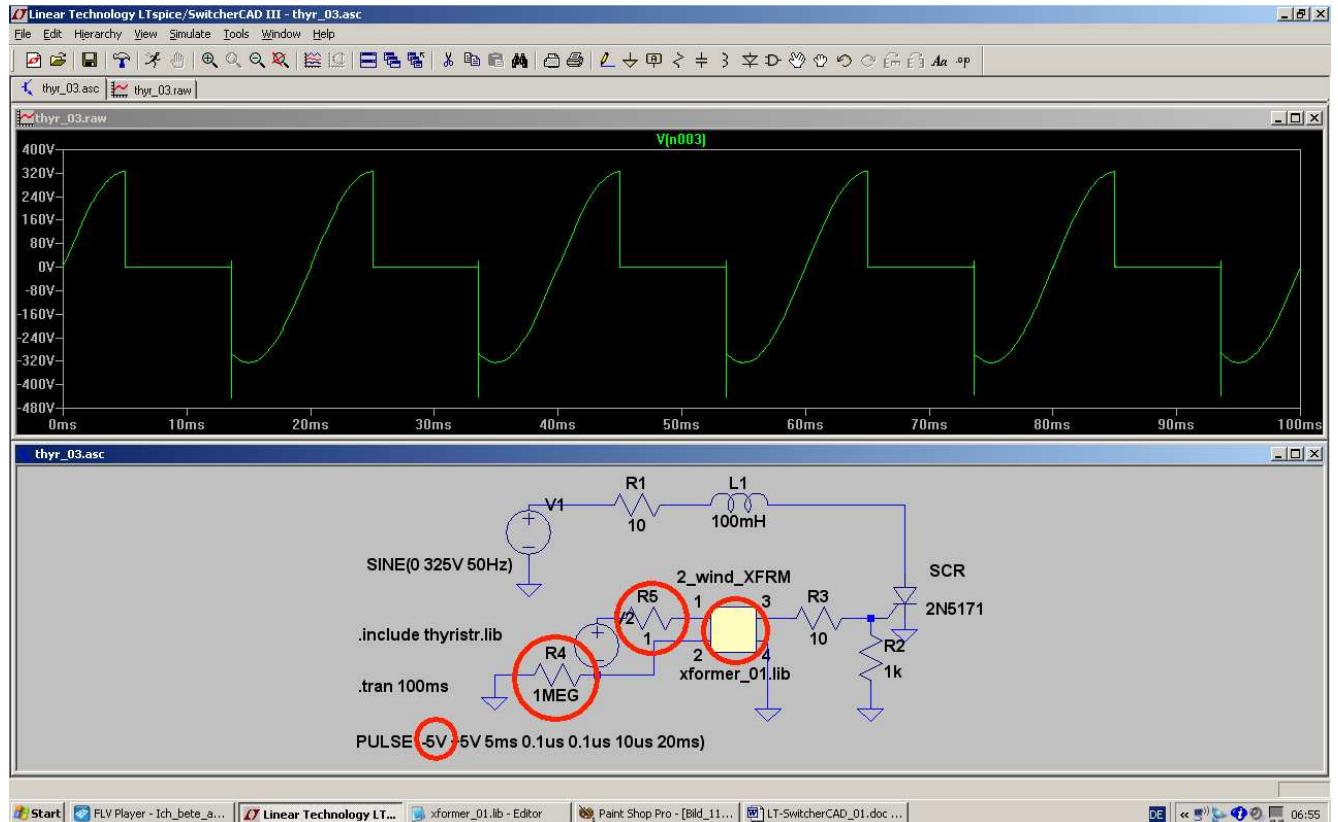


Also schalten wir mal $L = 100 \text{ mH}$ in Reihe zu einem Lastwiderstand von 10Ω und drücken dann den Simulationsbutton. Sehr gut ist zu sehen, wie der Thyristor jetzt nicht mehr wie vorher beim Nulldurchgang abschalten kann. Die Induktivität liefert nämlich noch solange Strom, bis die in ihr gespeicherte magnetische Energie verbraucht ist.

Ein kleiner Energierest erzeugt am Ende dann noch eine kurze gedämpfte Schwingung.

12.4. Zündung des Thyristors über einen Gate-Transformer

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“). Deshalb nehmen wir uns die vorige Schaltung her, bauen einen solchen Zündtrafo ein und sehen nach, ob alles noch wie gewünscht funktioniert.



Einige Details:

- 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 = 1 Mega-Ohm eingebaut werden**.
- 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.
- Eine Spannungsquelle darf bekanntlich nie direkt an eine Induktivität gelegt werden -- das hatten wir schon. Deshalb wird der **Widerstand R5 mit 1Ω zwischen Quelle und Primärseite des Trafos eingefügt**.
- Als Transformator holen wir das vor längerer Zeit selbst erstellte Teil „**xformer_01**“ aus der Bibliothek und bauen es in die Schaltung ein.

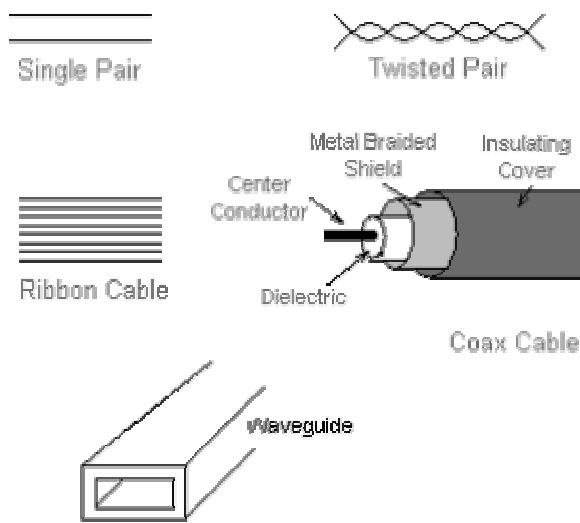
13. Neuntes Projekt: Echos auf Leitungen

13.1. 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:



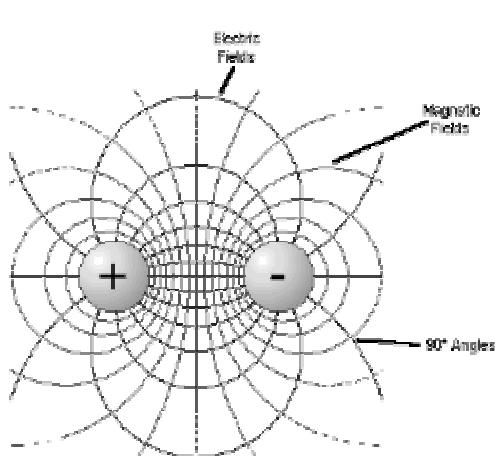
Das „**Single Pair**“ kann sich Jeder vorstellen: eine Hinleitung und eine Rückleitung, allerdings exakt in Kunststoff parallel geführt.

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

Das **Koaxkabel** sehen wir als Antennenleitung 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 „Hohlräum“, 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 in jedem Physikbuch nachlesen. (Siehe nebenstehendes Bild).

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:

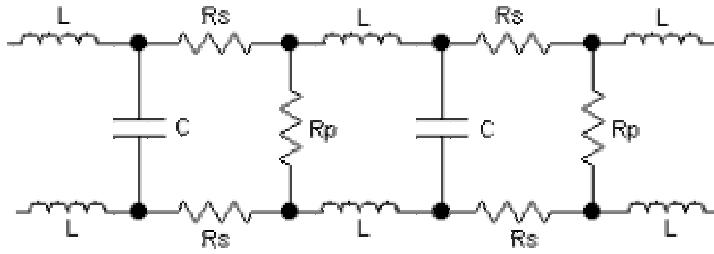
Jedes noch so kurze Leitungsstück besitzt eine bestimmte Eigenkapazität, die linear mit der Leitungslänge zunimmt!

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:

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

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 weg bewegt, 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:

$$v_{\text{Kabel}} = \frac{\text{Lichtgeschwindigkeit}}{\sqrt{\epsilon_r}}$$

wobei ϵ_r (...in Amerika heißt das auch mal „k“..) 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.

13.2. Echos auf Leitungen

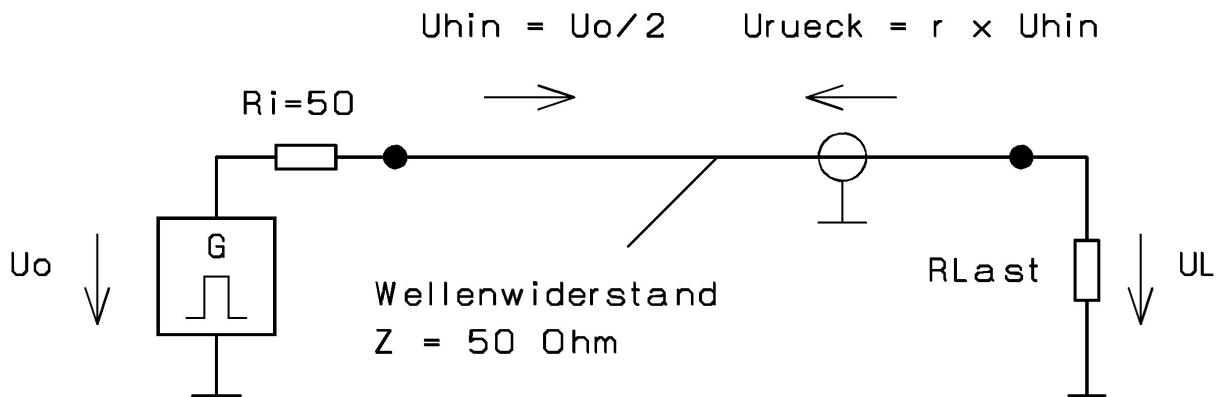
Bei hohen Frequenzen lassen sich Ströme und Spannungen nicht mehr so einfach messen, außerdem klappt auch die Sache mit Leerlauf- oder Kurzschluss-Messung zum Bestimmen des Innenwiderstandes einer Quelle nicht mehr richtig. Die Systembeschreibung und -berechnung geht deshalb dort von Größen aus, die auch **bei höchsten Frequenzen (z. B. bei $f = 100\text{GHz}$) leicht messbar sind und man benutzt 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 $R_i = R_a$) 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:

- Bei größerer Kabellänge „merkt man wegen der endlichen Signal-Ausbreitungsgeschwindigkeit (in Luft sind es in einer Nanosekunde 30cm) im ersten Moment noch nichts vom Verbraucher“.
- 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 $R_i = R_a$) und die vom Kabeleingang aufgenommene **maximal abgebare Wirkleistung** (zu der die Spannungsamplitude $U_0/2$ gehört) macht sich mit der Kabelgeschwindigkeit auf den Weg in Richtung Verbraucher.
- 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 ($R_{\text{LAST}} = Z = 50\Omega$) 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:

$$r = \frac{(Z_{\text{Last}} - Z)}{(Z_{\text{Last}} + Z)} = \frac{U_{\text{rueck}}}{U_{\text{hin}}} \quad \text{bzw.} \quad U_{\text{rueck}} = r \cdot U_{\text{hin}}$$

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

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:

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

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 ($Z = 50\Omega$, Typ RG58) mit einem Abschlusswiderstand von 75Ω verbunden. Die Dielektrizitätskonstante der Kabel-Innenisolation beträgt $\epsilon = 2,25$.

Bestimmen Sie die Signalverläufe

a) am Kabeleingang

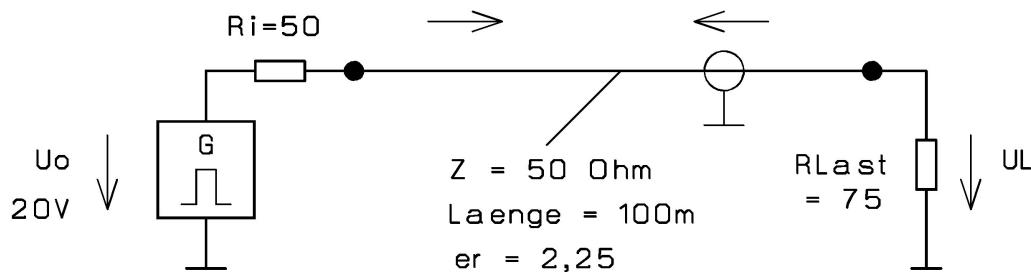
b) in Kabelmitte

c) am Kabelende

$$U_{\text{hin}} = U_0 / 2$$

$$= 10\text{ V}$$

$$\text{Urueck} = r \times U_{\text{hin}}$$



Lösung:

a) Bestimmung des Reflektionsfaktors:

$$r = \frac{R_{\text{LAST}} - Z}{R_{\text{LAST}} + Z} = \frac{75\Omega - 50\Omega}{75\Omega + 50\Omega} = +0,2$$

b) Bestimmung der Kabelgeschwindigkeit:

$$v_{\text{Kabel}} = \frac{c}{\sqrt{\epsilon_r}} = \frac{3 \cdot 10^8 \text{ m}}{\sqrt{2,25 \text{ s}}} = 2 \cdot 10^8 \frac{\text{m}}{\text{s}}$$

c) Signal-Laufzeit für 100 m Kabellänge:

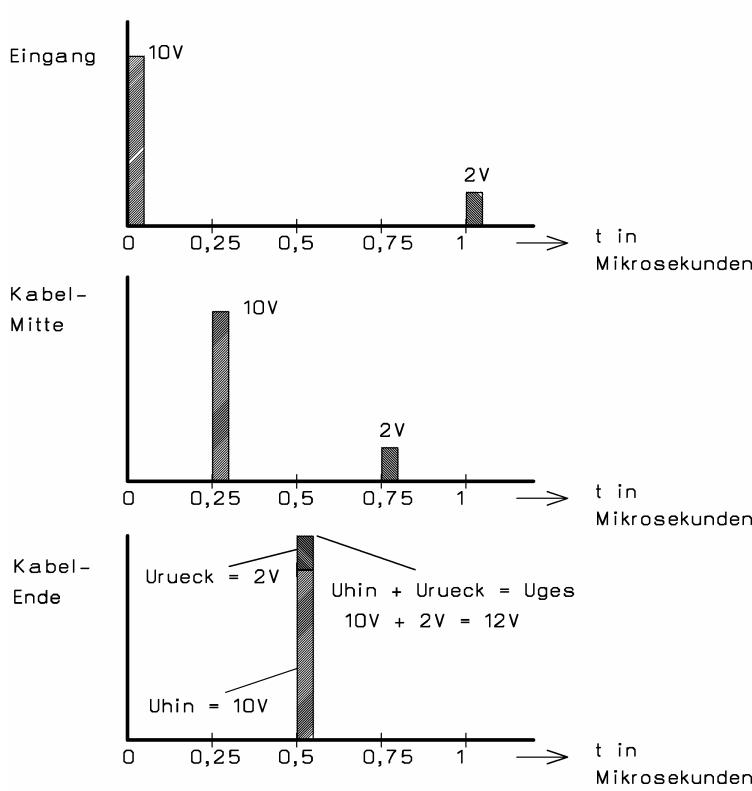
$$t_{\text{Delay}} = \frac{100\text{ m} \cdot \text{s}}{2 \cdot 10^8 \text{ m}} = 0,5 \cdot 10^{-6} \text{ s}$$

d) Hinlaufende Welle:

$$U_{\text{hin}} = \frac{U_0}{2} = \frac{20\text{ V}}{2} = 10\text{ V}$$

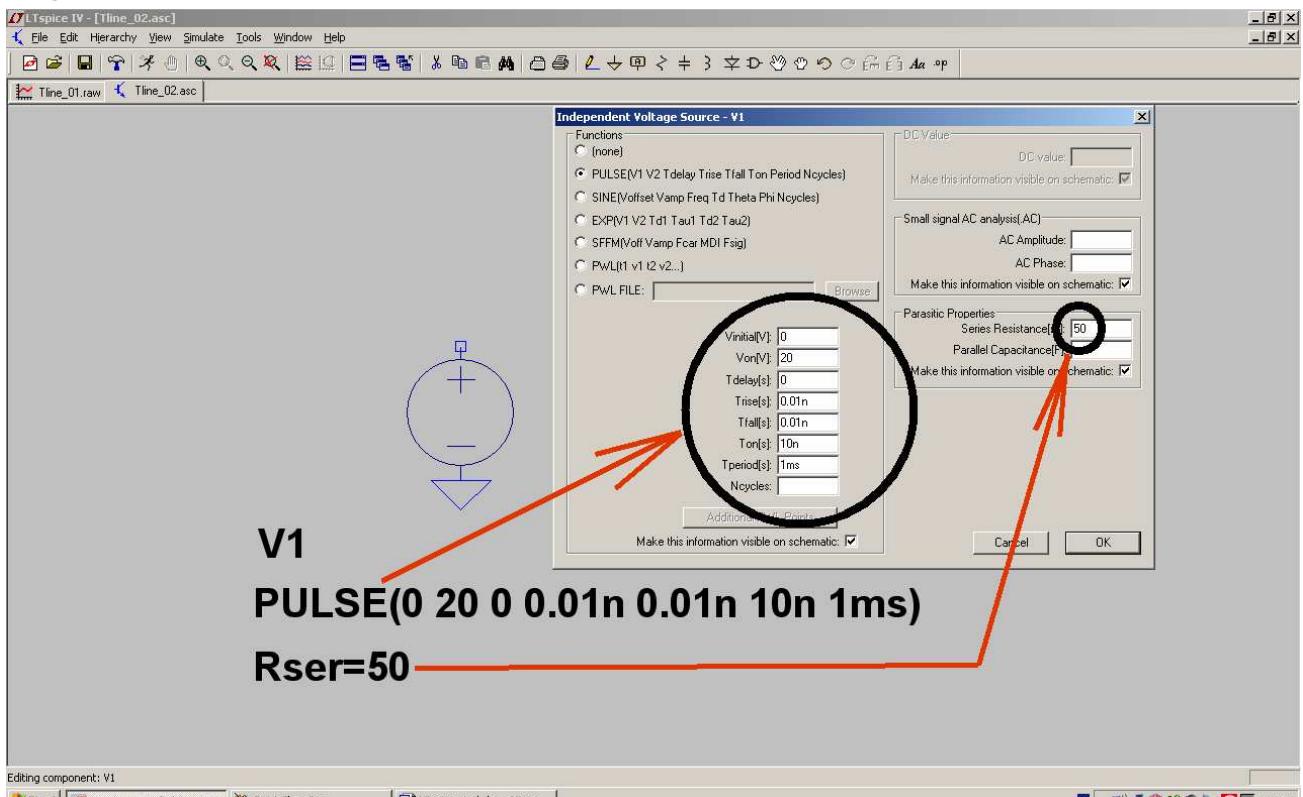
e) Rücklaufende Welle:

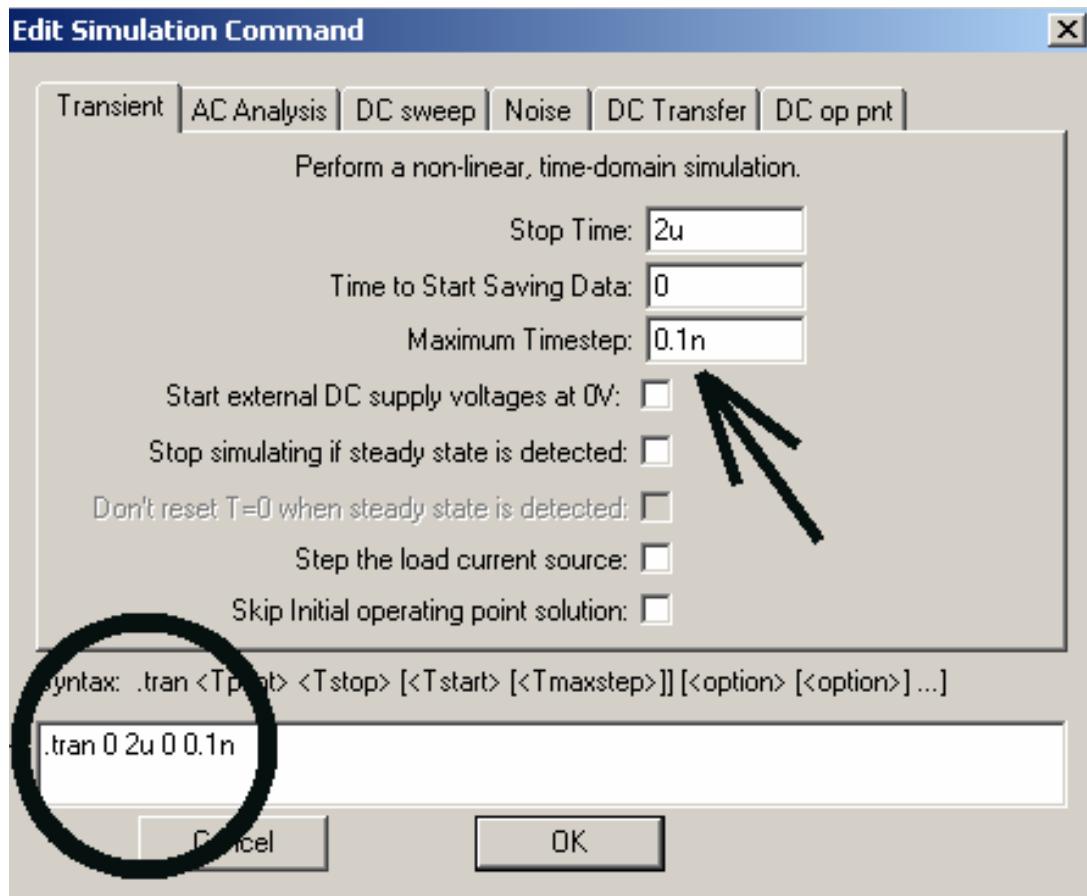
$$U_{\text{rück}} = r \cdot U_{\text{hin}} = 0,2 \cdot 10\text{ V} = 2\text{ V}$$



13.3. 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 Pikosekunden extrem kurz**. Also sorgen wir dafür, dass der größte mögliche Schritt **100 Pikosekunden** beträgt:

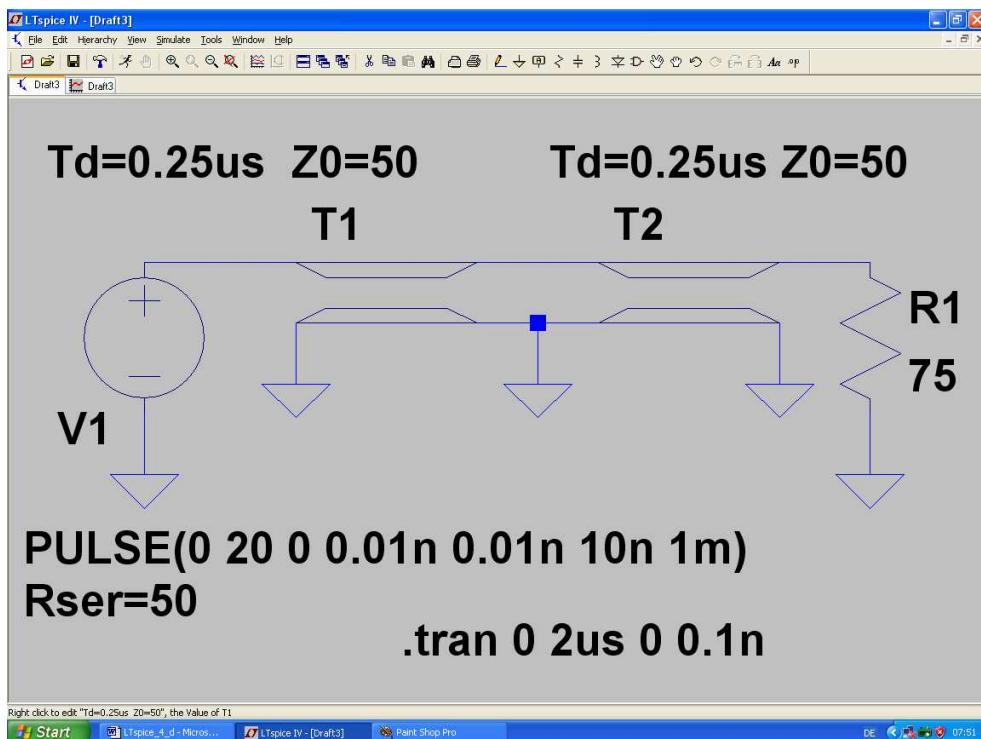
Jetzt geht es an die „**Transmission Line**“. Sie findet sich als „**tline**“ im Bauteilvorrat. Allerdings müssen wir vor dem Einsatz noch etwas rechnen, denn wir können bei ihr zwar die „**Kabellaufzeit“ als „delay time“** programmieren, aber **nicht die Kabellänge**.

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

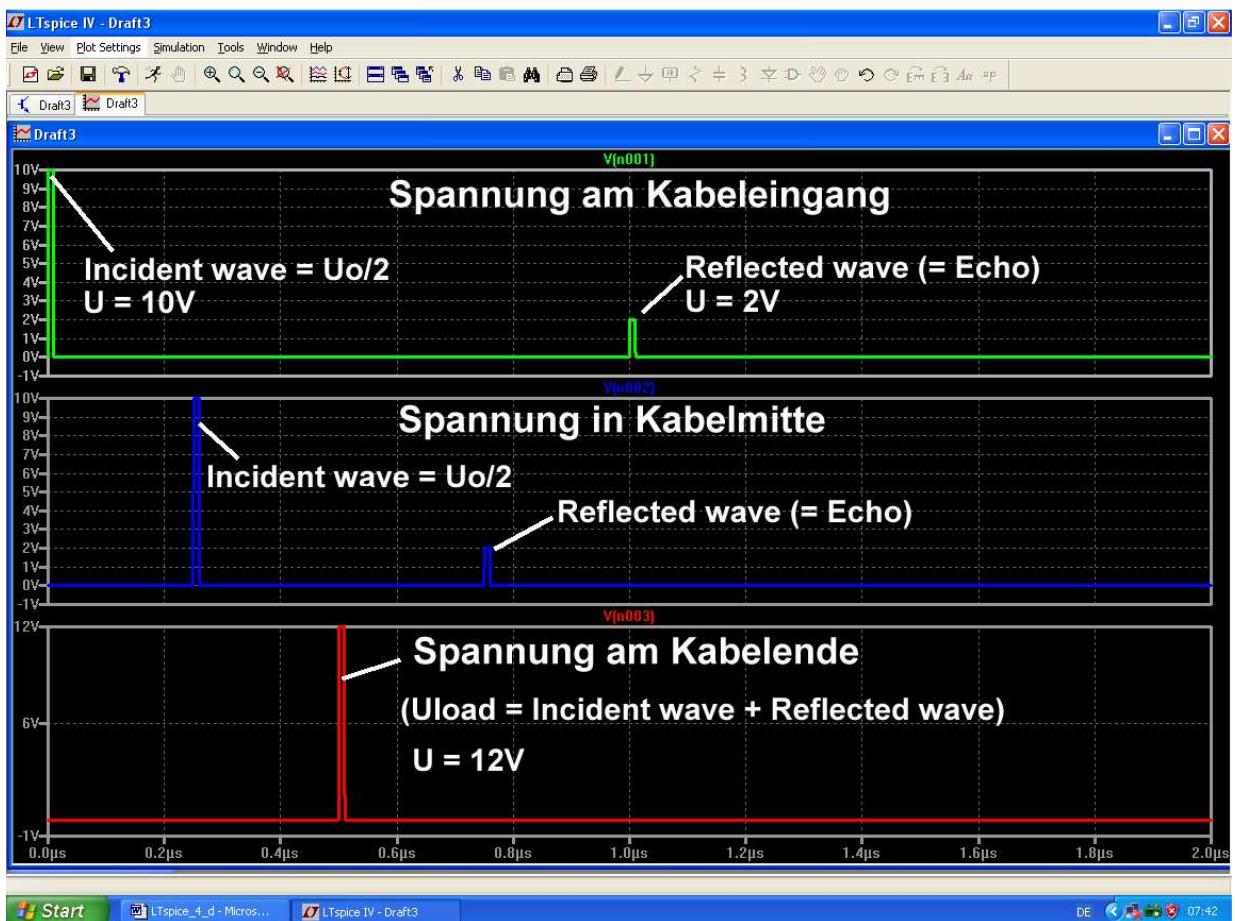
$$t_{\text{delay}} = \frac{l}{v_{\text{cable}}} = \frac{100m \cdot s}{2 \cdot 10^8 m} = 0,5 \mu s$$

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 nun so aussehen (Bitte nochmals alle Einstellungen kontrollieren...):

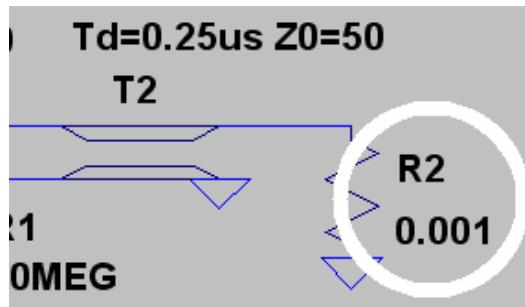


Nach der Simulation werden die drei messbaren Signale der Schaltung in getrennten Diagrammen dargestellt. Dann sieht das so aus und beim Vergleich mit Kapitel 13.2 kann man nur sagen: „Na also...“.



13.4. Leerlauf oder Kurzschluss als Last am Kabelende

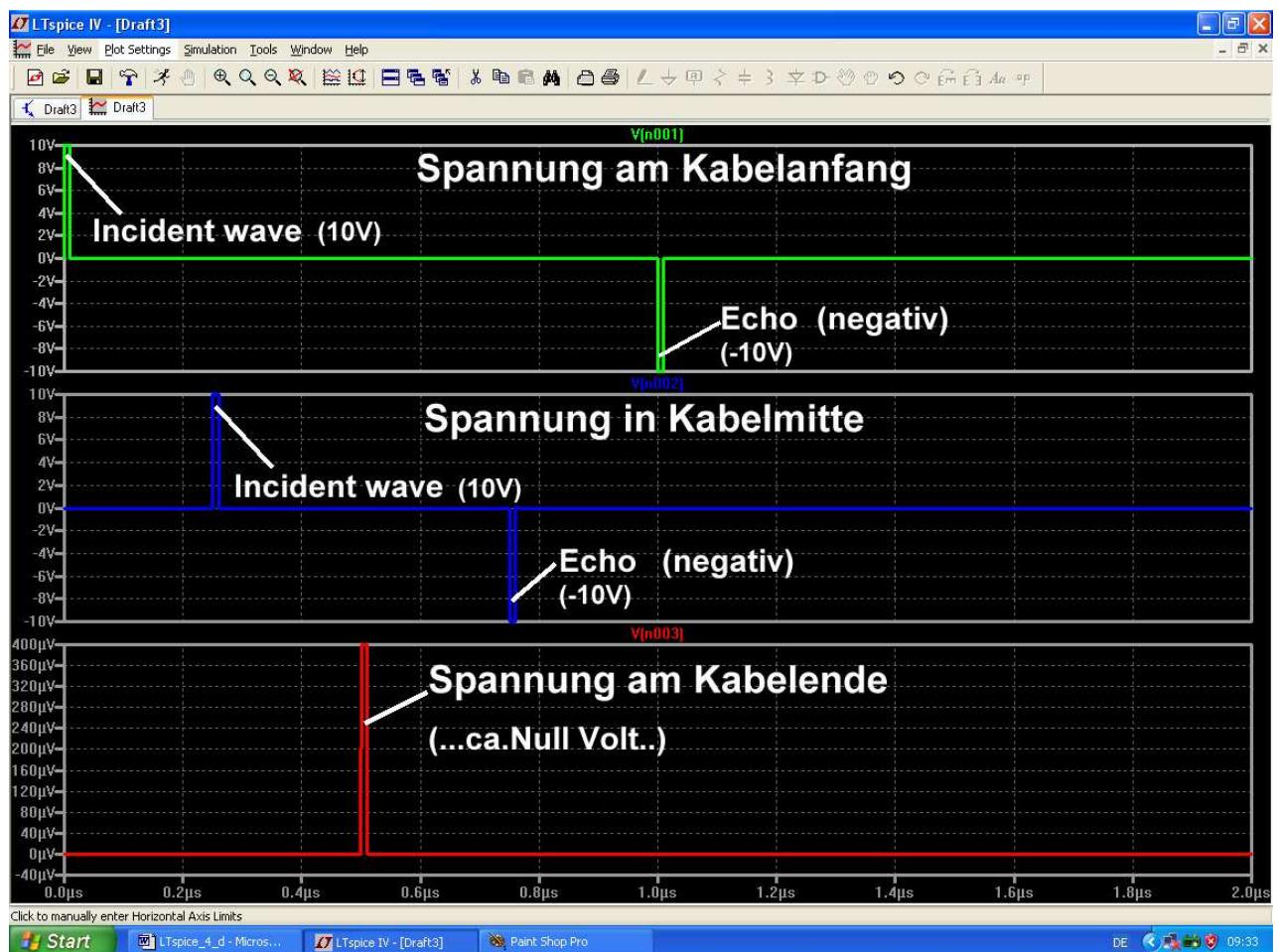
Da nehmen wir unsere bisherige Simulationsschaltung her und ändern nur den Wert des Abschlusswiderstandes.



a) Kurzschluss:

Ein Wert von $0,001\Omega$ (also 1 Milli-Ohm) ist hier realistisch und sinnvoll.

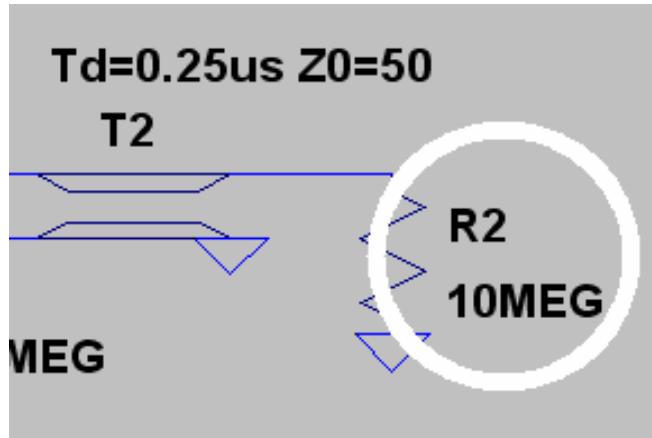
Damit erhalten wir folgende Signale:



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.

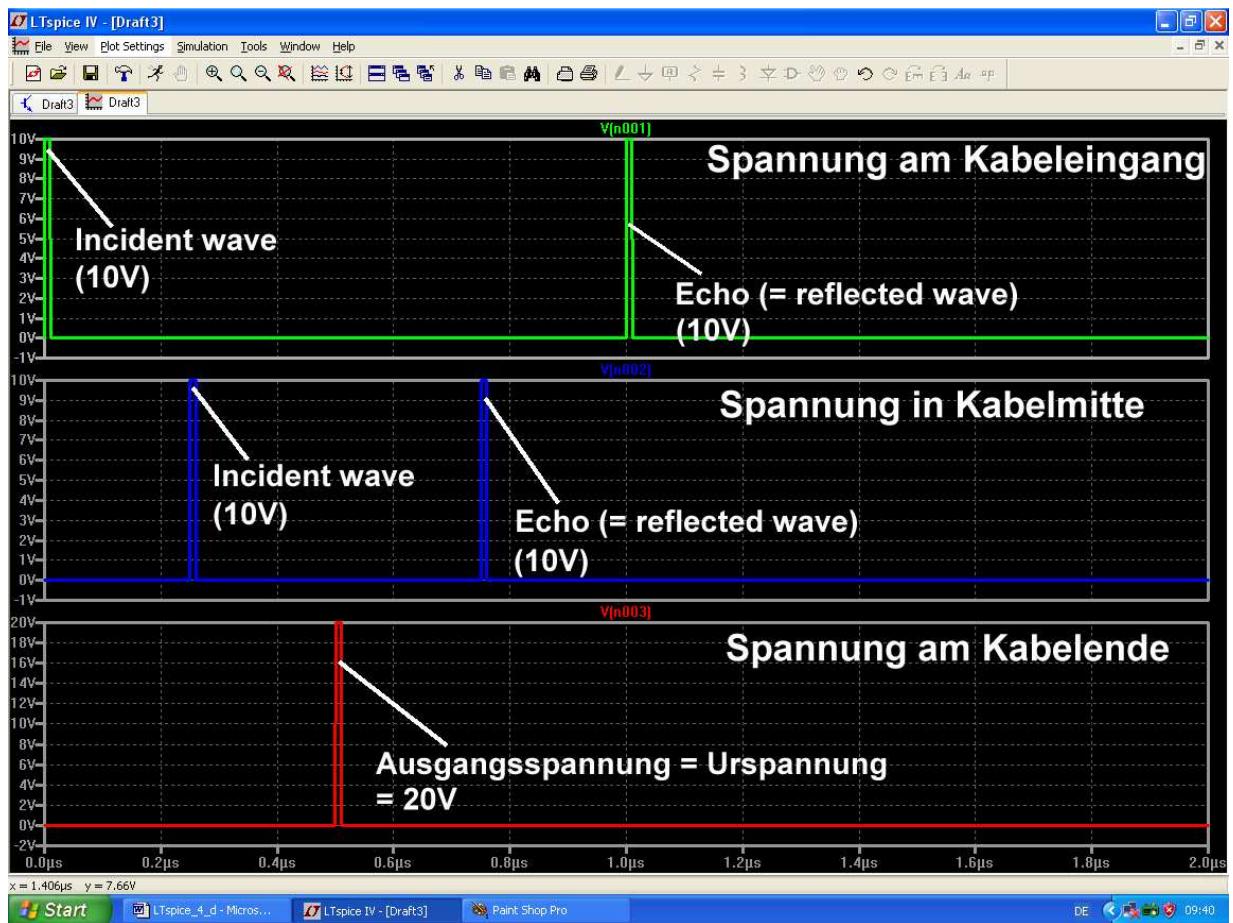
b) Leerlauf am Kabelende



Jetzt reicht ein Widerstand mit $10\text{M}\Omega$ 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 bringt die Simulation ab!

Und das bekommen wir zu sehen:



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.

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

13.5.1. Wie simuliere ich RG58-Kabel?

Dazu ist ein anderes Bauteil erforderlich, welches als

Itline

(= 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 Itline-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 $1,5\Omega$ 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 „ $1,5\Omega$ pro 1m“ sind eine erste grobe Schätzung für den Einstieg. Wie man damit zum richtigen Wert kommt, kriegen wir gleich....

13.5.2. Simulation der Kabeldämpfung bei 100MHz

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

Frequenz frequency MHz	Dämpfung typ. Meßwerte <i>attenuation typ. measured values</i> 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	20	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		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:

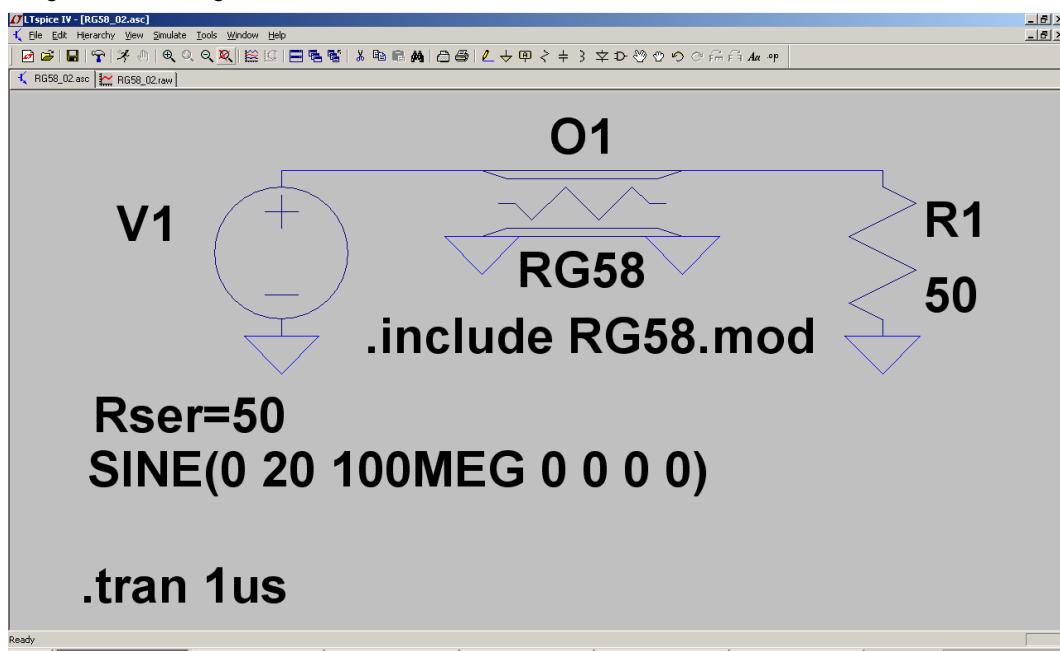
Laut obiger Tabelle beträgt die typische Kabeldämpfung bei f = 100MHz etwa 14,6dB pro 100m. Damit ist die Ausgangsspannung am Kabelende um den Faktor

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

kleiner.

Schritt 2:

Wir versorgen einen 50Ω-Abschlußwiderstand über ein 100m langes RG58-Kabel und speisen eine sinusförmige Spannung ins Kabel ein. Die Spannungsquelle hat einen Innenwiderstand von 50Ω und eine Urspannung von 20V. Damit -- und mit der verlustbehafteten Transmission Line „Iline“ 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

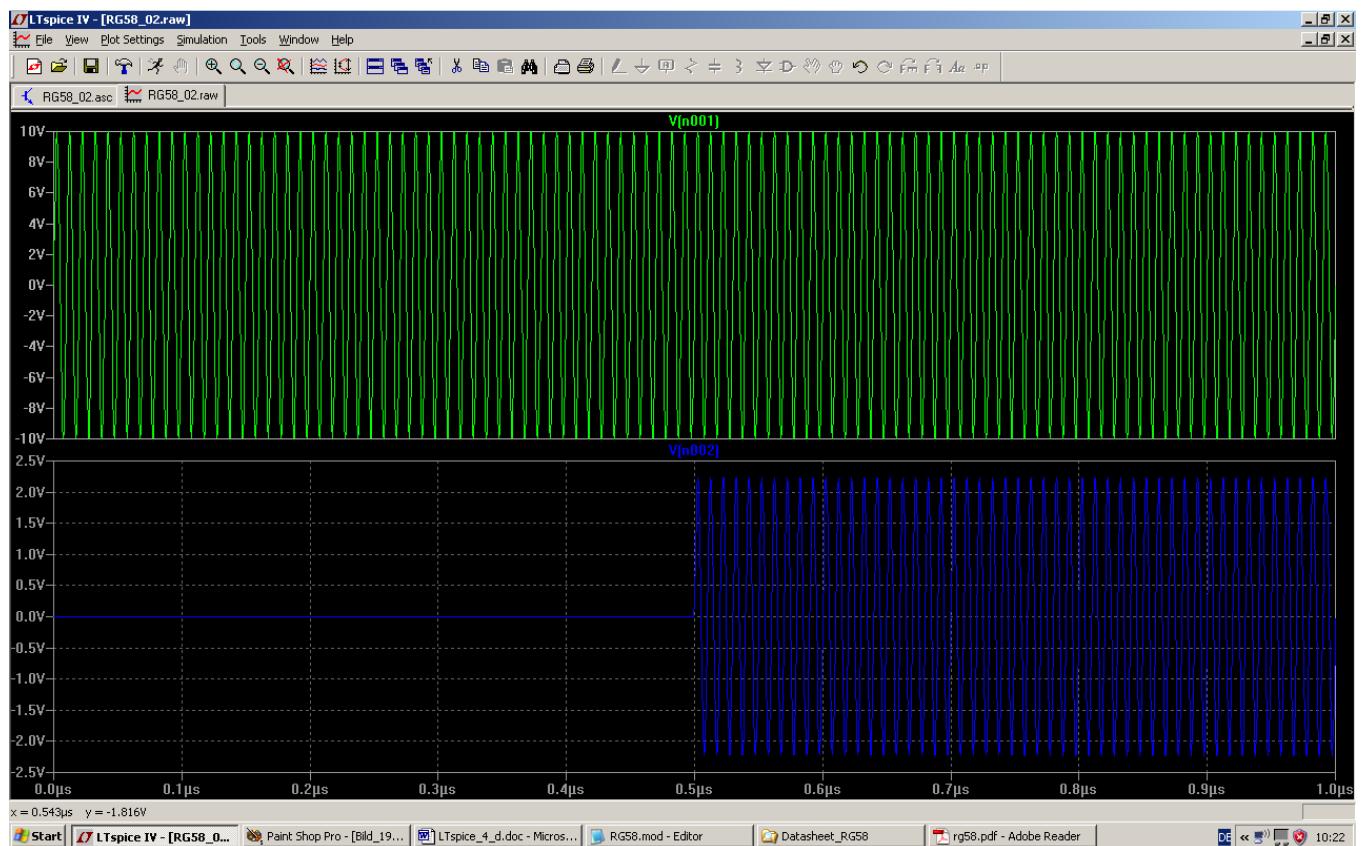
.include RG58.mod

auf dem Bildschirm erscheint.

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

Schritt 3:

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



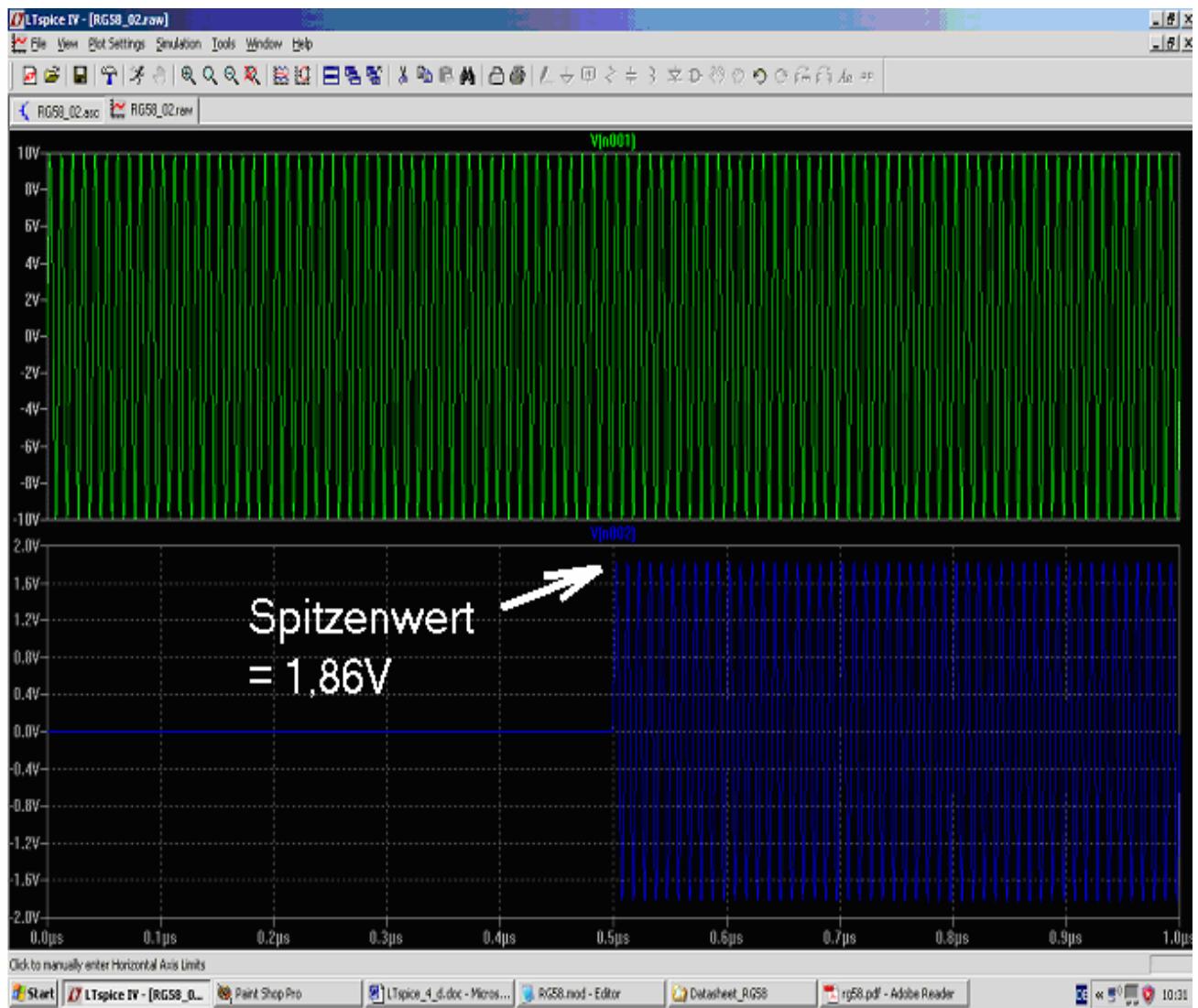
Man kann nun sehr schön die Signal-Laufzeit von 0,5 Mikrosekunden erkennen, die zu einer Kabellänge von 100m gehört.

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 von R = 1,7**.

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

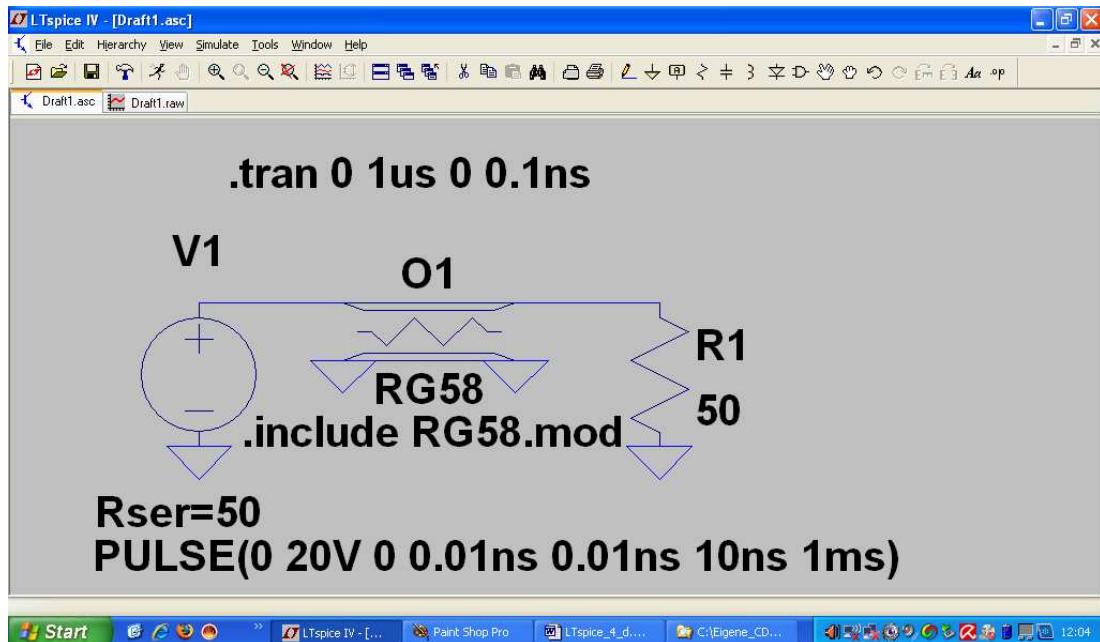


13.5.3. Speisung der RG58-Leitung mit einer Pulsspannung

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

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

an und schauen, was passiert:

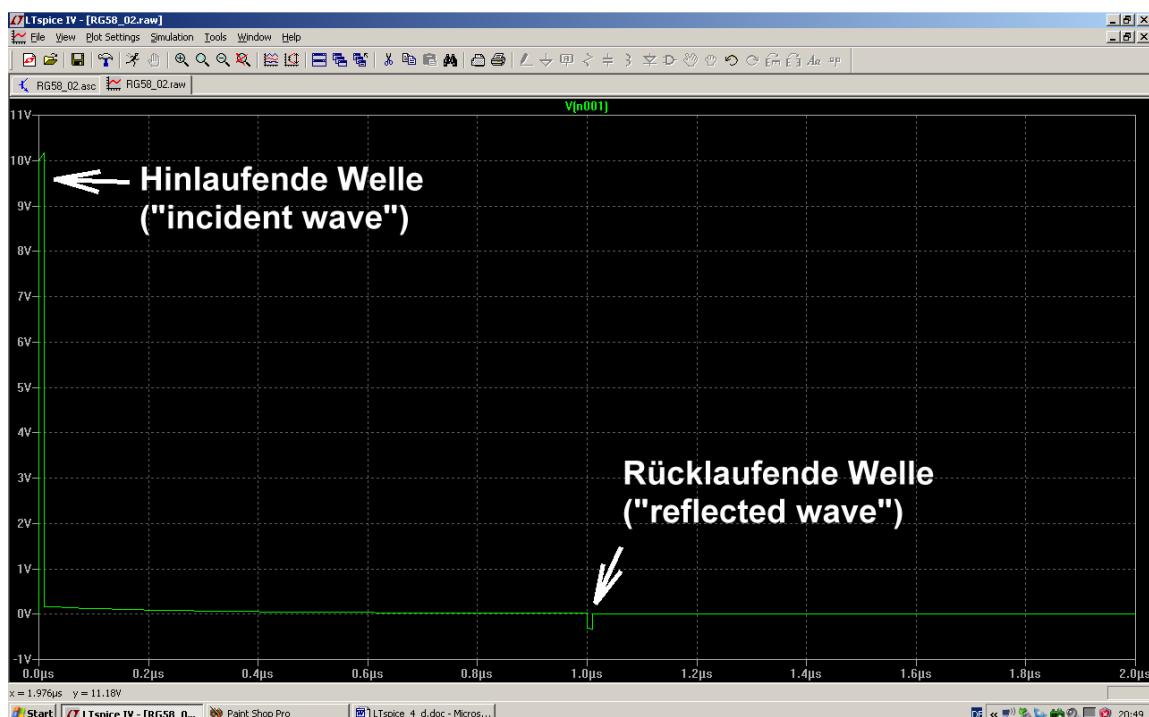
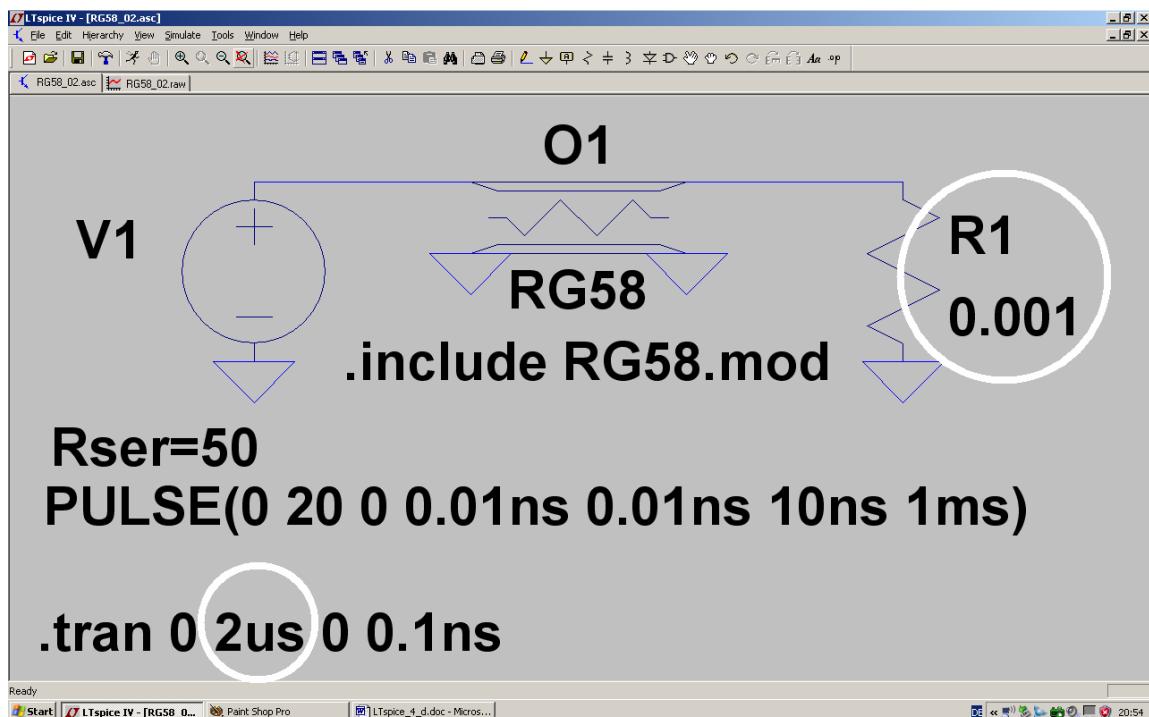


Auch hier weist die Ausgangsspannung den erwarteten Spitzenwert von 1,86V auf -- kein Wunder, denn die Dämpfung wird sich nicht ändern, da „R = 1,7 Ω / m“ konstant bleibt. So werden alle im Pulssignal enthaltenen Oberwellen um denselben Faktor gedämpft und damit wird zwar die Pulsamplitude kleiner, aber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

13.5.4. 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...



14. Zehntes Projekt: S-Parameter

14. 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

$$a = \sqrt{P_{\text{hin}}} = \sqrt{\frac{U_{\text{hin}}^2}{Z}} = \frac{U_{\text{hin}}}{\sqrt{Z}} = \frac{\left(\frac{U_0}{2}\right)}{\sqrt{Z}}$$

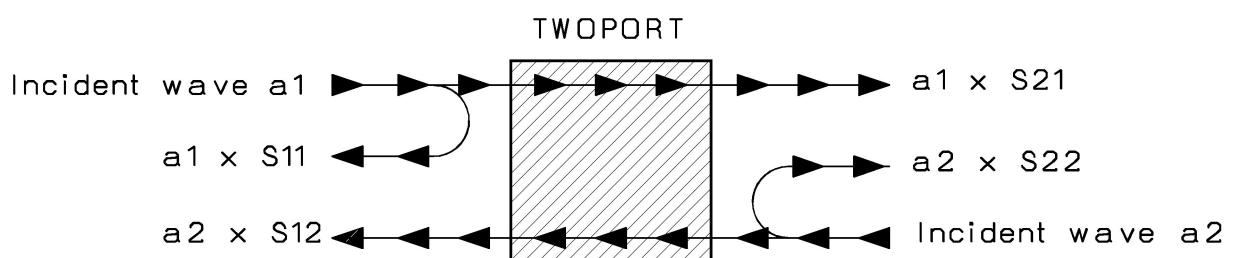
Entsprechend gilt für die „**rücklaufende Welle b**“:

$$b = \sqrt{P_{\text{rück}}} = \sqrt{\frac{U_{\text{rück}}^2}{Z}} = \frac{U_{\text{rück}}}{\sqrt{Z}}$$

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 an einem Port wird bei einem solchen „TWOPORT“ stets auch ein Signal am anderen Port erzeugen. Diese Wirkung lässt sich ebenfalls leicht messen und stellt die zweite Größe zur Beschreibung der Baustein-Eigenschaften dar.

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 a₁** aus (= „incident wave“). In der Praxis werden wir dann als Folge ein Echo als **rücklaufende Welle b₁** (= 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 a₁**, der von den Abweichungen des Eingangswiderstandes gegenüber Z = 50Ω herröhrt und
- 2) einem zweiten Anteil, der durch **Rückwirkungen des Ausgangs** (bei dem sich ja auch irgendwelche Signale a₂ herumtreiben) **auf den Eingang** erzeugt wird.

Damit lässt sich das Echo so ausdrücken:

$$b_1 = a_1 \cdot S_{11} + a_2 \cdot S_{12}$$

Bedeutung und Messung der verschiedenen Koeffizienten:

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

$$S_{11} = \frac{b_1}{a_1} \quad \text{für } a_2 = \text{Null}$$

Aber das kennen wir doch, denn damit ist **S11 nichts anderes als unser Eingangs - Reflektionsfaktor r aus dem vorigen Rechenbeispiel -- bei diesem Zweitor natürlich für korrekten Abschluss am Ausgang**. 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**“ vom Ausgang zurück zum Eingang. Er liefert uns Informationen über die **Rückwirkungen** im Zweitor, wenn der Eingang nicht angesteuert und zusätzlich korrekt mit $Z = 50\Omega$ abgeschlossen wird. Auf den Ausgang wird in diesem Fall mit einem Signalgenerator die Leistung a_2 „draufblasen“:

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

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 a_2 quält. Mit einem Richtkoppler können wir dann in der Ausgangsleitung folgendes „Echo b_2 “ 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 50Ω abgeschlossen, dann ist a_1 automatisch Null. Also kann ein aus dem „Ausgang des Zweitors herauslaufendes Signal“ nur durch eine Reflexion am nicht perfekten Innenwiderstand entstanden sein. **Damit ist S_{22} nichts anderes als der Ausgangs - Reflektionsfaktor des Zweitores** (natürlich bei korrektem Abschluss des Einganges mit $Z = 50\Omega$)!

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

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 $Z = 50\Omega$ abgeschlossen, dann haben wir die klassische Verstärkerschaltung vor uns. Dann ist nämlich $a_2 = \text{Null}$.

S_{21} ist folglich nichts anderes als die „Spannungsverstärkung des Zweitores bei korrekter Anpassung aus Ausgang“!

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

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!

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

$$Z_{22} = Z \cdot \frac{(1 + S_{22})}{(1 - S_{22})}$$

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“)

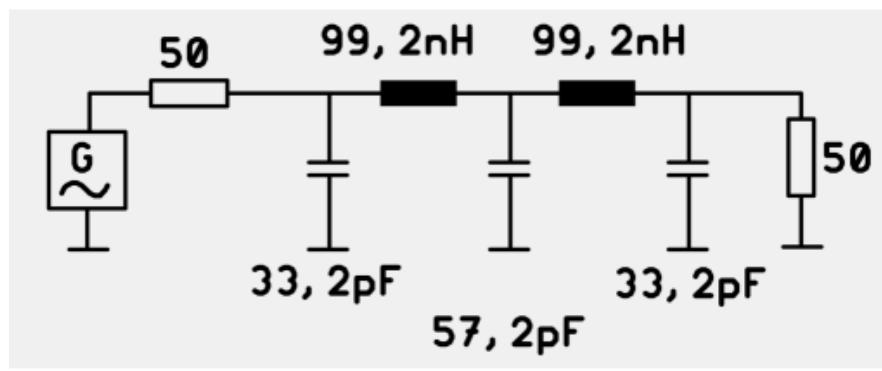
S22 = 0,5 / -20 Grad (Auch der Innenwiderstand des Bausteins weist leider keine exakten 50 Ohm auf. Wer nun die 20 Grad Phasenwinkel einfach zu Null setzt, käme auf 150 Ohm....bitte nachrechnen!)

14.2. 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	$f_g = 110 \text{ MHz}$
Spulenarme Ausführung	
Filtergrad	$n = 5$
Systemwiderstand	$Z = 50\Omega$
„Ripple“ (= Welligkeit der Dämpfung im Durchlassbereich) = 0,1 dB. (Zu diesem Ripple-Wert gehört ein Maximalwert von $S_{11} = -16,4 \text{ dB}$)	



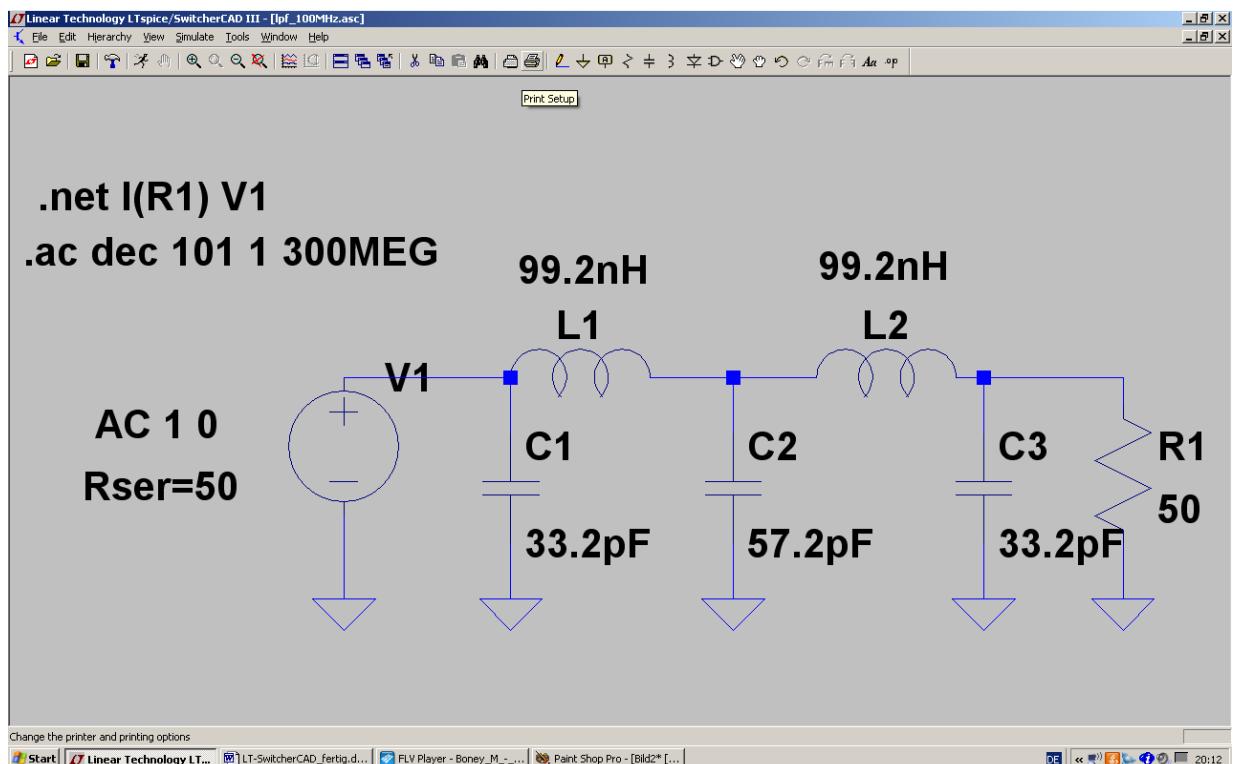
Für die Berechnung solcher Filter gibt es etliche Programme im Internet. Das altbewährte DOS-Programm „fds.zip“ liefert dafür die Bauteilwerte:

$$C_1 = C_3 = 33,2 \text{ pF}$$

$$C_2 = 57,2 \text{ pF}$$

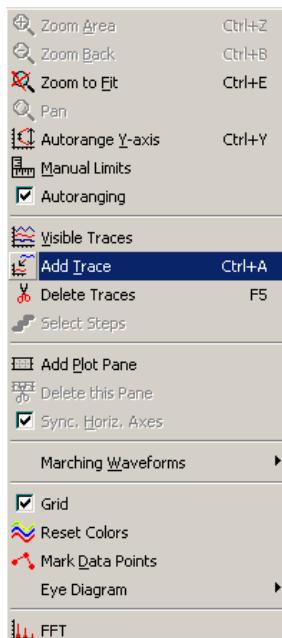
$$L_1 = L_2 = 99,2 \text{ nH}$$

Damit erstellen wir uns folgende Schaltung mit dem Editor des LTSpice/SwitcherCAD-Programms:



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)**

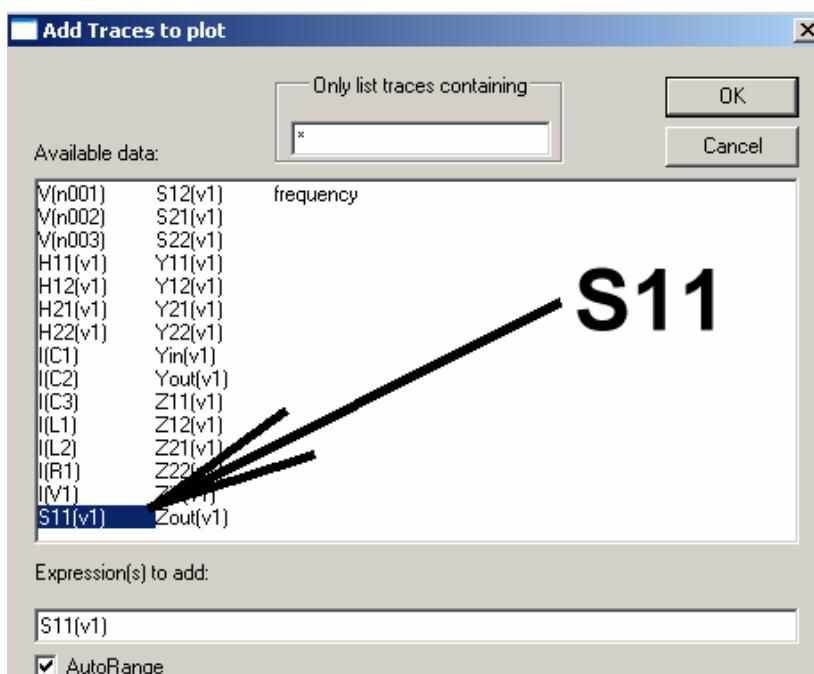


Jetzt wird simuliert und anschließend geht es so weiter:

1. Schritt:

Wir haben zunächst ein noch leeres Ergebnisdiagramm vor uns und klicken deshalb mit der rechten Maustaste darauf.

Im auftauchenden Menu wählen wir „**Add Trace**“



2. Schritt:

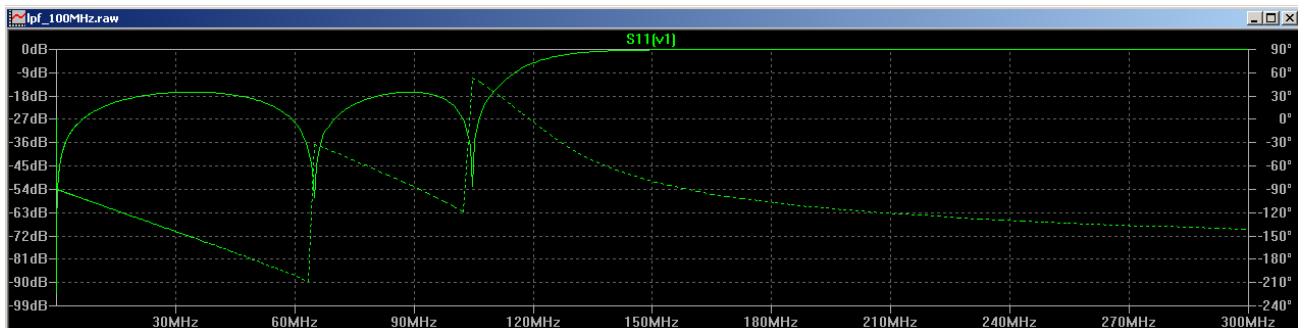
In der Auswahlliste gehen wir auf

„**S11(v1)**“

los und bestätigen mit OK.

(Bitte mal genau hinsehen:
 in der Liste gibt es auch noch die
 Optionen „Y-Parameter“ und „Z-
 Parameter“....)

3. Schritt:

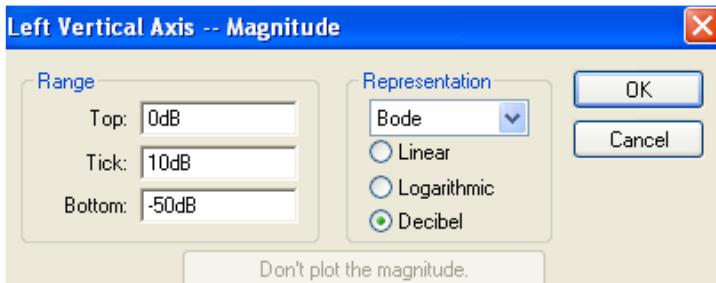


Was da auftaucht, ist ja schon ganz brauchbar, aber noch nicht sehr schön dargestellt. Außer der „Magnitude in dB“ ist nämlich auch noch der Phasenverlauf dargestellt und der interessiert uns im Augenblick nicht besonders. Auch die „dB“-Teilung der linken Achse ist noch verbesserungsfähig.

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.

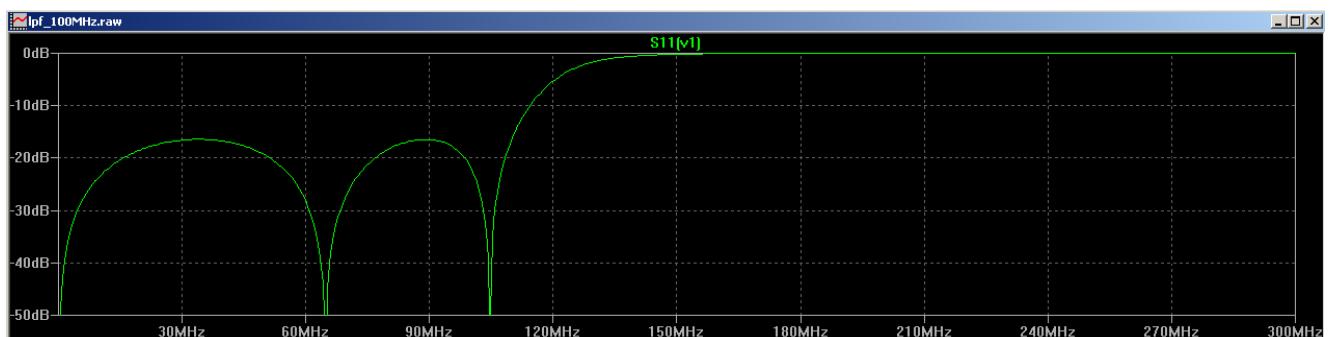
Damit ist die Phasendarstellung verschwunden.



Wenn wir anschließend zusätzlich nochmals mit der rechten Maustaste auf das Diagramm klicken und

GRID

aktivieren, sollte das Ergebnis so aussehen:



4. Schritt:

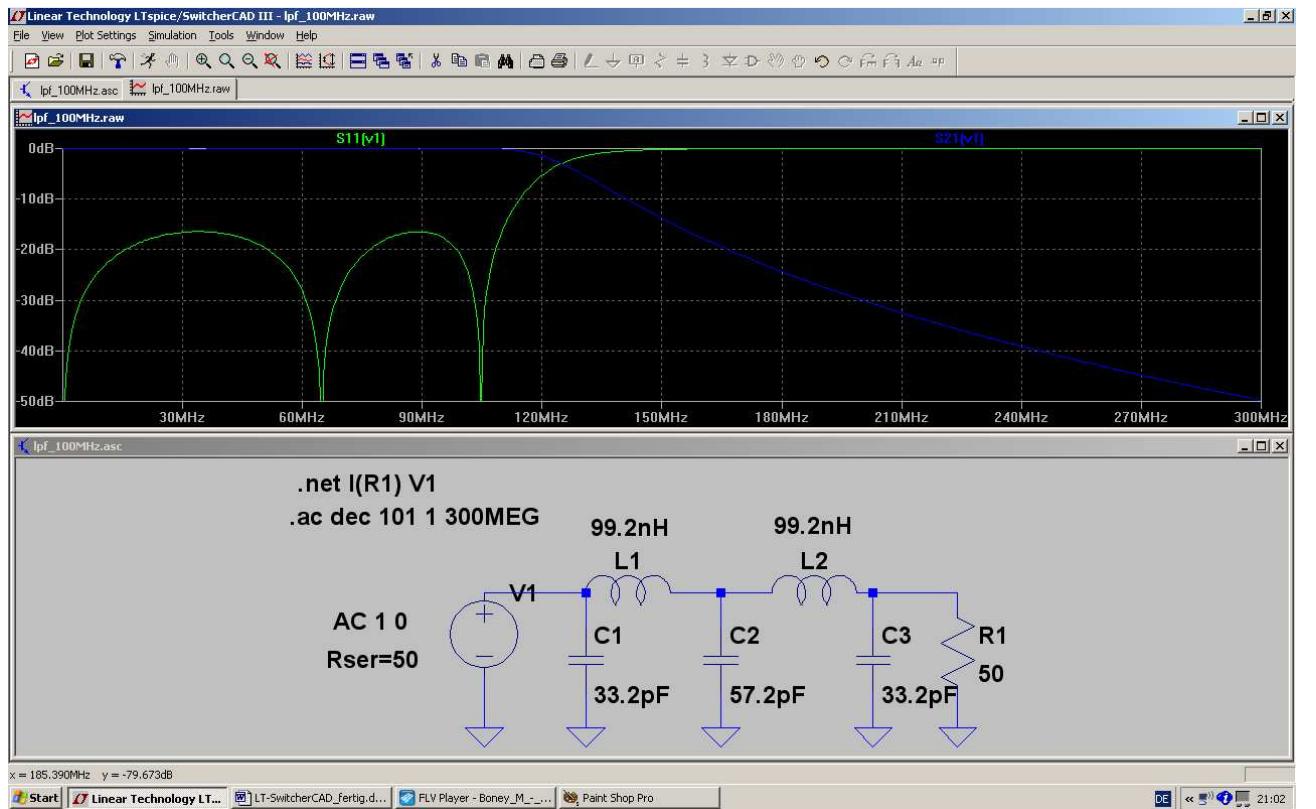
Nun sollte es keine Mühe mehr machen, zusätzlich den Parameter S21 in das Diagramm zu holen.

Also los:

Rechter Mausklick auf das Diagramm / Add Trace / S21 wählen und OK drücken.

Dann sollte das komplette Ergebnis auf dem Bildschirm zu sehen sein. Allerdings muss man nun nochmals das Spiel mit der linken Achsenteilung wiederholen, um den Bereich von 0.....-50dB korrekt darzustellen.

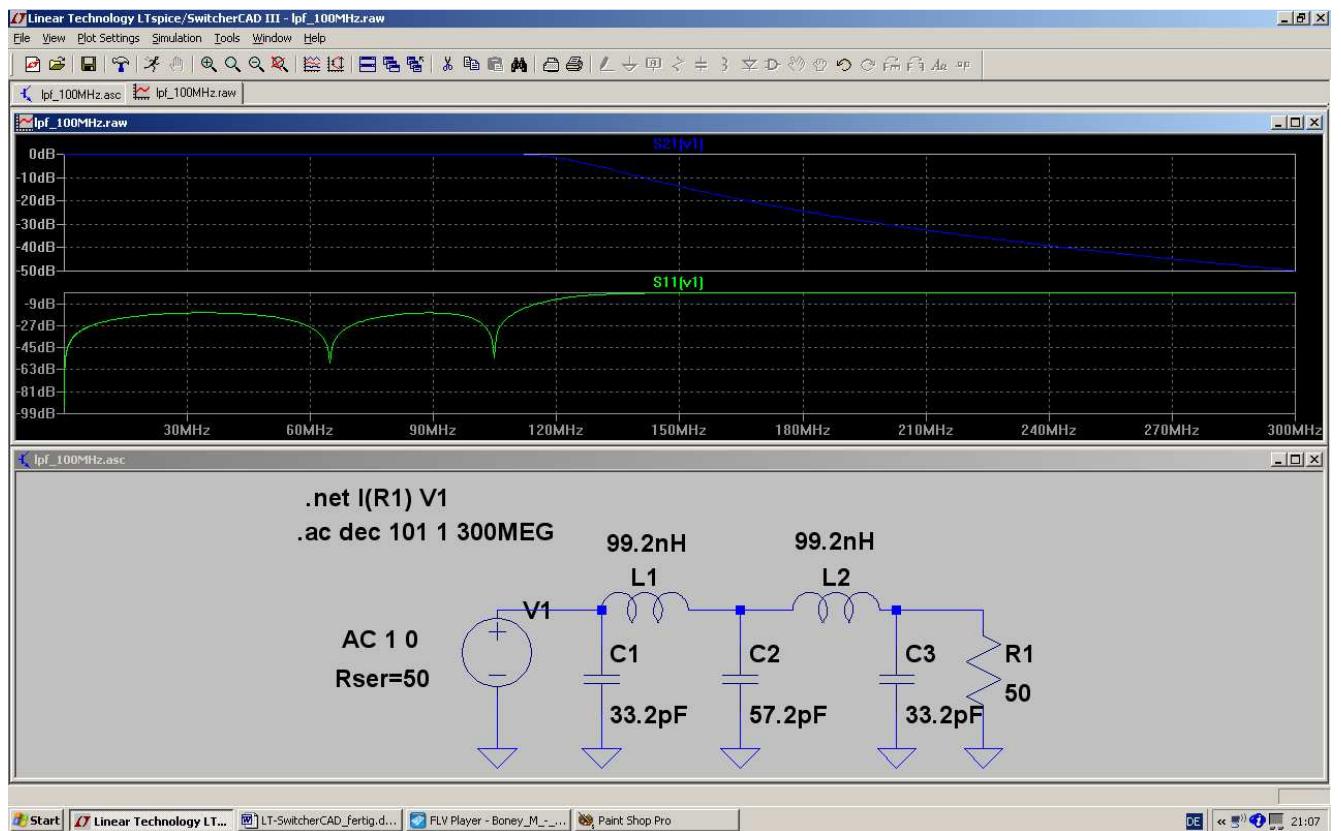
Bitte erst versuchen und hinterher auf der nächsten Seite spicken!



Übrigens:

Wer gerne die beiden Parameter in getrennten Diagrammen darstellen möchte, der klickt wieder rechts auf das
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:



15. Elftes Projekt: Double Balanced Mixer (= Ringmodulator)

15.1. 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:

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

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:

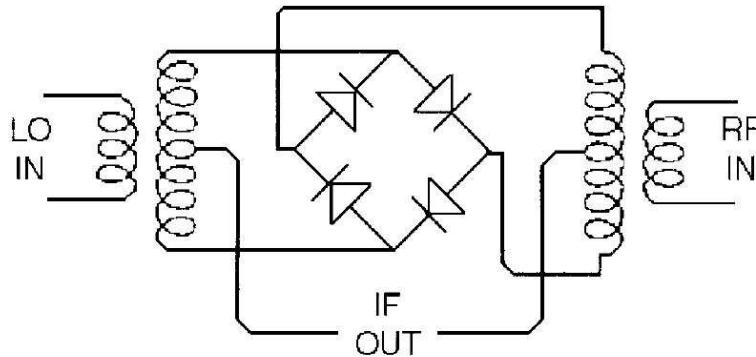
Das Rechtecksignal besteht bekanntlich selbst wieder aus einer Summe von sinusförmigen Signalen. Neben der „Grundwelle“ findet man die „Oberwellen“, und die haben die dreifache, fünffache, siebenfache, neunfache.....Grundfrequenz, wobei die Amplituden entsprechend dem „Grad“ der Oberwelle abnehmen. Der 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 multipliziert und erzeugt deshalb am Ausgang ebenfalls ein „Signal-Pärchen“ mit der zugehörigen Summen- und Differenzfrequenz.

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

15.2. 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.



Sie besteht aus einem Ring von 4 Dioden (bitte aufpassen: sie sind **NICHT** so gepolt wie beim Brückengleichrichter!) sowie zwei Übertragnern. Jeder Übertrager ist mit drei gleichen Wicklungen ausgestattet. Die beiden Sekundärwicklungen werden dabei in Reihe geschaltet und die „Mittenanzapfung“ herausgeführt.

Figure 2. A popular diode-ring mixer topology.

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. 50Ω 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 Bauklotzchen benutzt 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 „Schalteistung“ 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).

15.3. 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 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.

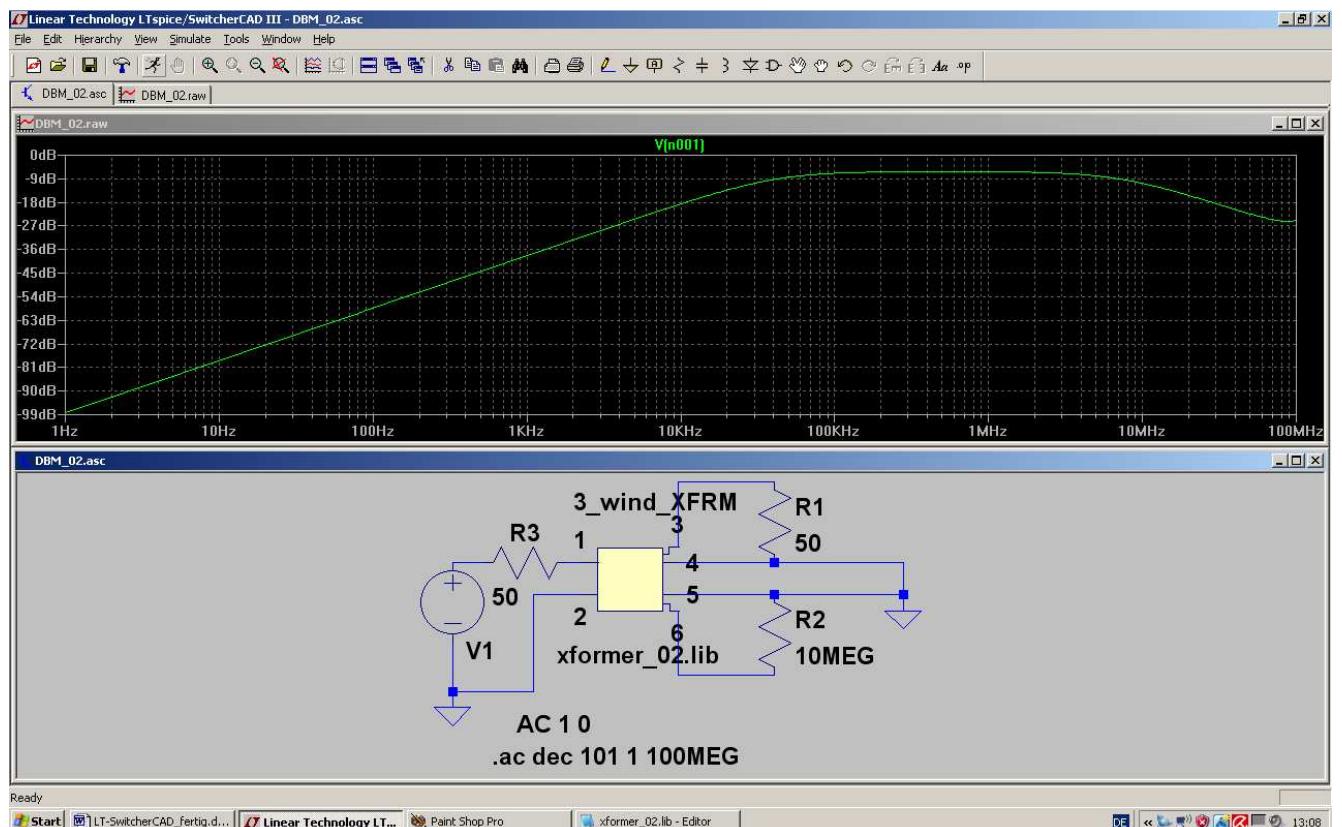
Beispiel:

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 $10\text{M}\Omega$.).

Das führt zu folgender Simulation der Spannung an R1, bei der

jede einzelne Wicklung eine Induktivität von $100\mu\text{H}$ aufweist (bei einem Wicklungswiderstand von 1Ω und einer Wicklungskapazität von 1pF).



Sehr schön ist nun der eingeschränkte Übertragungsbereich zu erkennen.

15.4. 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.

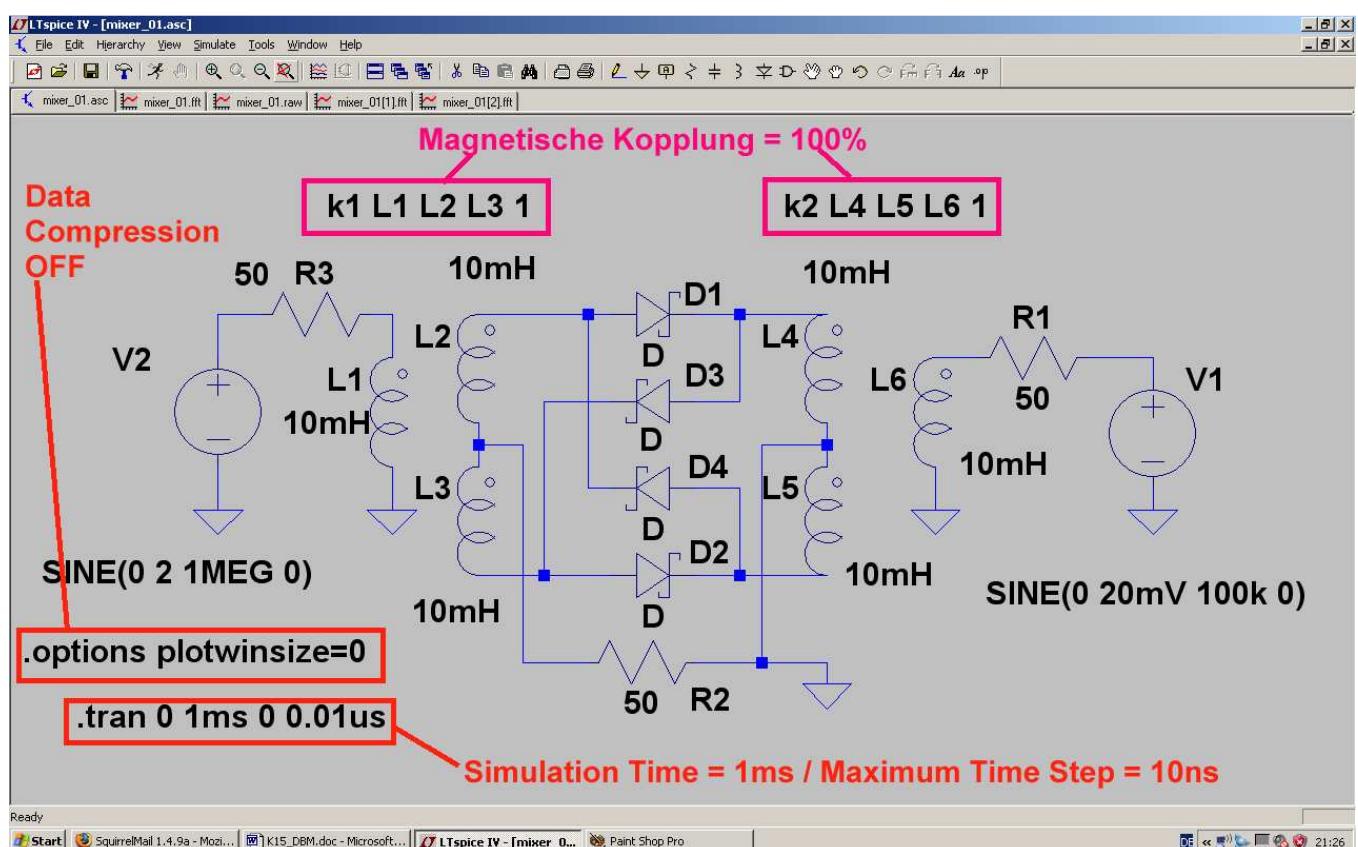
Beide idealen Übertrager bestehen (Siehe Kapitel 6.2) aus jeweils drei Einzelinduktivitäten vom Typ „ind2“, die zu 100% magnetisch gekoppelt sind. Jede Wicklung weist eine Induktivität von 10mH auf.

Die Schottky-Dioden tragen die Bezeichnung „schottky“ und finden sich in der LTSpice-Bauteilbibliothek.

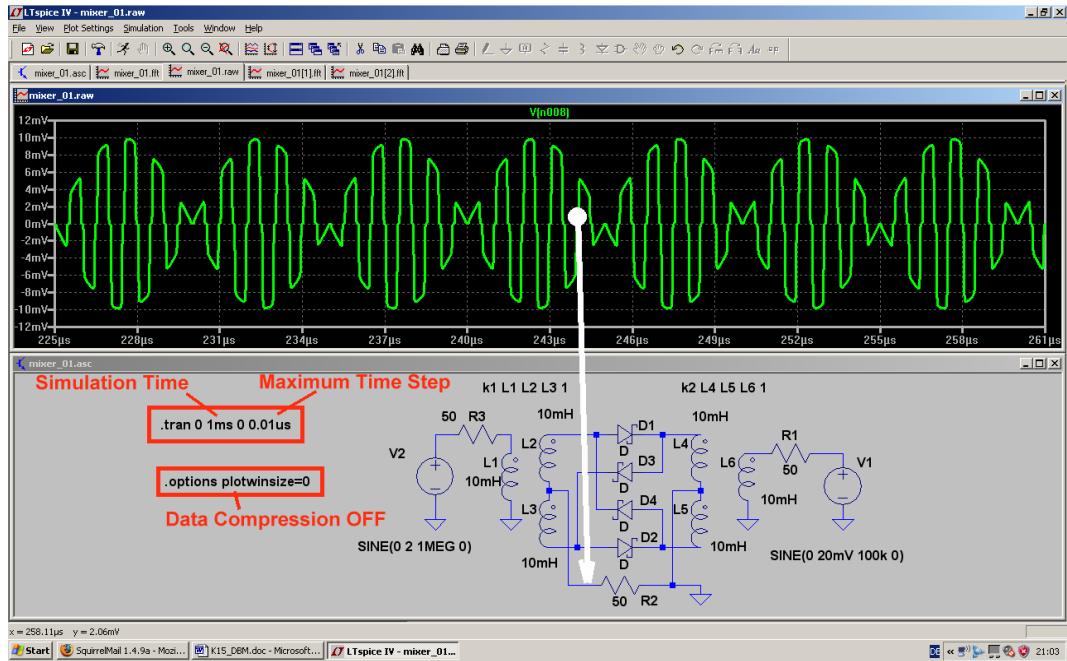
Eine Simulationszeit von 1ms ergibt eine Frequenzauflösung von $1/1\text{ms} = 1000\text{Hz}$. Fordert man mindestens 100 000 echte Samples, dann muss ein Maximum Time Step von $1\text{ms}/100000 = 10\text{ns}$ gewählt werden. Dies gehört zu einer Samplefrequenz von $1/10\text{ns} = 100\text{MHz}$ und so braucht man bis zu einer Signalfrequenz von 50MHz keine Aliasing-Effekte zu befürchten.

Die Datenkompression wird durch die Direktive „**,,options 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:

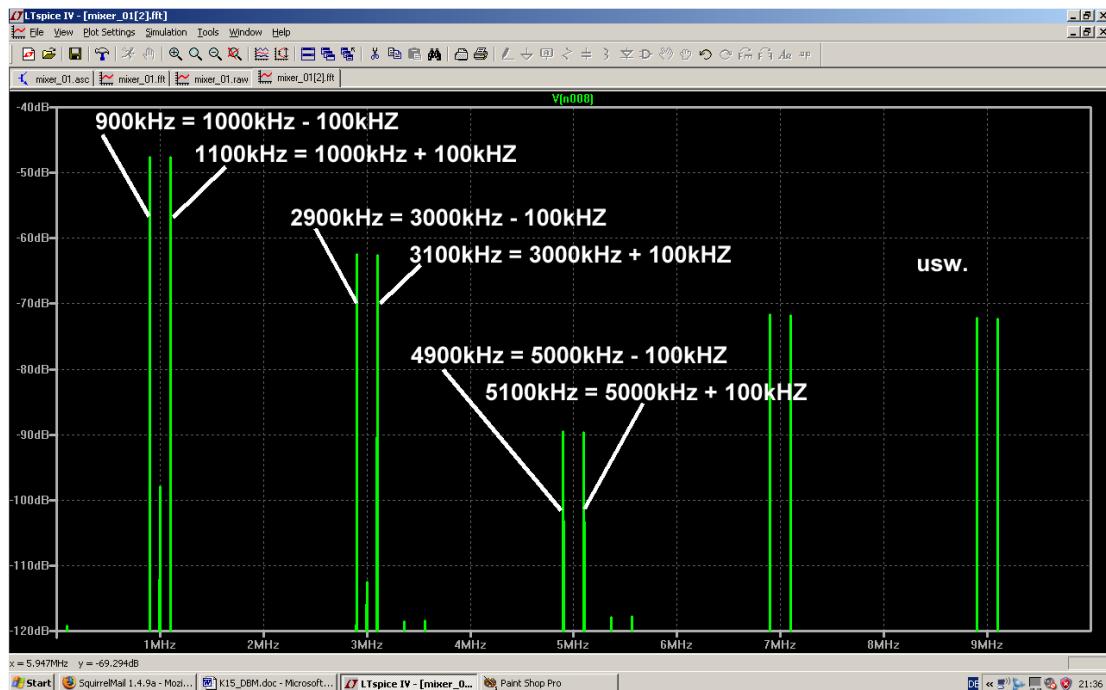


Man erkennt sehr gut die dauernde Umpolung des RF-Signals durch das LO-Signal. Ebenso sieht man, dass wegen der Schwellspannung der Schottky-Dioden (= ca. 0,4v) bei jedem Nulldurchgang ein kurzer „Aussetzer“ zu beobachten ist (= unterhalb von einer LO-Signalamplitude von etwa 2 x 0,25V sperren die Dioden 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 R2.

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-Singals, 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...

16.Zwölftes Projekt: Digitale Schaltungssimulation

16. 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: $R = 10\text{k}\Omega$) 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 = +1 Volt (...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

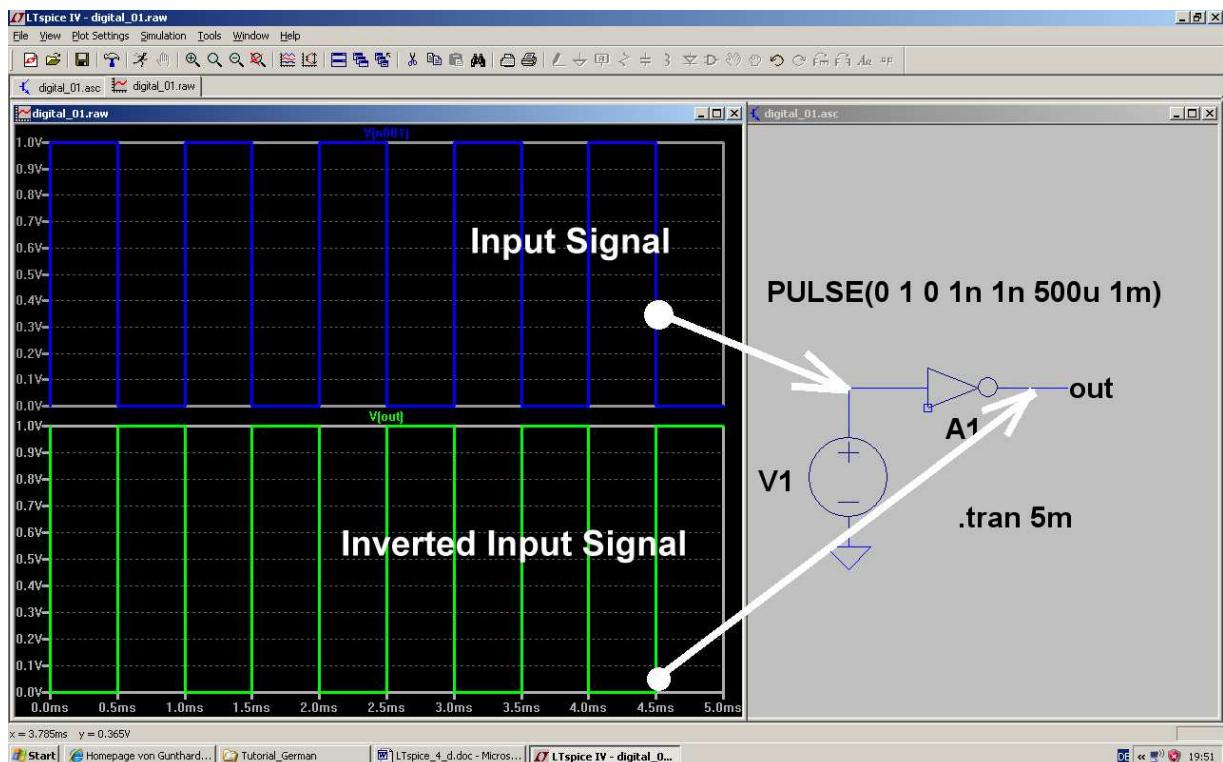
wählen. Dadurch taucht ein weiteres Diagramm auf. Falls nötig, wiederholt man diese Prozedur solange, bis man genügend Diagramme für die Darstellung aller gewünschten Signale zur Verfügung hat. Wer das nicht tut, wird ganz schnell wahnsinnig....

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.

16.2. 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 ($U_{min} = 0V$ / $U_{max} = +1V$ / 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:

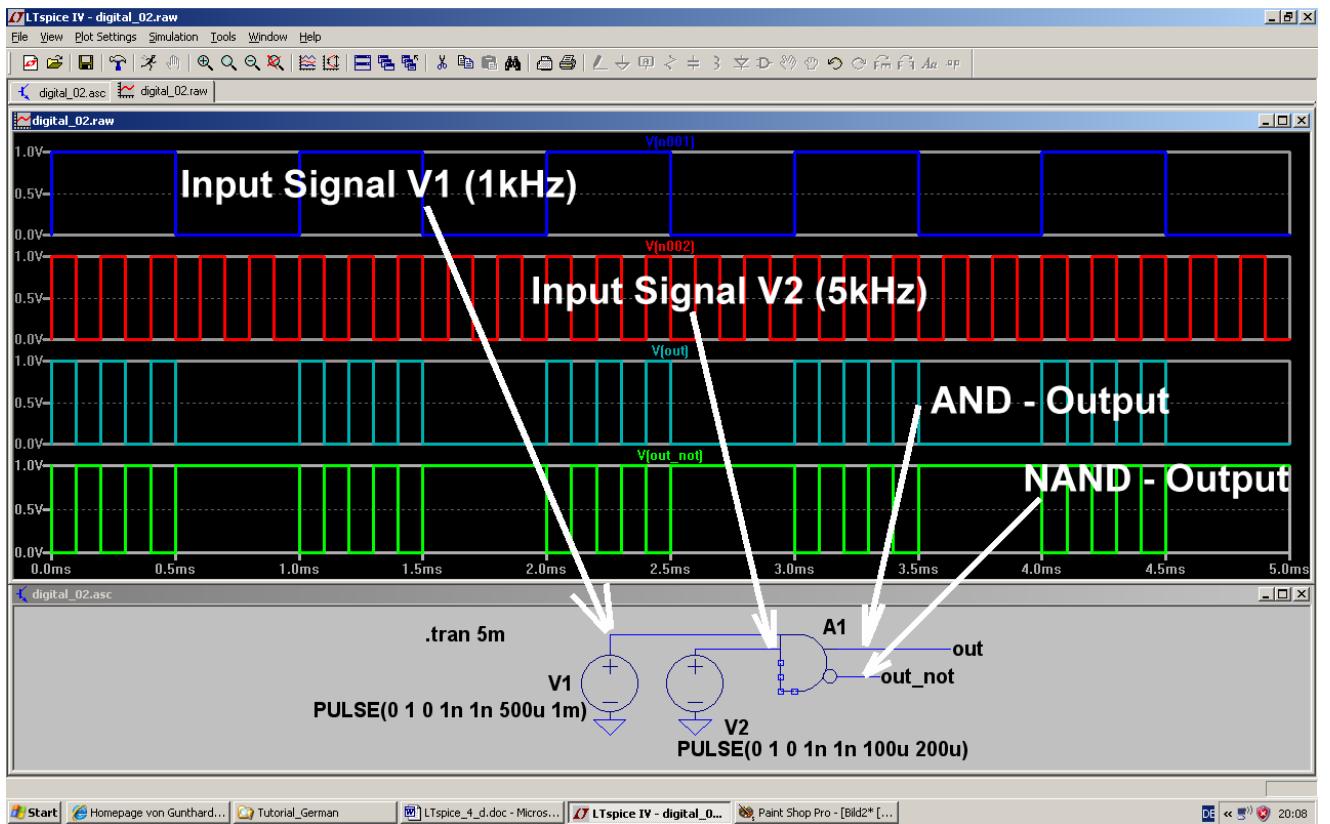


16.3. Der AND-Baustein

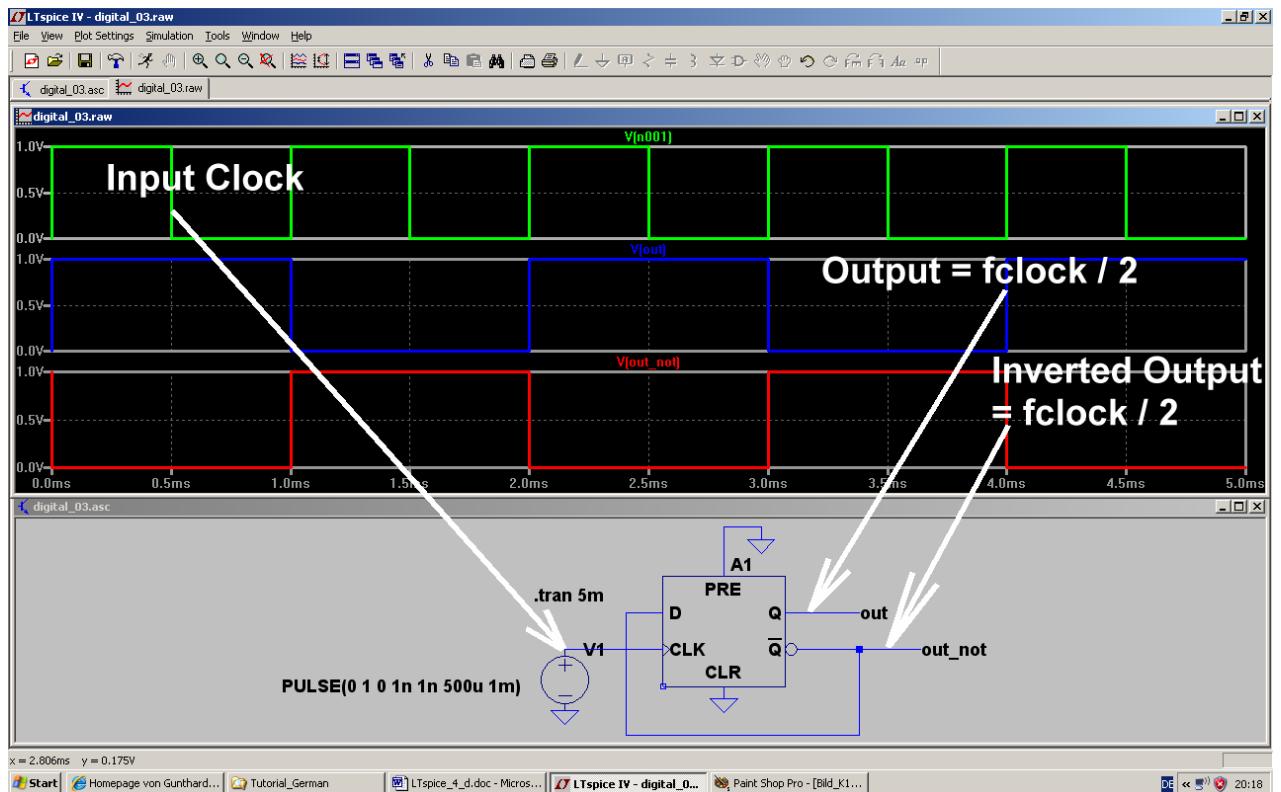
Wir wollen eine UND-Verknüpfung mit 2 Eingängen realisieren. Am einen Eingang liegt ein Digitalsignal mit $f = 1\text{kHz}$, am anderen dagegen eines mit $f = 5\text{kHz}$. Simuliert wird von 0....5ms.

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:



16.4. Das D-Flipflop



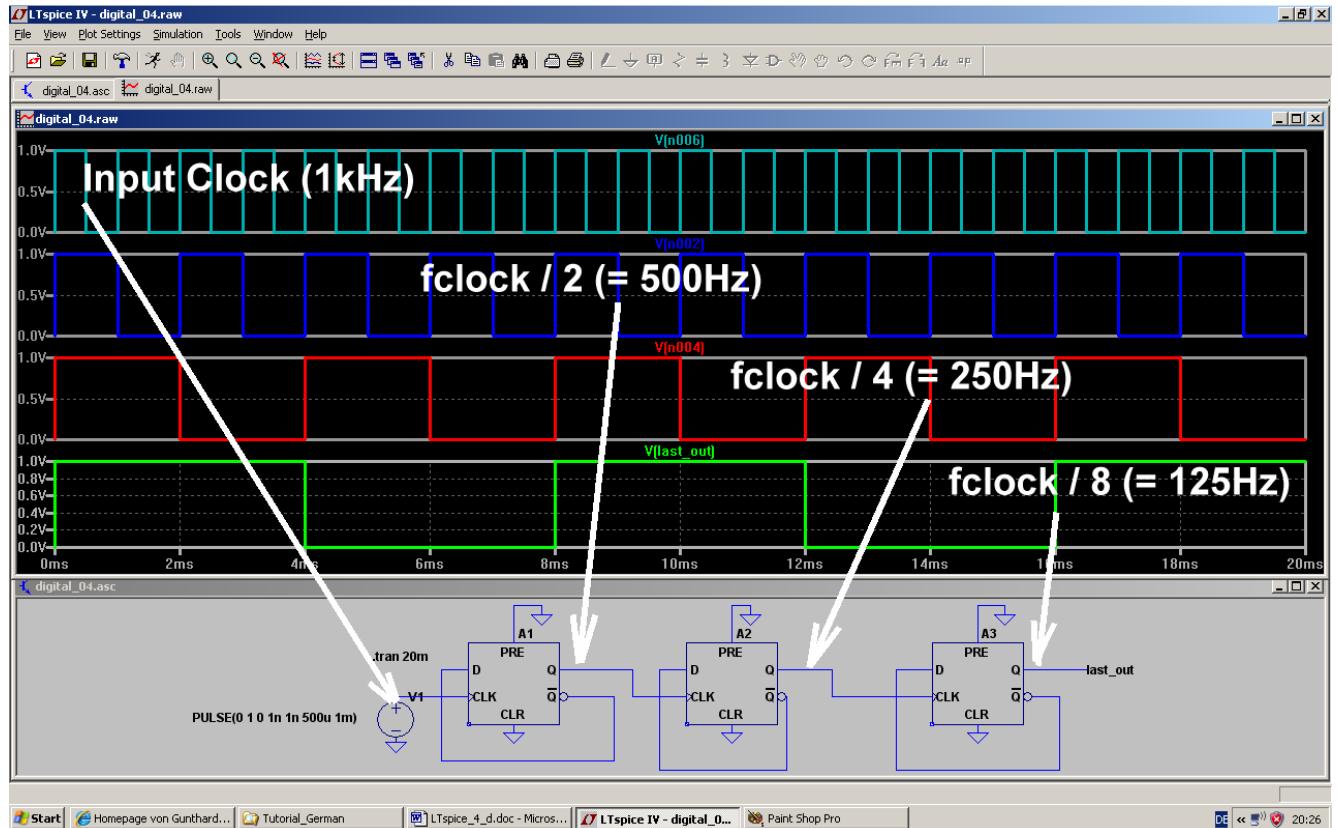
Die Schaltung stellt den bekannten Binär-Frequenzteiler dar, bei dem am Ausgang die halbe Eingangsfrequenz beobachtet werden kann. Dazu wird der invertierte Q-Ausgang mit dem D-Eingang verbunden. Die Input Clock-Frequenz beträgt 1 kHz.

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

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

16.5. 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 $f = 1\text{kHz}$.

Wer dieses Kapitel als etwas dürfzig 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...

17. Dreizehntes Projekt: Rausch-Simulation

17.1. Grundlagen

Hinweis:

Die folgenden 3 Kapitel sind der Einführung in meinen zweiteiligen Artikel in der Zeitschrift „UKW-Berichte“ entnommen. Er lautet:

Praxisprojekt: Rauschzahlmessung mit älteren Spektrum-Analyzern

und ist in den Heften 1 und 2 / 2008 erschienen.

17.1.1. „Rauschen“ -- woher kommt das?

Das lässt sich schnell und präzise beantworten: in jedem elektrischen Widerstand bewegen sich Elektronen, wenn Strom fließt. Sobald aber Wärme mit im Spiel ist (....und das ist automatisch oberhalb des absoluten Nullpunktes der Fall...) , werden diese Teilchen immer unruhiger und nehmen nicht mehr den geraden Weg von Minus nach Plus. Sie stoßen zusammen, prallen zurück, werden nach vorne geschleudert oder weichen zur Seite aus.... der Strom schwankt also durch den Wärme-Einfluss dauernd und völlig unregelmäßig um kleinere und größere, aber winzige Beträge.

Dieser Effekt trägt den Namen „**Thermal Noise**“ (= Thermisches Rauschen). Selbst wenn **gar keine äußere Spannung angelegt ist**, merken wir diese Eigenbewegungen der Ladungsträger und messen an den Anschlüssen des Bauteils eine kleine „Rausch-Leerlaufspannung U_{NOISE} “. Sie lässt sich folgendermaßen berechnen:

$$U_{NOISE} = \sqrt{\frac{4hfBR}{\frac{hf}{e^{kT}} - 1}} \quad (1)$$

mit

h = Planck'sches Wirkungsquantum

k = Boltzmann – Konstante = $1,38 \times 10^{-23}$ 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:

$$U_{NOISE} = \sqrt{4kTB} \quad (2)$$

Stellt man diese Formel etwas um, dann sieht man plötzlich sofort, was los ist:

$$\frac{\left(\frac{U_{NOISE}}{2}\right)^2}{R} = kTB \quad (3)$$

- Das ist schlicht und einfach eine Leistungsangabe! Folglich kann jeder Ohmsche Widerstand -- unabhängig von seinem Widerstandswert! -- durch den Einfluss der Wärme die Rausch-LEISTUNG (= „Available Noise Power“) kTB an seinen Anschlüssen abgegeben.
- Sobald man nun diesen Widerstand als Spannungsquelle mit Uberspannung U_{NOISE} und (rauschfreiem) Innenwiderstand R auffasst, muss man an diese Quelle einen (rauschfreien) Lastwiderstand mit dem gleichen Wert R anschließen, um Leistungsanpassung zu erhalten. Am Lastwiderstand liegt dann die halbe Uberspannung aus Formel (2) und der Lastwiderstand erhält diese „verfügbare Rauschleistung kTB “ (= Available Noise Power).
- Diese 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). Die Spektrale Rauschleistungsdichte (= Leistung pro Hertz Bandbreite) ist normalerweise unabhängig von der Frequenz, in diesem Fall spricht man von „Weißem Rauschen“.

Ganz wichtig ist nun folgende Sache:

Fast immer denkt man bei Empfängern und Systemen in „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

P₀ = 1 Milliwatt am Systemwiderstand

gesetzt. Das ergibt den

$$\text{Leistungs-Pegel} = 10 \cdot \log\left(\frac{\text{Leistungswert}}{1\text{mW}}\right) \quad \text{in dBm} \quad (4)$$

Betrachtet man nun die obige Sache mit der Rauschleistung „kTB“ etwas genauer, dann lässt sich eine interessante Vereinfachung einführen:

$$kTB = (kT) \cdot B = (\text{Rauschleistungsdichte}) \cdot (\text{Bandbreite})$$

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 bei Raumtemperatur (T₀ = 290K) einen verfügbaren Rauschpegel und damit eine verfügbare Rauschleistungsdichte von

-174dBm pro Hz Bandbreite

Bei größeren Bandbreiten als 1Hz ist der Rest dann ganz einfach:

Gesamt verfügbarer Rauschpegel in dBm = -174dBm + 10log(Bandbreite in Hz)

Nehmen wir dazu eine kleine Aufgabe:

„Wie groß ist „im Leerlauf“ die Rauschspannung an den Anschlussklemmen eines Widerandes von 50 Ohm, wenn mit einer Bandbreite von 100kHz gearbeitet wird?“

Lösung:

Rauschpegel bei Anpassung = -174dBm + 10log(100000) = -174dBm + 50dB = -124dBm

Dazu gehört eine verfügbare Rauschleistung von

$$P = 1\text{mW} \cdot 10^{\frac{-124}{10}} = 1\text{mW} \cdot 10^{-12,4} = 4 \cdot 10^{-16}\text{W}$$

Das würde an einem ABSCHLUSS-Widerstand von 50 Ohm folgende Spannung ergeben:

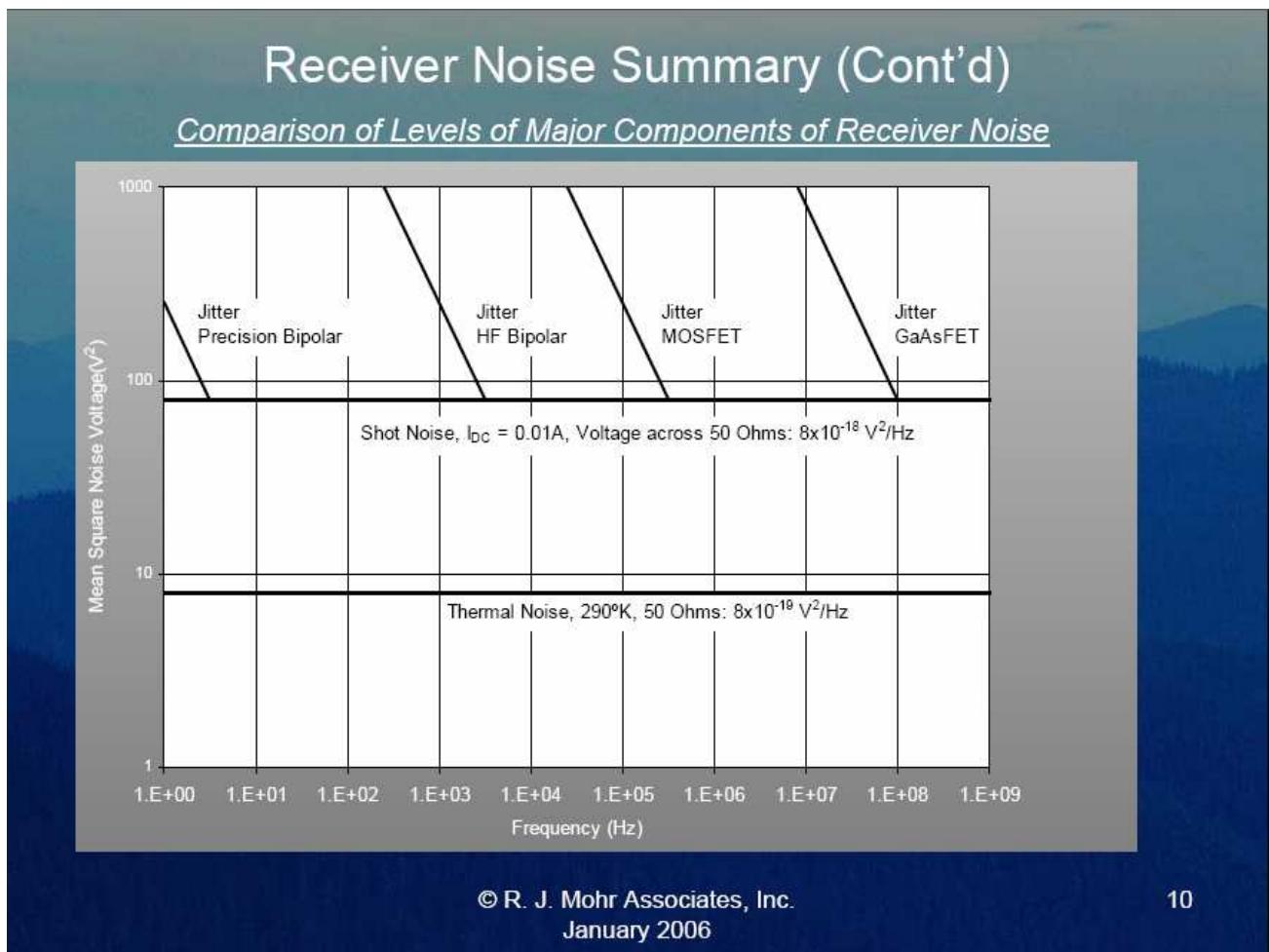
$$U_{\text{NOISE}} = \sqrt{P \cdot R} = \sqrt{50\Omega \cdot 4 \cdot 10^{-16}\text{W}} = 141\text{nV}$$

Wegen des fehlenden Abschlusses ist die Leerlaufspannung doppelt so groß, also 282nV.

17.1.2. Weitere Rauschquellen

In jedem aktiven Bauteil (Röhre, Bipolar-Transistor, Sperrschiicht-FET, MOSFET, HEMT usw.) haben wir es außer mit dem thermischen Rauschen noch mit zwei zusätzlichen Rauscharten zu tun:

- Shot-Noise** (= Schrot-Rauschen) tritt bei Vakuumdiode und P-N-Übergängen durch Ungleichförmigkeiten des Stromflusses beim Durchqueren der Potential-Unterschiede auf. Es handelt sich hierbei um breitbandiges, weißes Rauschen.
- 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 Gasentladungsrohren als Rauschquellen, die ein sehr breitbandiges „Plasma-Noise“ erzeugten.

Sehen wir uns nun an, wie diese verschiedenen Rauscharten in einer Schaltung berücksichtigt und in einer einzigen Größe zusammengefasst werden können.

17.1.3. Rauschtemperatur und Noise Figure eines Twoports

Sobald wir verschiedene Bausteine (= Twoports = Zweitore) eines Kommunikationssystems mit demselben Systemwiderstand (normalerweise $= 50\Omega$, aber in der Radio- / Fernseh- / Videotechnik $= 75\Omega$) in Reihe schalten, interessieren uns automatisch die Qualitätsforderungen zur korrekten Übertragung des Nutzsignal. Sie werden oft durch den SINAD-Wert (= Signal to Noise and Distortion) ausgedrückt. Nach oben wird dabei der Signalpegel durch Übersteuerungsgefahr, Verzerrungen, Begrenzung und Intermodulation limitiert, nach unten ist es hauptsächlich das Rauschen (aber auch andere Störungen, wie z. B. das Übersprechen aus anderen Kanälen), das den Signal-Rauschabstand bestimmt und so den Mindest-Signalpegel im System vorgibt. Deshalb muss jeder Baustein neben seinen S-Parametern auch durch sein Rauschverhalten spezifiziert werden.

Beim Rauschen sind folgende Angaben möglich und üblich (...leider sind sich die verschiedenen Autoren manchmal nicht einig, welcher Name bzw. Buchstabe nun gerade der richtige ist....):

- a) Die „**äquivalente Rauschtemperatur T_e** “ drückt das Eigenrauschen des Bausteins durch eine „zusätzliche Übertemperatur“ des ansteuernden Quellwiderstandes aus. Der Baustein selbst wird dabei als rauschfrei gedacht.
 - b) Die „**NOISE FIGURE (NF)**“ gibt an, um wie viel dB sich der **Signal-Rauschabstand beim Durchlaufen des Bausteins oder Systems verschlechtert** (...Die ansteuernde Quelle liefert dabei nicht nur das gewünschte Nutzsignal, sondern auch ein Rauschen, das von ihrem Innenwiderstand stammt).
 - c) Der „**NOISE FACTOR (F)**“ ist dagegen die NOISE FIGURE nach b), aber nicht in dB, sondern als schlichtes Leistungs-Verhältnis ausgedrückt.
-

17.2. Simulation der Spektralen Rauschleistungs-Dichte

Das klingt sehr kompliziert, ist es aber nicht. Wenn wir uns nämlich an das Einführungskapitel erinnern, dann stand da die Sache mit der identischen Rauschleistung, die jeder Widerstand (...unabhängig von seinem eigenen Widerstandswert) bei Leistungsanpassung abgeben kann:

$$kTB = (kT) \cdot B = (\text{Rauschleistungsdichte}) \cdot (\text{Bandbreite})$$

Die Rauschleistungsdichte in dieser Formel ist nun **diejenige Leistung, die in 1Hz Bandbreite** zu finden ist. Die gesamte Rauschleistung erhält man dann durch Multiplikation dieses Wertes mit der Bandbreite des Systems (...oder durch Integration über die Bandbreite, falls sich die Rauschleistungsdichte mit der Arbeitsfrequenz ändert....).

Oft interessiert man sich jedoch für die zugehörige -- und leichter messbare! -- Spannung, die zu dieser Rauschleistung gehört. Aus der gemessenen oder simulierten Spannung kann dann leicht über die bekannte Formel

$$P = \frac{U^2}{R}$$

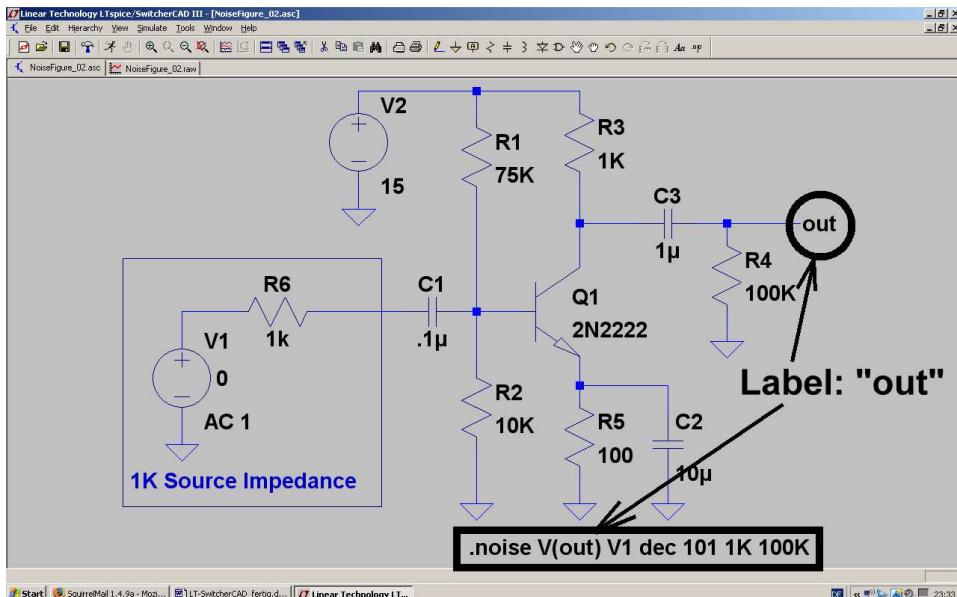
mithilfe des zugehörigen Systemwiderstandes die Leistung ermittelt werden (und umgekehrt). Deshalb zieht man oft die **Wurzel aus der Spektralen Rauschleistungsdichte** und bekommt dann einen Wert für die „Spektrale Rauschspannungsdichte“

Diese Größe wird dann üblicherweise mit der Einheit

$$\frac{\text{Nanovolt}}{\sqrt{1\text{Hz}}}$$

angegeben und ergibt nach Multiplikation mit dem Faktor $\sqrt{\text{Messbandbreite}}$ (bzw. nach korrekter Integration) die **messbare Rausch-Gesamtspannung**.

Sehen wir uns das mal praktisch an dem bei der Software mitgelieferten Beispiel „NoiseFigure.asc“ an. Es handelt sich hier um einen einstufigen Transistorverstärker für den Audiobereich. So sieht das Schaltbild aus, nachdem wir es etwas verändert haben:



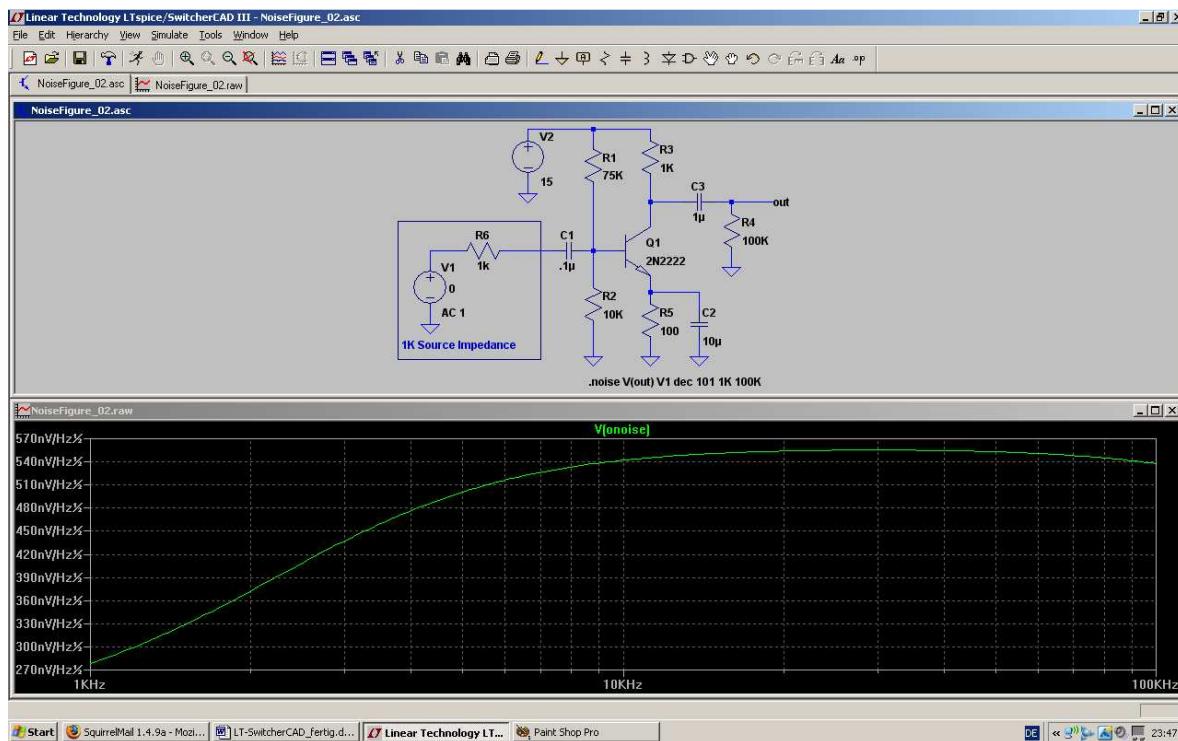
Wir erkennen:

Am Ausgang muss ein Label (am besten: „out“) eingefügt werden, auf den sich die Rauschsimulation beziehen kann.

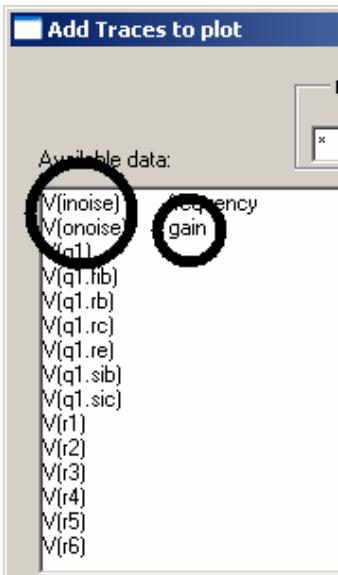
Das erforderliche Simulation Command für die Rauschsimulation wird über den Pfad „Simulation / Edit Simulation Command / Noise“ erzeugt und sagt folgendes:

„Führe eine Rauschsimulation durch und verwende dabei die Spannung V(out) am Label „out“ als Ausgangssignal. Das Rausch-Eingangssignal liefert die Spannungsquelle V1 mit ihrem Innenwiderstand R6 = 1k. Benütze einen dekadischen Sweep von 1kHz bis 100kHz mit 101 Punkten pro Dekade.“

Hier das Ergebnis für $V(\text{onoise})$ = Output Noise Voltage Density nach dem „Antippen“ des Ausgangspins mit dem Cursor:



Bitte genau die Kalibrierung der vertikalen Achse in der vorhin besprochenen Einheit beachten.....



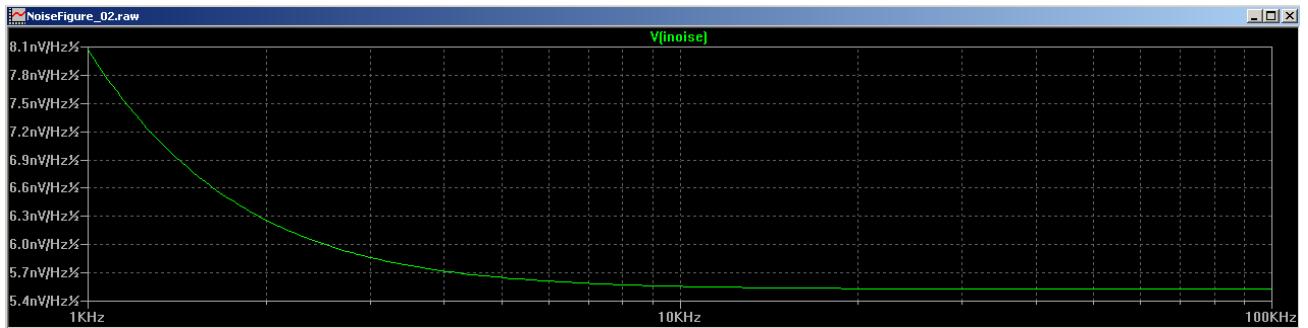
Es macht nun keinerlei Mühe, ein zusätzliches neues Diagramm aufzumachen und die auf den Eingang umgerechnete

Rauschspannung $V(\text{inoise})$ = Input Noise Voltage

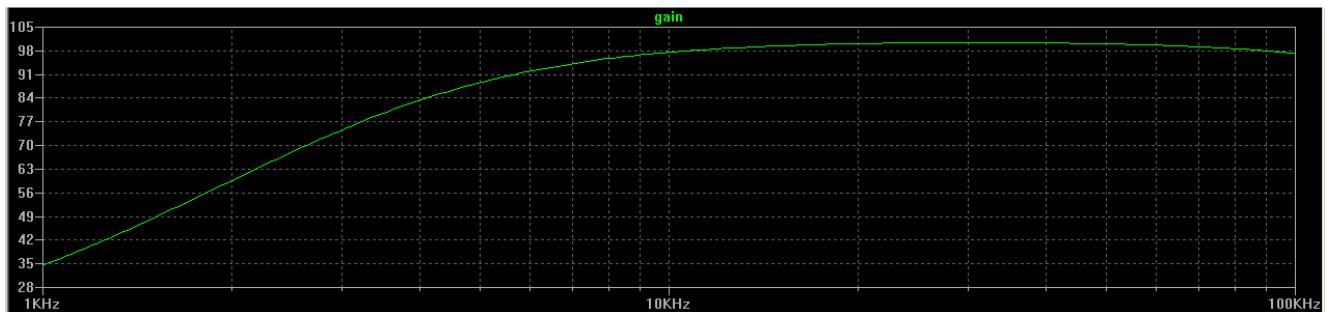
darstellen zu lassen. Diese Spannung ist als zusätzliche Signalquelle in Reihe zur (rauscharmen) V_1 zu denken. Dazu klickt man rechts auf das Diagramm, geht in „**Add Trace**“ und wählt aus der nebenstehenden Liste „ **$V(\text{inoise})$** “.

Wie man sieht, steht in dieser Auswahl auch die bereits angezeigte Ausgangsspannung **$V(\text{onoise})$** . Aber richtig interessant ist die Möglichkeit, sich die Verstärkung (= **gain**) ebenfalls zeigen zu lassen. Sie wurde vom Simulationsprogramm sowieso berechnet und archiviert, da man von der Ausgangsspannung zur Eingangsspannung kommt, wenn man diese durch die Verstärkung dividiert.

So sieht dann die **Eingangsräuschanwendung $V(\text{inoise})$** aus.....



...und so die **Verstärkung („gain“)**:



17.3. Simulation der Noise Figure in dB

Bei Kommunikationssystemen sind meist viele Bausteine oder Baugruppen hintereinandergeschaltet. Deshalb interessiert stets die Qualitätsveränderung der Übertragung beim Durchlaufen eines solchen „Twoports“ mit einem bestimmten Eigenrauschen. Sie wird durch die

Noise Figure NF (= Rauschzahl) in dB

ausgedrückt und so definiert:

Die „**NOISE FIGURE (NF)**“ gibt an, um wie viel dB sich der (ebenfalls in dB angegebene) **Signal-Rauschabstand beim Durchlaufen des Bausteins oder Systems verschlechtert**.

Achtung:

Die ansteuernde Quelle liefert dabei nicht nur das gewünschte Nutzsignal, sondern auch ein Rauschen, das nur von ihrem Innenwiderstand stammt.

Rechnet man dagegen den dB-Wert dieser Noise Figure in ein lineares Verhältnis um, dann haben wir den „Noise Factor F“ vor uns.

Bei unserer SPICE-Simulation müssen wir deshalb folgendermaßen bei der Bestimmung von NF vorgehen

- a) Um auf „dB“ zu kommen, müssen wir mit der Beziehung

$$NF = 10 \times \log(\dots)$$

arbeiten.

- b) Innerhalb der Klammer steht dann das Verhältnis der (vom Ausgang auf den Eingang zurückgerechneten) Rauschleistung zu der vom Innenwiderstand der Quelle in die Schaltung eingespeisten Rauschleistung.
Wären beide Leistungen gleich groß, dann hätten wir einen rauschfreien Baustein vor uns und die Noise Figure NF wäre

$$NF = 10 \times \log(1) = \text{Null dB}$$

Also gibt es in der Praxis nur NF-Werte größer als Null.....

Das Eigenrauschen eines Widerstandes steigt linear mit der absoluten Temperatur an und man hat sich früher stillschweigend darauf geeinigt, von 290K (= 17 Grad Celsius) als üblicher Rauschtemperatur auszugehen. Im Zuge des Klimawandels findet man allerdings jetzt meist Berechnungen, die eine absolute Temperatur von 300K (= 27 Grad Celsius) beim Bauteil voraussetzen.

Und so wird das Verhältnis von der auf den Eingang zurückgerechneten Ausgangsrauschleistung P_{noise} zur eingespeisten reinen Quellwiderstands-Rauschleistung P_{source} errechnet:

$$\frac{P_{noise}}{P_{source}} = \frac{\left(\frac{V_{noise}^2}{4 \cdot R_{source}} \right)}{k \cdot T \cdot B} = \frac{V_{noise}^2}{4 \cdot k \cdot T \cdot B \cdot R_{source}}$$

Darin ist „k“ die Boltzmann-Konstante, „T“ die absolute Temperatur in Kelvin, „B“ die Messbandbreite und R_{source} der Quellwiderstand.

Denken wir nun noch daran, dass die auf den Eingang zurückgerechnete Rauschleistung schon automatisch für eine Bandbreite von 1Hz gilt, dann können wir die Bandbreite „B“ im Nenner auch noch ignorieren und weglassen.

Damit kommen wir zur folgenden endgültigen Formel, die wir in ein SPICE-Kommando umsetzen müssen:

$$NF = 10 \cdot \log \left(\frac{V_{\text{noise}}^2}{4 \cdot k \cdot T \cdot R_{\text{Source}}} \right) \text{ in dB}$$

Um wenn wir nun mit dem Klimawandel gehen und eine Temperatur von 300K ansetzen, wird daraus die endgültige Formel

$$NF = 10 \cdot \log \left(\frac{V_{\text{noise}}^2}{4 \cdot k \cdot 300 \cdot R_{\text{Source}}} \right) \text{ in dB}$$

Jetzt kommt ein Problem:

Bei der SPICE-Simulation mit LTSwitcherCAD können wir diese Berechnung mit nachfolgender Darstellung in einem Diagramm nur über eine Funktion in einer PLOT-Anweisung erzielen!

Das dafür erforderliche korrekte SPICE-Command zur Berechnung von „NF(1k)“ in dB lautet dann

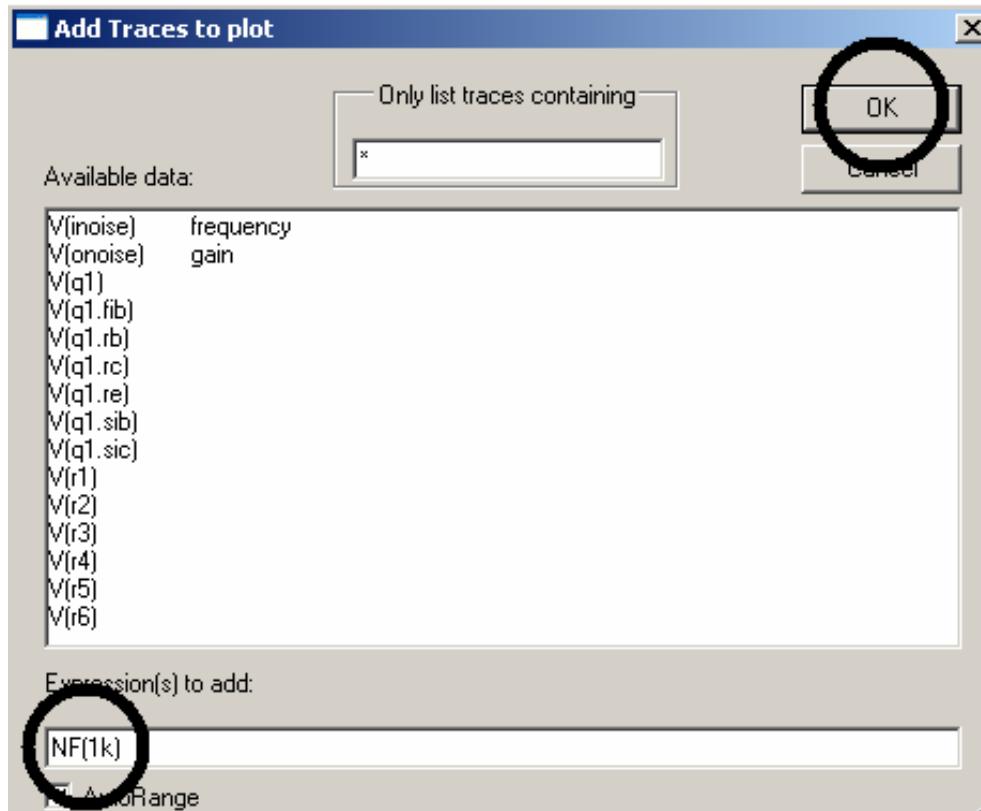
.func NF(1k) 10 * log10(V(inoise) * V(inoise) / (4*k*300*1k))

und diese Zeile muss mit einem Texteditor in eine neue Datei

plot.defs

eingetippt werden. Diese wird dann im LTSpice – Programmordner gespeichert.

Achtung: Nach dem Abspeichern dieser neuen Datei „plot.defs“ muss unserer LTSpice erst geschlossen und dann nochmals neu gestartet werden, damit diese Erweiterung wirksam wird!!



Nun geht es recht schnell:
Man sorgt zuerst durch einen rechten Mausklick und „Add Plot Pane“ für ein leeres Ergebnisdiagramm.

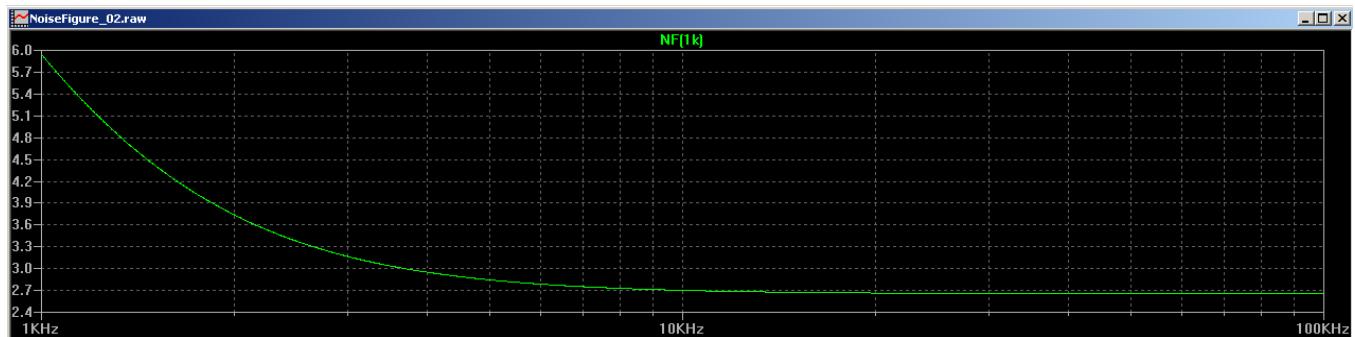
Darauf wird anschließend nochmals rechts geklickt und „Add Trace“ aufgerufen.
In die leere Kommandozeile (Siehe linkes Bild) tippt man nun

NF(1k)

Zur Darstellung der Noise Figure bei einem Quellwiderstand von $1\text{k}\Omega$ ein und drückt OK.

Das folgende Ergebnis weist lediglich einen Schönheitsfehler auf:

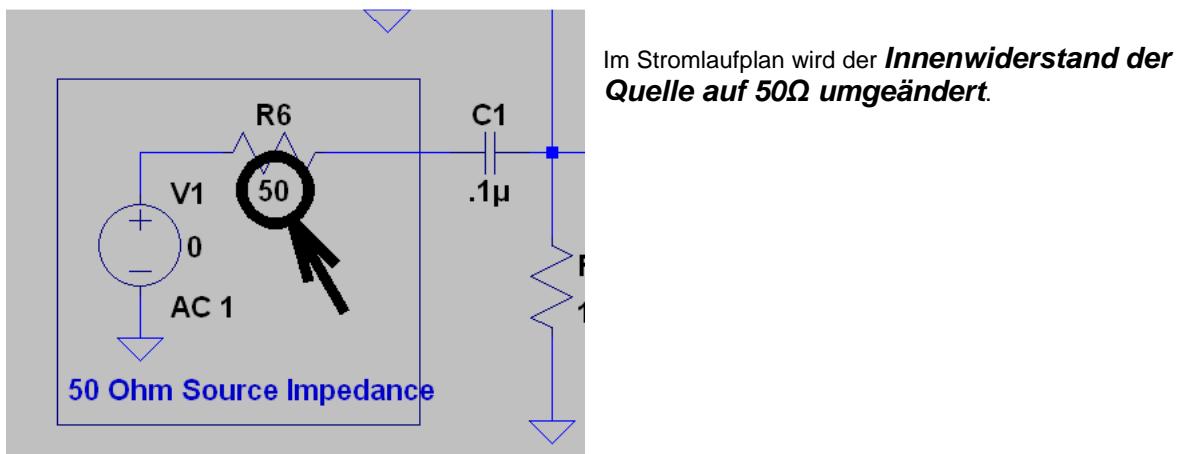
Es fehlt die dB-Eichung der vertikalen Achse und die lässt sich auch nachträglich nicht reinflicken...



Zusatzaufgabe:

Bei Kommunikationssystemen arbeitet man meist mit einem Systemwiderstand von 50Ω . Wir wollen deshalb die Simulation der Noise Figure NF für diesen Widerstandswert wiederholen.

Schritt 1:



Schritt 2:

```
plot.defs - Editor
Datei Bearbeiten Format Ansicht ?
File: C:\Programme\LTC\SwCADIII\plot.defs
*
* Define parameters and functions that you whish to be able to use in
* data plots in this file with .param and .func statements.

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

Die Datei „**plot.defs**“ wird mit dem Texteditor aufgerufen, die Berechnungsfunktion auf „**NF(50)**“ umgeschrieben und schließlich alles erneut als „**plot.defs**“ gespeichert.

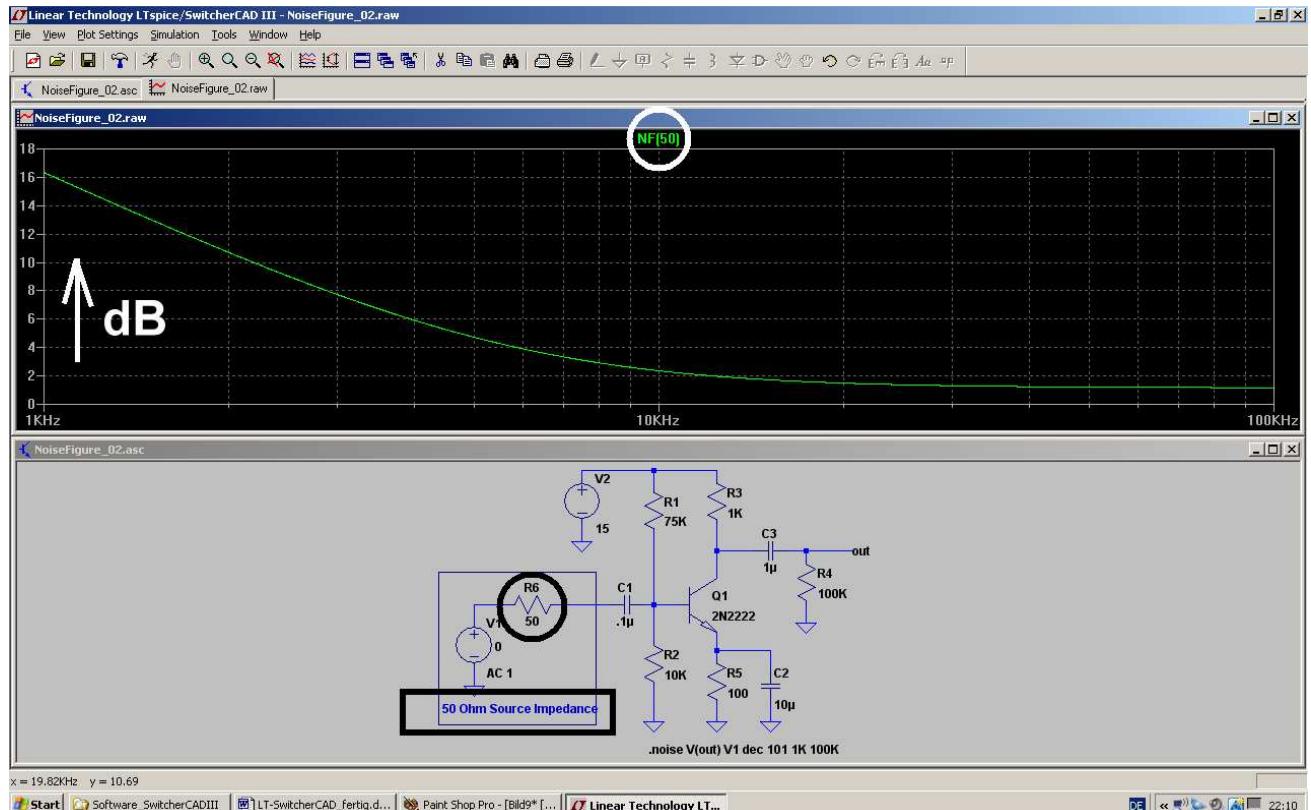
Achtung:

Jetzt muss das LTSpice-Programm erst nochmals neu gestartet werden, damit diese geänderte Funktion aktiv werden kann.....

Schritt 3:

Nun wird simuliert, eine neue „Plot Pane“ geöffnet und anschließend über „Add Trace“ dafür gesorgt, dass „NF(50)“ dargestellt wird.

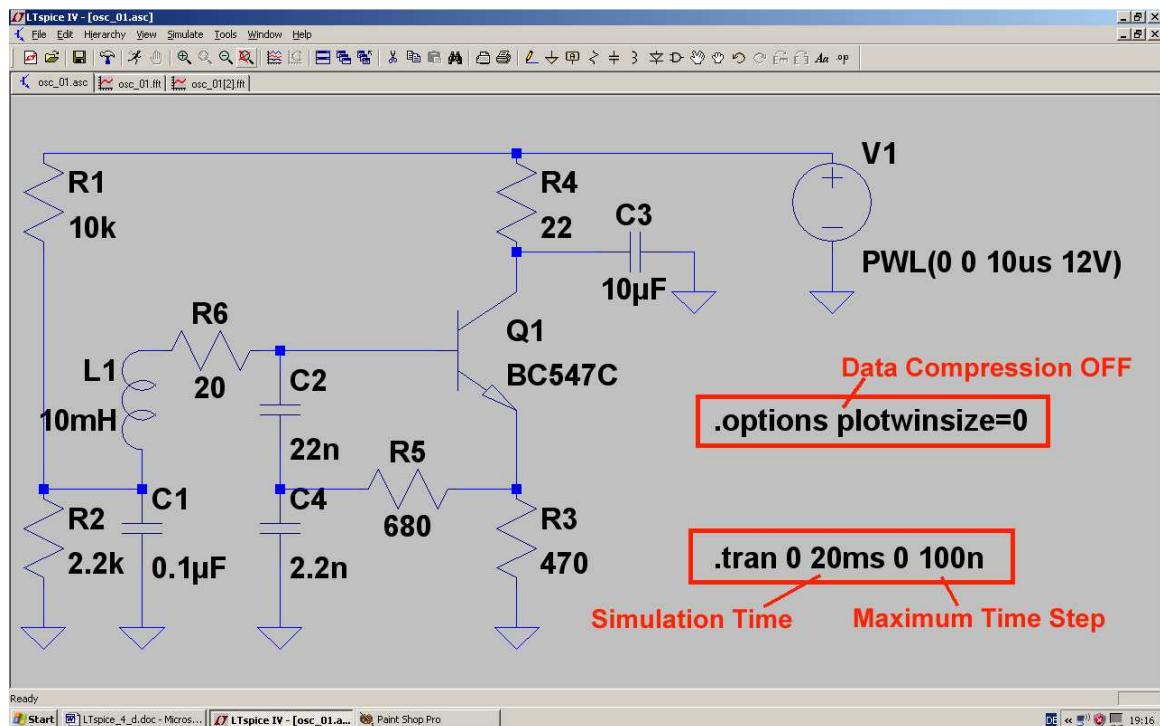
Dazu ist beim Aufruf des „Add trace“- Menüs in der Zeile „Expression(s) to add“ der Eintrag „NF(50)“ erforderlich!



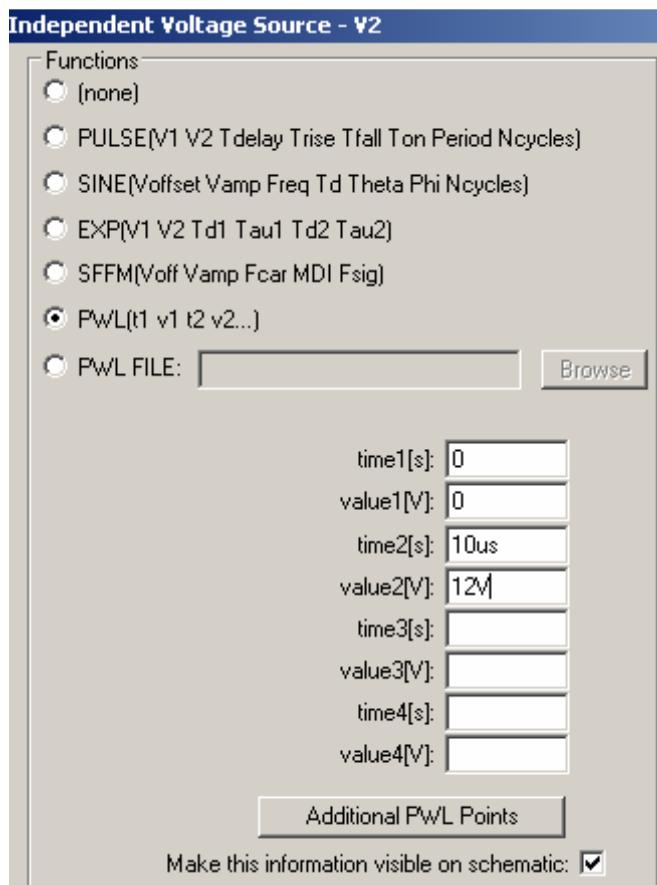
Die senkrechte Achse ist nun wieder in dB geteilt, aber diese Angabe muss man sich zum Diagramm dazu denken.....

18. Untersuchung eines Sinus-Oszillators

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



Folgende Informationen sind dazu wichtig:



1) Damit die Schaltung in der Simulation überhaupt schwingt, müssen wir ihr einen „Schubs“ erteilen. Dazu programmieren wir die Betriebsspannung als

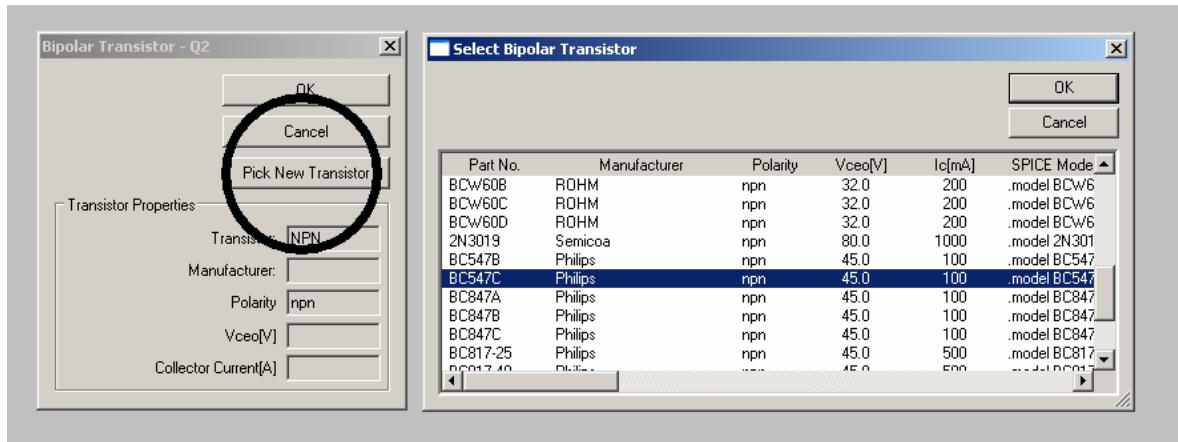
„Rechtecksprung von 0 auf +12V mit einer Anstiegszeit von 10 Mikrosekunden“.

Deshalb klicken wir nach dem Platzieren der Betriebsspannungsquelle mit der rechten Maustaste auf ihr Symbol, klicken auf „Advanced“ und programmieren diesen Verlauf.

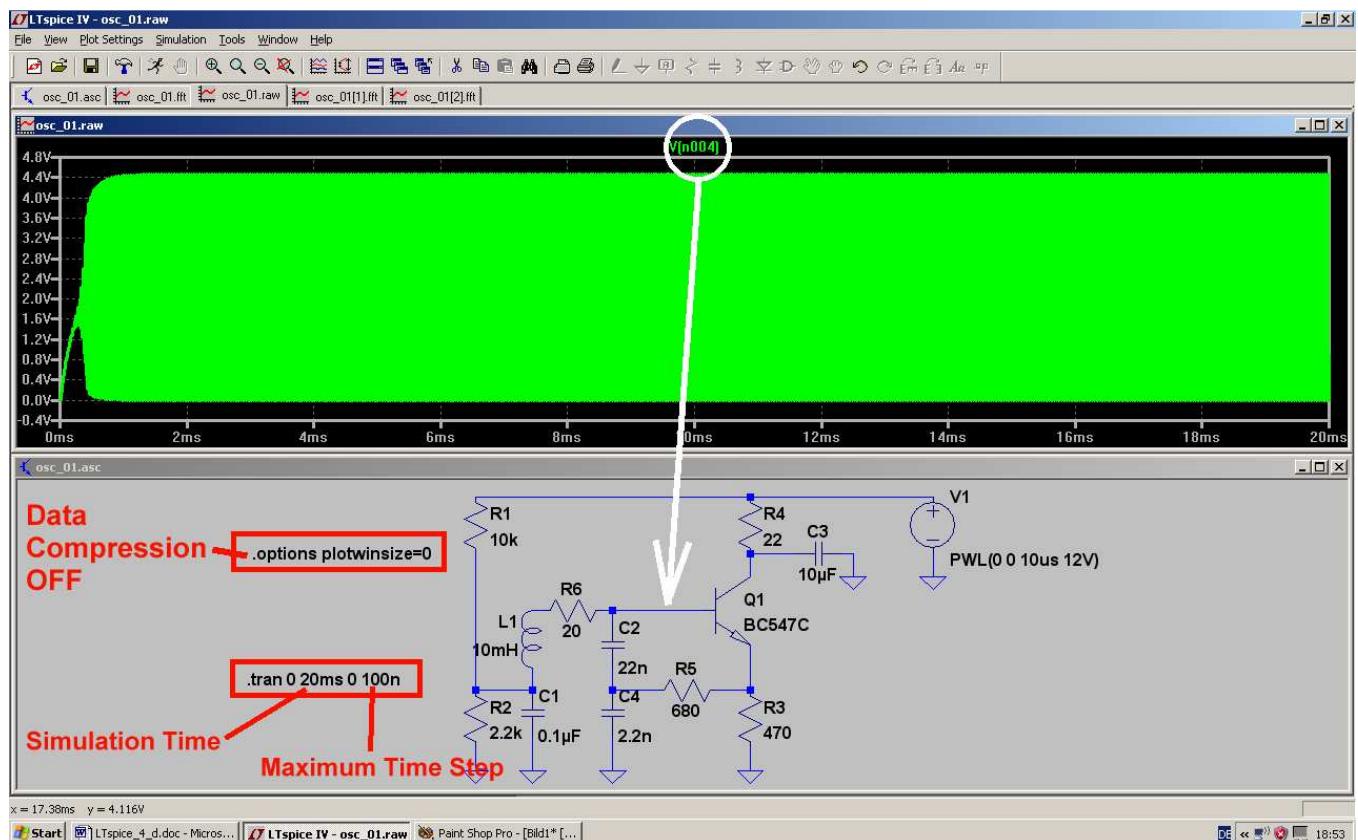
2) Bei den „Simulation Settings“ wählen wir „Time domain (Transient)“ und einen Simulationszeitraum von **0....20 Millisekunden**. Das ergibt später im Spektrum eine **Frequenzauflösung von 1/20ms = 50Hz**.

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 $20\text{ms} / 100\text{ns} = 200\,000$ und können bei der FFT mit 131072 Werten arbeiten. Dazu müssen wir aber auch die **Datenkompression** mit der Direktive „**.options plotwinsize=0**“ ausschalten.

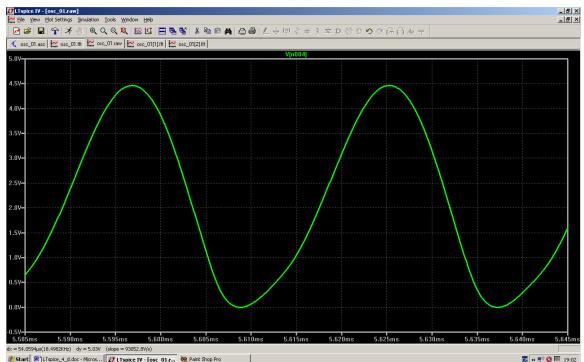
3) Beim Transistor holen wir zunächst aus der Bibliothek das Symbol „**npn**“ und setzen es in der Schaltung ab. Sobald wir nun mit der Maus auf das Symbol rollen und rechts klicken, kommen wir an das Menü „**Pick New Transistor**“ heran. Darin findet sich der Typ „**BC547C**“, und mit OK wird er in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



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



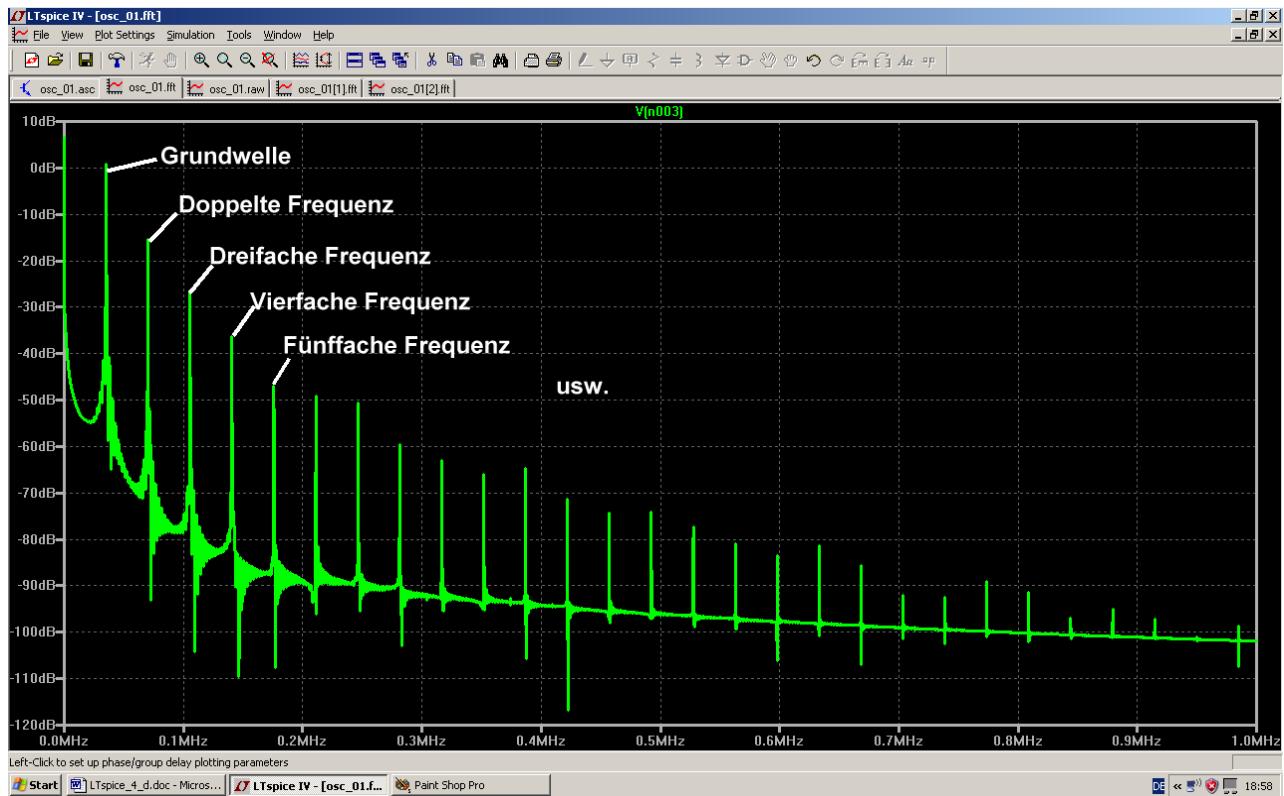
Interessant ist es nun, nur einige Schwingungen herauszuzoomen. Dann kann man nämlich nicht nur die Kurvenform auf Verzerrungen kontrollieren, sondern auch (z. B. mit Hilfe des Cursors) über die Periodendauer des Signals seine Frequenz bestimmen:



Eine Messung der Periodendauer auf diese Art ergibt 28 Mikrosekunden. Dazu gehört eine Frequenz von 35,7kHz. Die Kurvenform ist noch nicht so richtig schön sinusförmig und das deutet darauf hin, dass die „Mitkopplung“ noch zu stark ist.

Aber 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 131 072 Punkten. 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 Emitterwechselstromes in den Schwingkreis, deckt dessen 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 und die Schaltung kann nicht mehr schwingen.

Aufgabe:

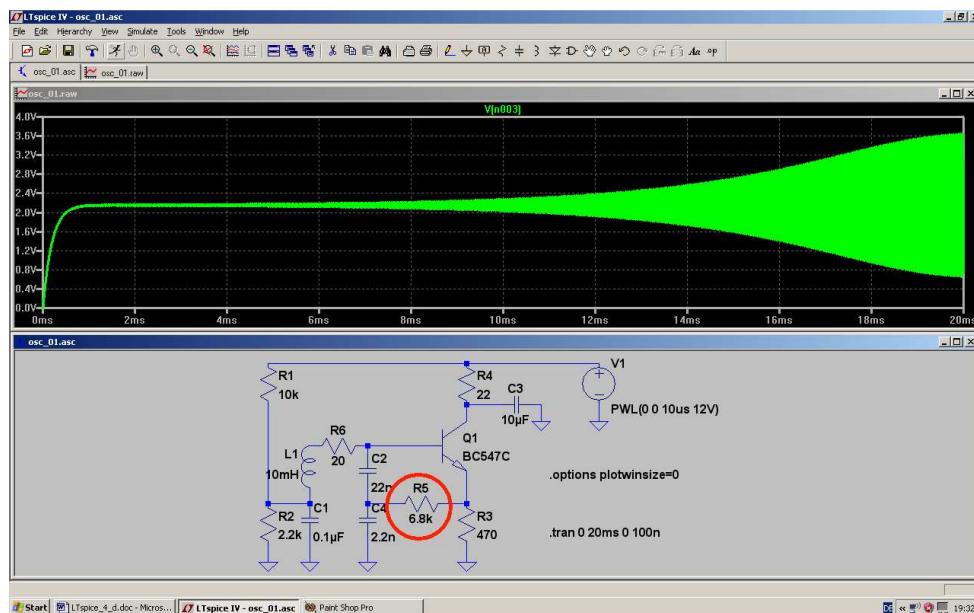
Versuchen Sie, diesen Punkt durch wiederholtes Erhöhen von R5 und anschließender Simulation möglichst genau zu finden. Beobachten Sie dabei auch die Verbesserung der Kurvenform und die Veränderung des zugehörigen Oberwellenanteil über das simulierte Frequenzspektrum.

Beachten Sie auch, dass das „Anschwingen“ umso länger dauert, je näher Sie diesem Punkt kommen -- den da startet der Oszillator nicht mehr.

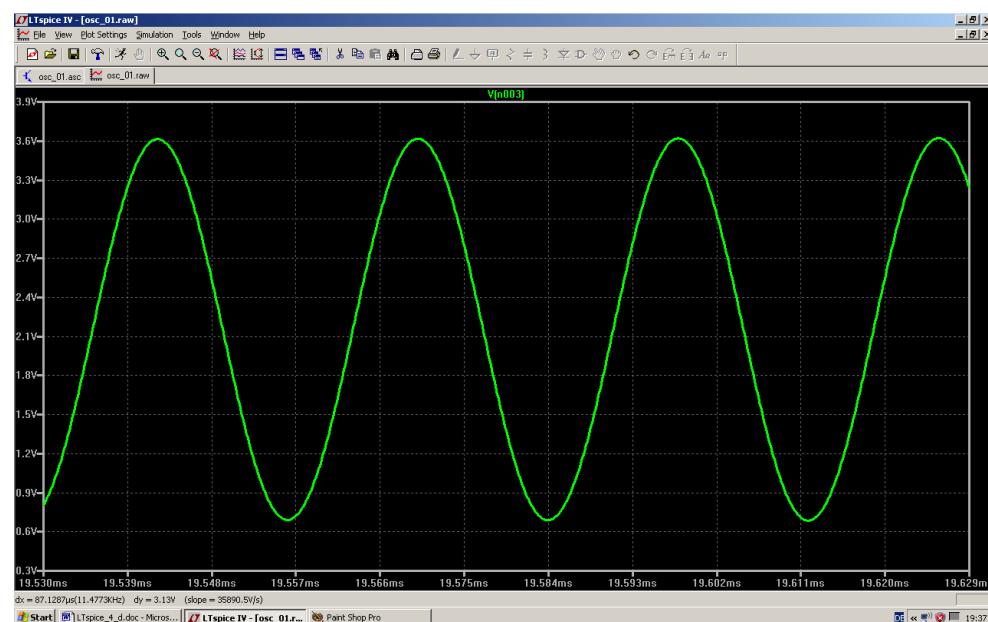
Dieser Punkt ist bei $6,8k$ schon fast erreicht. Deshalb wird man in der Praxis höchstens $5,6k$ als nächsten Normwert einsetzen.

Prüfen Sie diese Information nach.

Musterlösung:

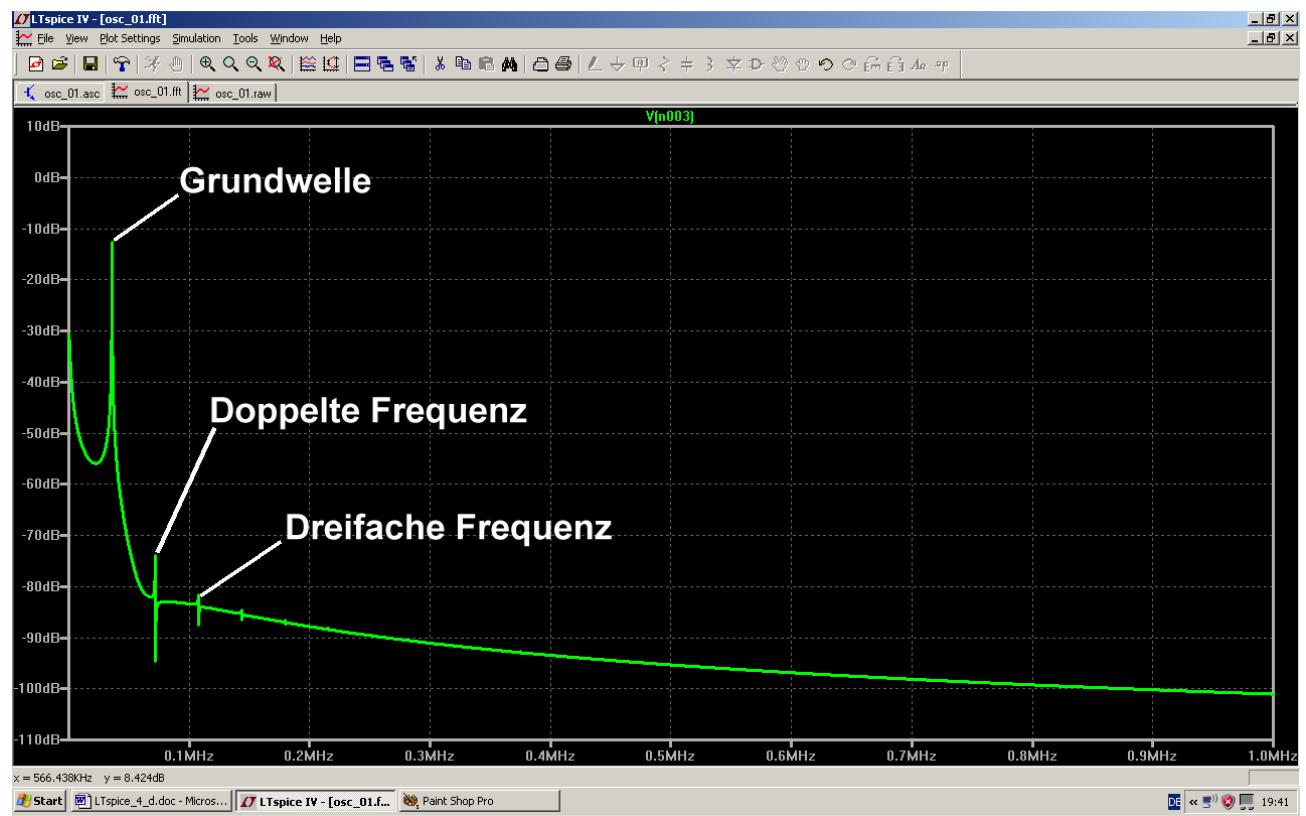


Da sieht man (bei $R5 = 6,8k$) deutlich das mühsame Anschwingen....



...aber die Kurvenform ist nun natürlich (fast) perfekt!

Und der Blick auf das Frequenzspektrum bestätigt das.



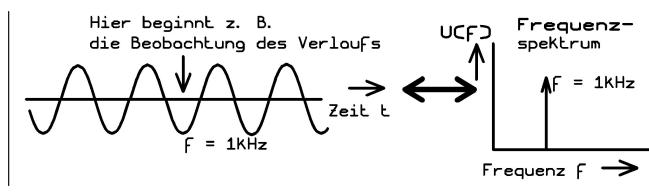
19. Signalformen und Oberwellen

19.1. 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:

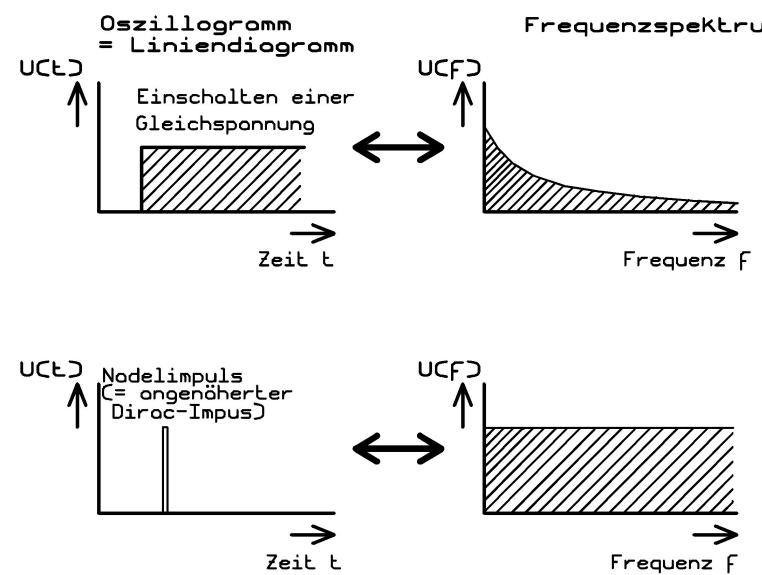


Nur eine rein sinusförmige Spannung (hier: $f = 1 \text{ kHz}$), die vor sehr langer Zeit eingeschaltet wurde und die in absehbarer Zeit nicht aufhört (...also weiterläuft bis zu den Flügen von Raumschiff Enterprise im Jahr 2150....) besteht nur aus einer einzigen Frequenz.

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 Oszilloscop-Signale („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!**

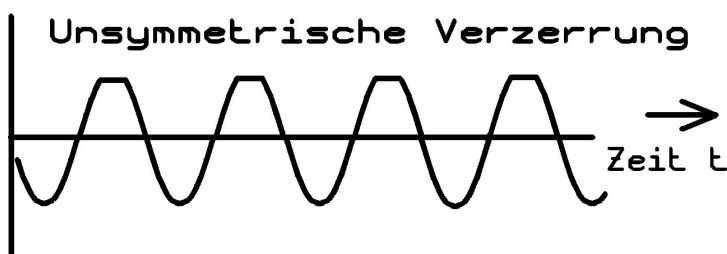
Verzerrungen bei periodischen Signalen

Da merken wir uns gleich den Schlüsselsatz:

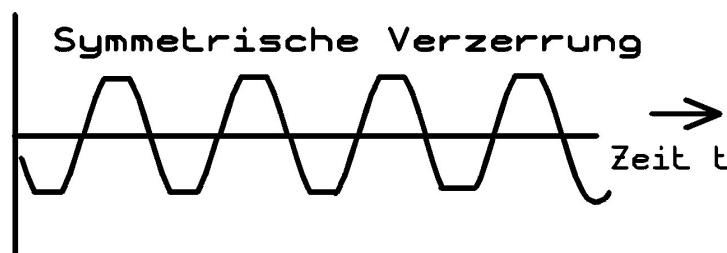
Auch die allerkleinste, aber sich regelmäßig wiederholende Abweichung von der idealen Sinusform führt zum zusätzlichen Auftreten neuer Signale im Frequenzspektrum!

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:

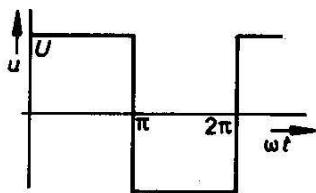
Sobald man am idealen Sinus irgendwas im **Rhythmus eines anderen Signals verändert** (= Amplitude, Phase oder Frequenz), gelangt man zu den entsprechenden

Modulationsarten.

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

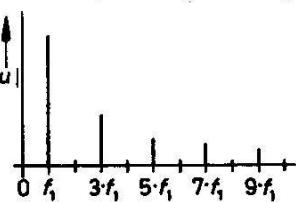
Auf dem nächsten Blatt folgt noch eine kleine Zusammenstellung einiger Signalformen und der zugehörigen Spektren:

Rechteckspannung

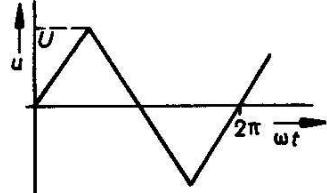


$$f(t) = \frac{4 \cdot U}{\pi} (\sin \omega t + \dots + \frac{1}{3} \sin 3 \omega t + \frac{1}{5} \sin 5 \omega t + \dots)$$

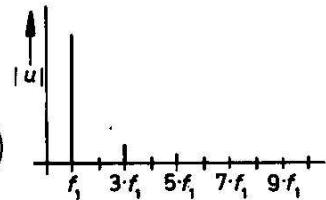
Spektrum: Absolutwerte der auftretenden Frequenzen



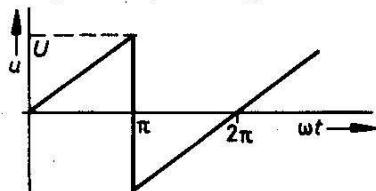
Dreieckspannung



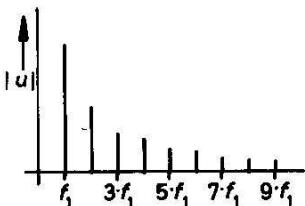
$$f(t) = \frac{8 U}{\pi^2} (\sin \omega t - \dots + \frac{1}{3^2} \sin 3 \omega t + \frac{1}{5^2} \sin 5 \omega t - \dots)$$



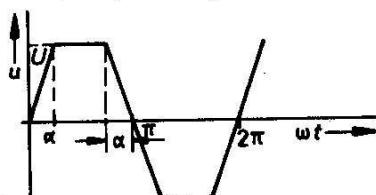
Sägezahnspannung



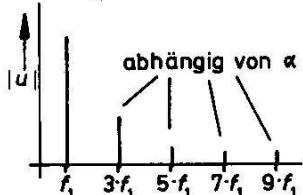
$$f(t) = \frac{2 U}{\pi} (\sin \omega t - \frac{1}{2} \sin 2 \omega t + \dots + \frac{1}{3} \sin 3 \omega t - \dots +)$$



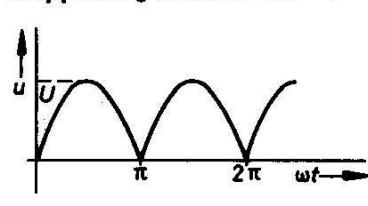
Trapezspannung



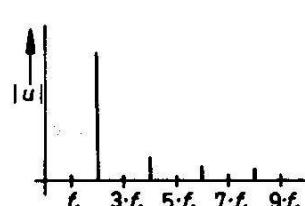
$$f(t) = \frac{4 \cdot U}{\pi \cdot \alpha} (\sin \alpha \cdot \sin \omega t + \dots \text{tot} + \frac{\sin 3 \alpha}{3^2} \cdot \sin 3 \omega t + \dots + \frac{\sin 5 \alpha}{5^2} \sin 5 \omega t + \dots)$$



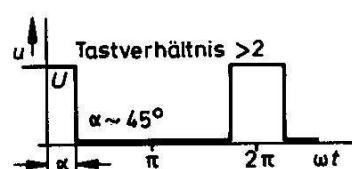
Doppelweg-Gleichrichtung



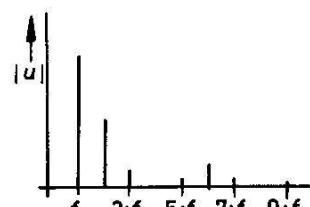
$$f(t) = \frac{4 U}{\pi} \left(\frac{1}{2} - \frac{1}{1 \cdot 3} \cos 2 \omega t - \dots - \frac{1}{3 \cdot 5} \cos 4 \omega t - \dots - \frac{1}{5 \cdot 7} \cos 6 \omega t - \dots \right)$$



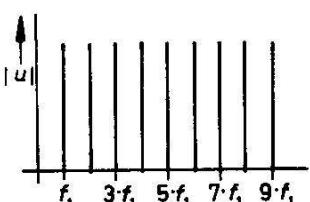
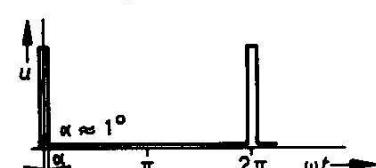
Rechteckspannung Tastverhältnis > 2



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



Nadelimpulse



19.2. 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 = 1 Millisekunde.

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 $10\text{ms} / 100\text{ns} = 100\,000$ echte Samples zur Verfügung. Die Datenkompression schalten wir (mit Rücksicht auf die geplante FFT) durch die Direktive

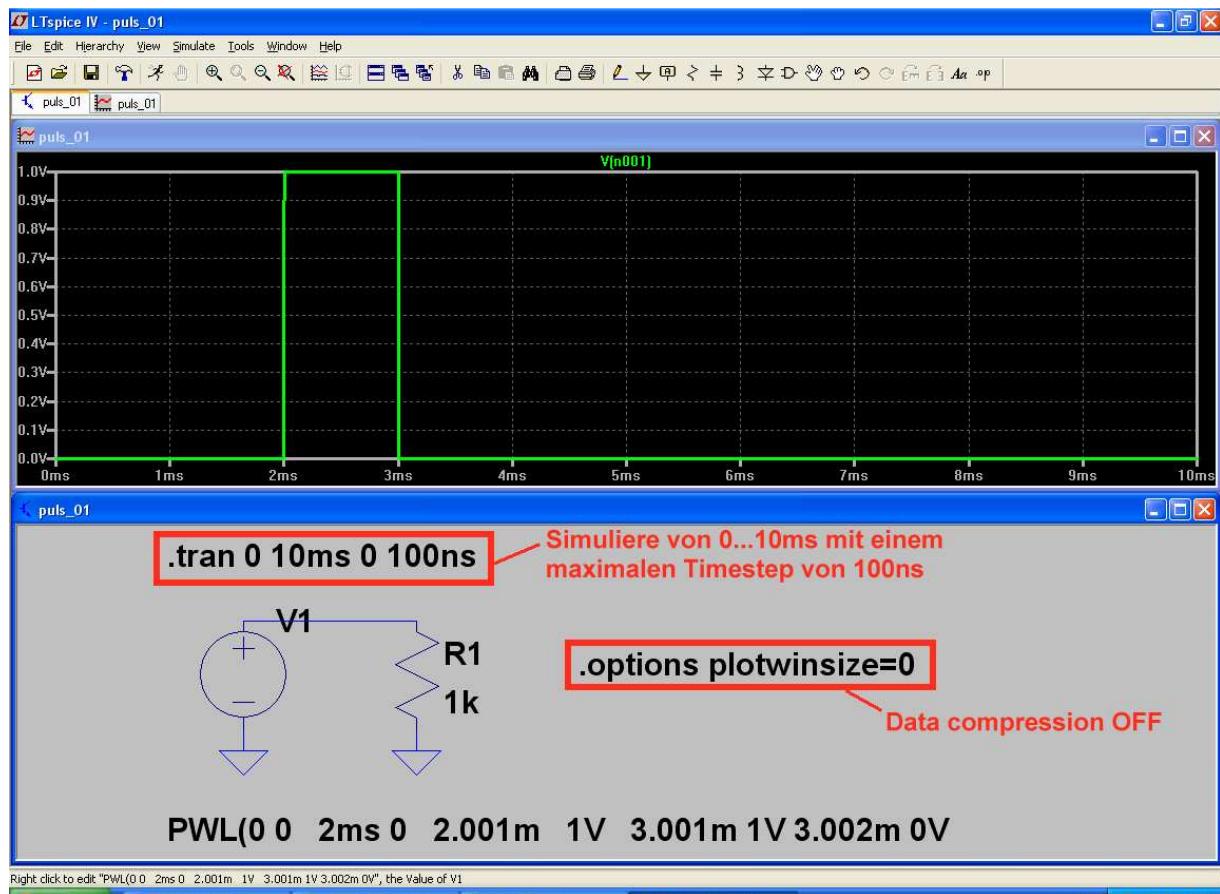
.options plotwinsize=0

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

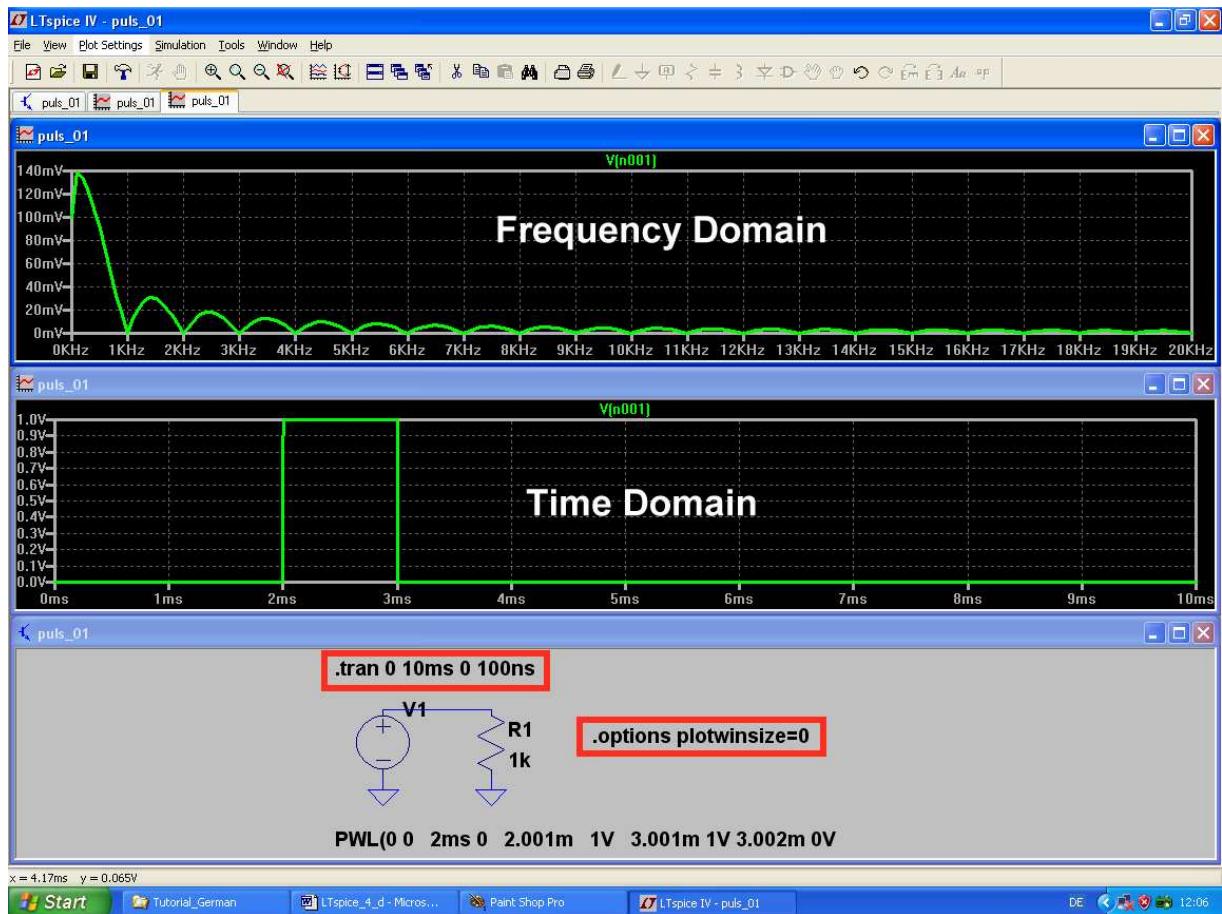
Simuliert man anschließend und wählt hinter „Window“ die Option „**Tile Horizontally**“, dann muss Folgendes zu sehen sein:



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:

- 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 **Stopffrequenz 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:

- 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“.
- Ihre Amplitude verläuft nach einer „ $\sin x / x$ “-Funktion.
- Die **Nullstellen** im Leistungsdichte-Verlauf liegen **IMMER bei Vielfachen vom Kehrwert der Pulslänge**, also bei

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

(Bei unserer Pulslänge von 1ms liegen die Nullstellen deshalb stets bei $n \times 1\text{kHz}$)

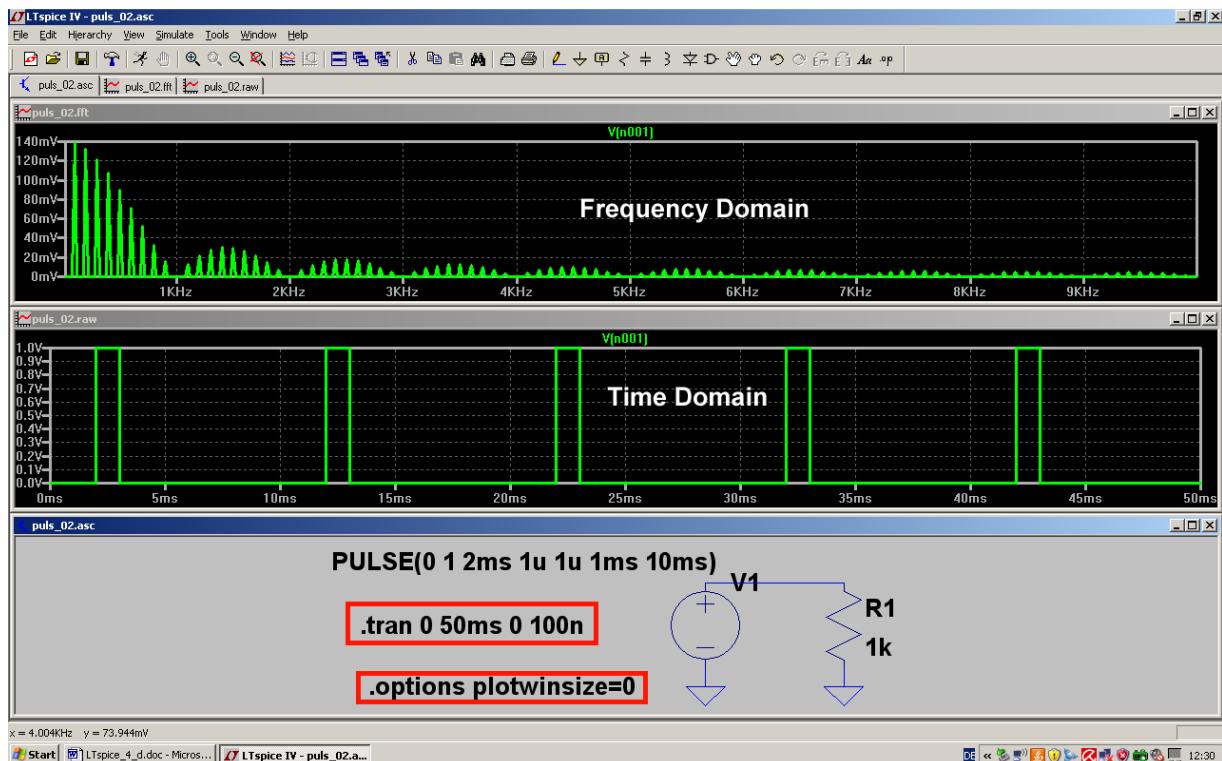
Je kürzer der Puls wird, desto weiter wird folglich die erste Nullstelle zu höheren Frequenzen hin verschoben.

19.3. 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 $1 / 50\text{ms} = 20\text{Hz}$ 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:

Senkrechte Achse: Linear

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



Man erkennt:

- 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**.
- 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

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

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

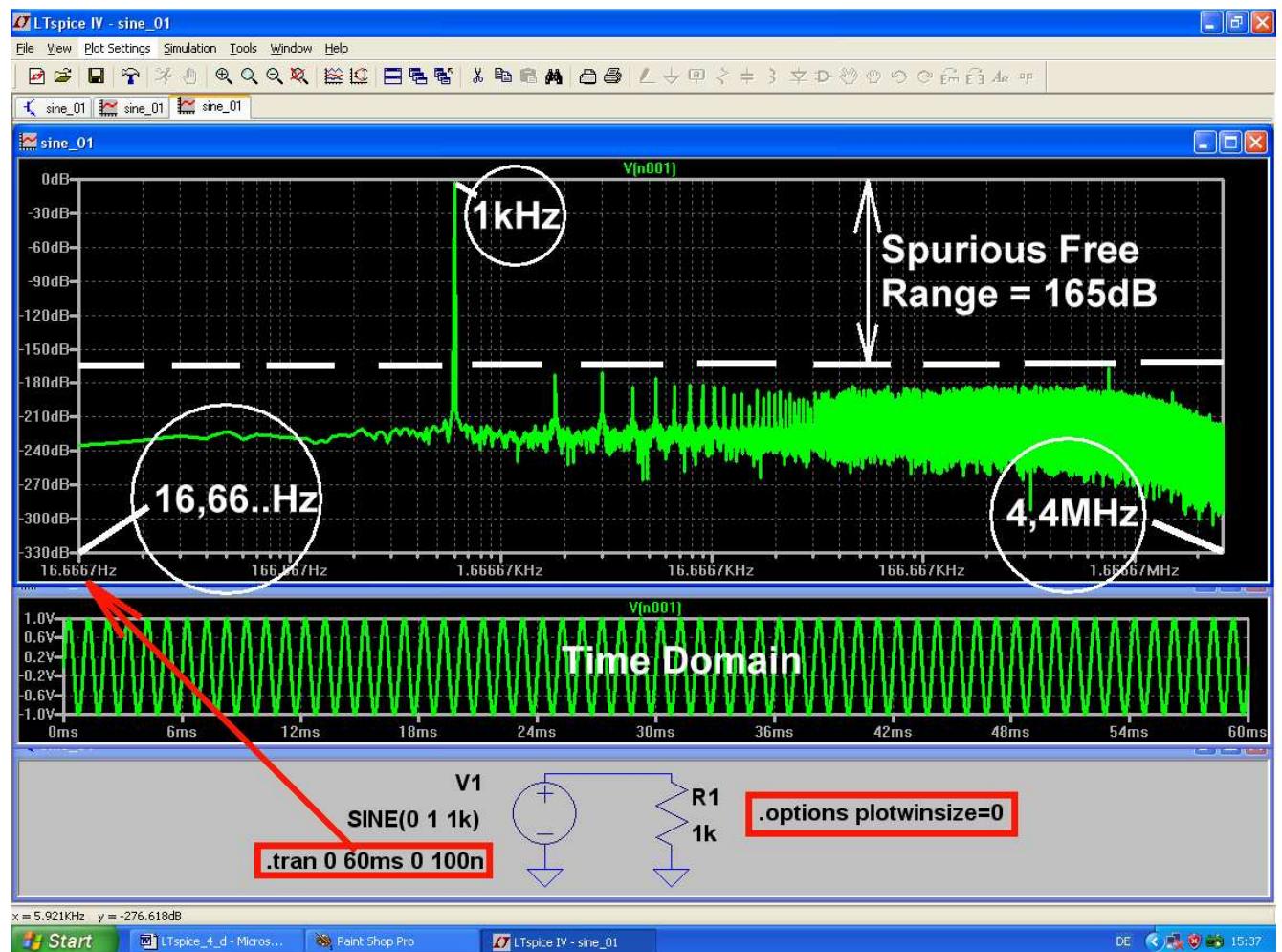
19.4. Der ideale Sinus

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 Voraussetzung 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...

19.5. 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:

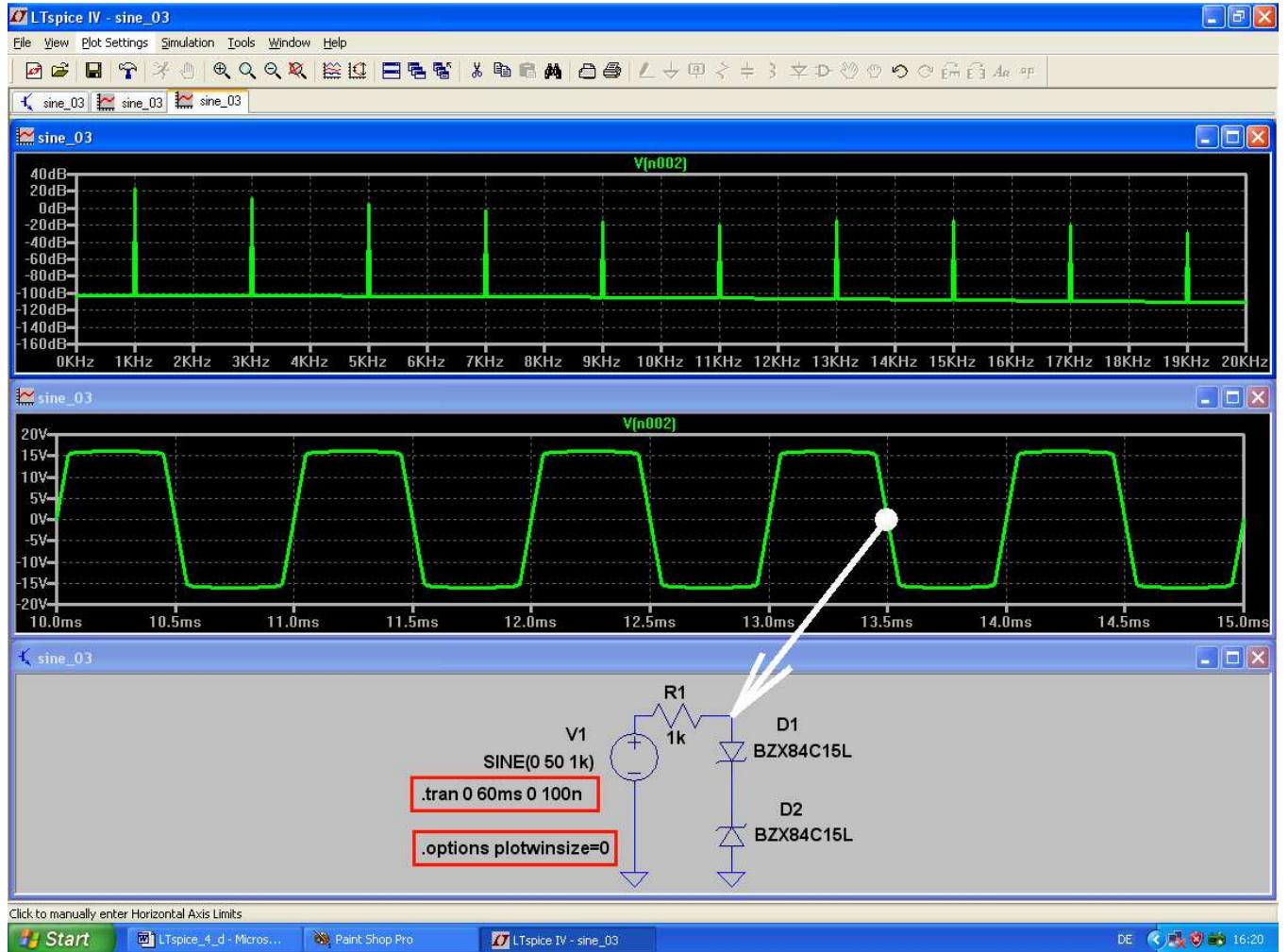


19.6. Der symmetrisch begrenzte Sinus

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

Wieder wird ein Sinus mit $f = 1\text{kHz}$ / $U_{peak} = 50\text{V}$ 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.



20. 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 Leistungsdichte“ 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 $U_a(t)$ für einen beliebigen Verlauf der Eingangsspannung $U_e(t)$ erhält man, indem man den zeitlichen Verlauf der Eingangsspannung $U_e(t)$ mit der vorher ermittelten Impulsantwort $g(t)$ „faltet“ (englisch: application of the „convolution integral“).

$$U_a(t) = U_e(t) * g(t)$$

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: Dirac-Impuls speist einen 160Hz-RC-Tiefpass

Wir stellen uns folgende Aufgabe:

Speisen Sie einen **Tiefpass ($R = 100\text{k}\Omega$, $C = 10\text{nF}$)** 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:

$$f_{\text{Grenz}} = \frac{1}{2\pi \cdot RC} = \frac{1}{2\pi \cdot 100\text{k}\Omega \cdot 10\text{nF}} = 159\text{Hz}$$

Das ergibt:

$$RC = \frac{1}{2\pi \cdot f_{\text{Grenz}}} = \tau = 100\text{k}\Omega \cdot 10\text{nF} = 10^{-3} \text{s} = 1\text{ms}$$

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 1 Voltsekunde ergeben eine **Pulsamplitude von 1 Megavolt. Die Anstiegs- und Abfallzeit muss jeweils 1 Nanosekunde (= 1μs / 1000) betragen.**

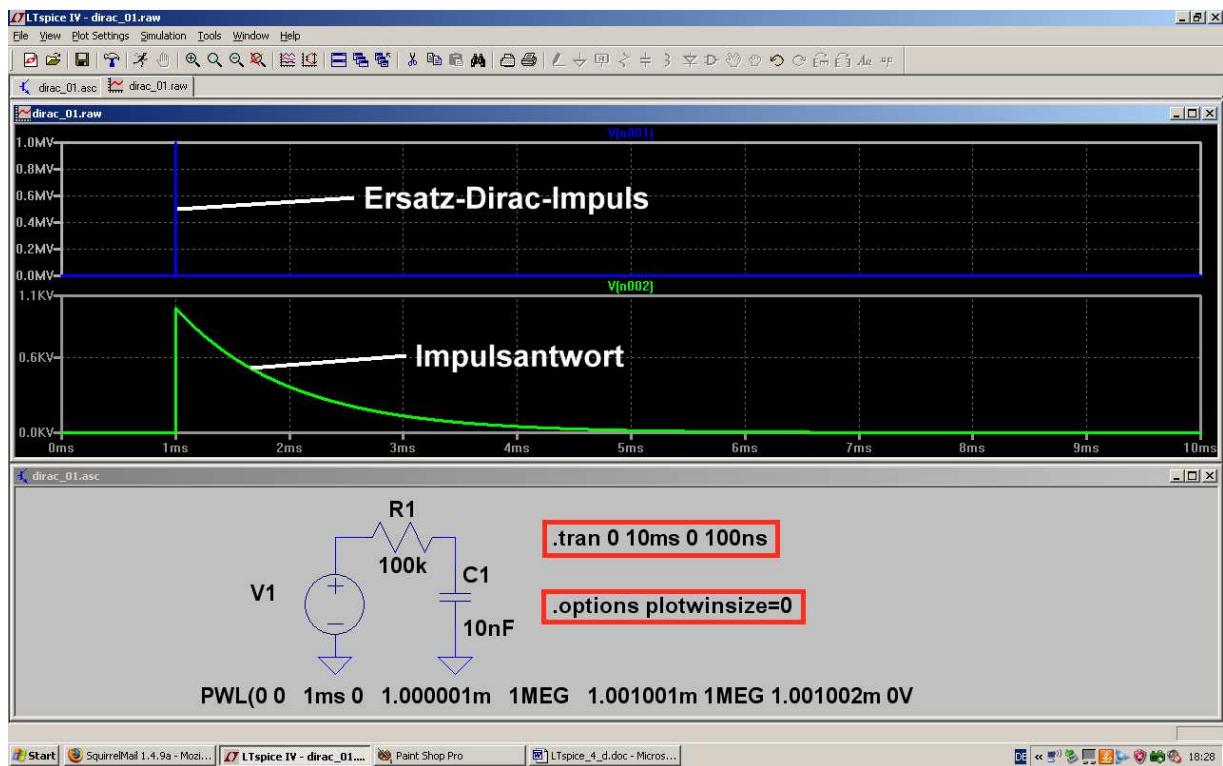
Wir brauchen eine PWL-Quelle und programmieren sie folgendermaßen:

Bei $T = \text{Null}$:	Amplitude Null
Bei $T = 1\text{ms}$	Amplitude Null
Bei $T = 1.000001\text{ms}$	Amplitude 1MEG
Bei $T = 1.001001\text{ms}$	Amplitude 1MEG
Bei $T = 1.001002\text{ms}$	Amplitude Null

Auf der nächsten Seite sieht man den Bildschirm nach der Simulation.

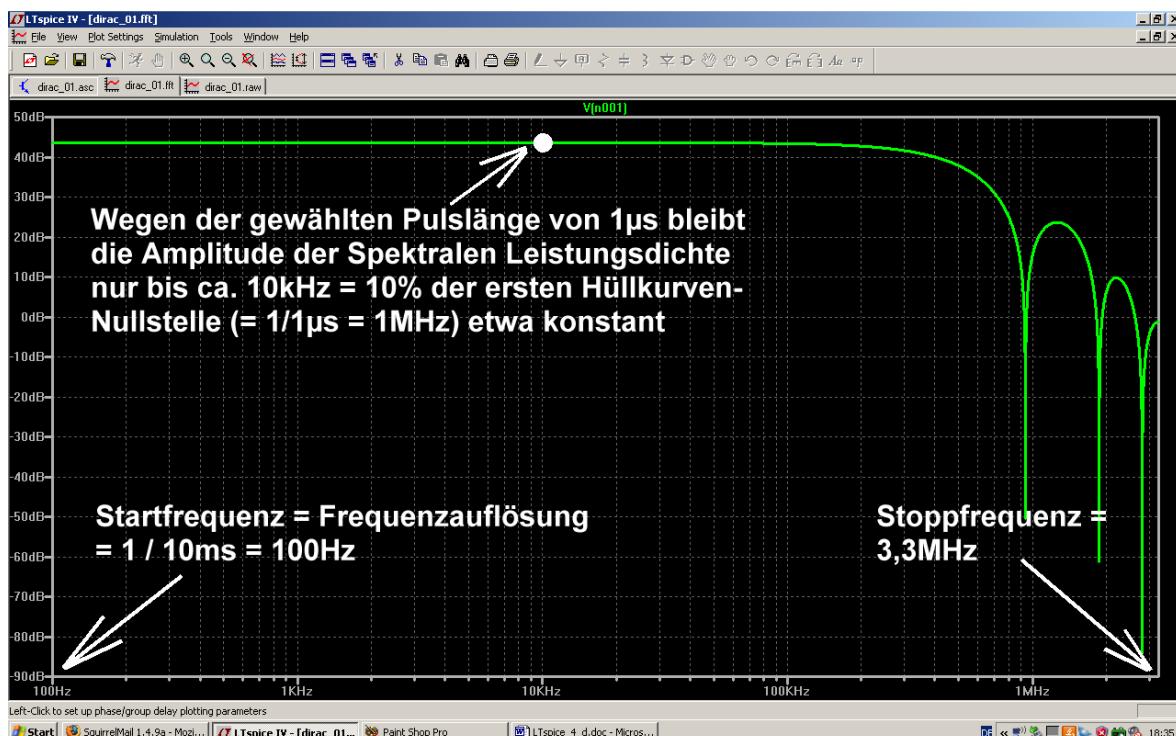
Lösung zu c) Schaltung und Simulation in der Time Domain

Die Programmierung der PWL-Quelle sowie das Simulationskommando finden sich in der unteren Schirmhälfte.



Lösung zu d): FFT der Eingangsspannung

Wir rollen den Cursor auf das Diagramm mit dem Eingangspuls und klicken mit der rechten Maustaste. 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 $t_{\max} = 10\text{ms}$

$$1 / t_{\max} = 1 / 10\text{ms} = 100\text{Hz}$$

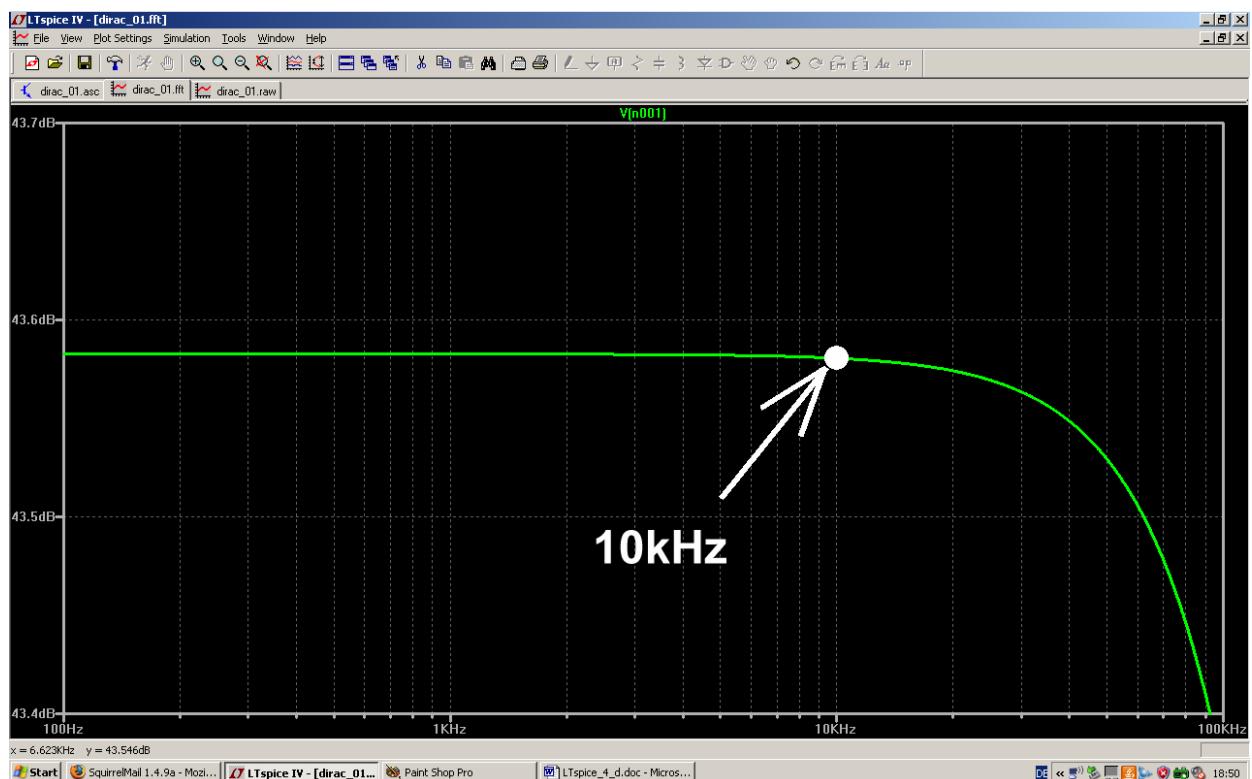
b) Die höchste im Diagramm darstellbare Frequenz (= Stoppfrequenz beim AC-Sweep....) sowie der Dynamikbereich der FFT-Anzeige werden durch die Samplefrequenz bei der Time-Domain Simulation UND durch die Anzahl der für die FFT eingesetzten Samples bestimmt. (Die Samplefrequenz bei der Time Domain Simulation erhält man als $1 / \text{maximaler Timestep}$).

c) Wir haben eine Pulslänge von 1 Mikrosekunde gewählt und das führt zu einer ersten „Nullstellenfrequenz bei der Hüllkurve“ von

$$1 / 1\mu\text{s} = 1\text{MHz}$$

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...

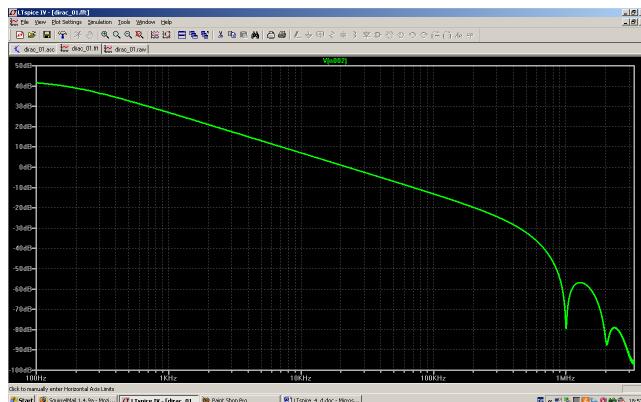
Zusatzaufgaben:

- 1) Testen Sie die Behauptung a) durch Simulationen mit unterschiedlichen **Simulationszeiten** (z. B. 20 ms / 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 Regelungstechniker im Bode-Diagramm auch als „Amplitudengang“ bezeichnet wird. Also werfen wir mal einen Blick auf die vom FFT-Programm ausgegebene Kurve.

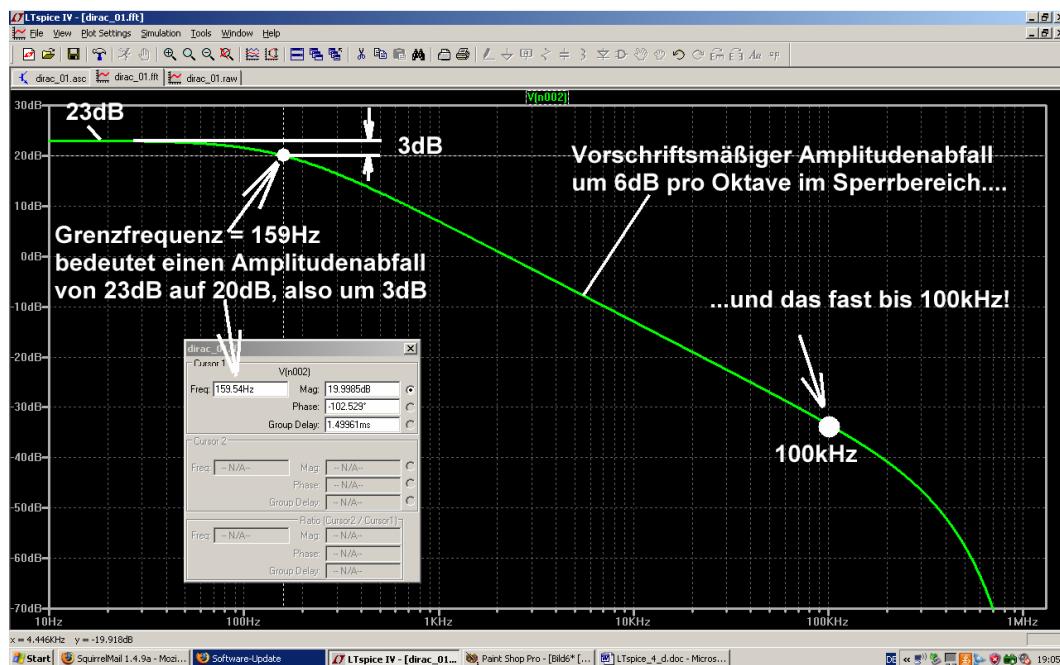


Zunächst sehen wir nur -- wie es sich für einen solchen Tiefpass gehört -- den Amplitudenabfall im Sperrbereich mit -6dB pro Oktave. Bei hohen Frequenzen tauchen natürlich die Schwächen unseres Ersatz-Dirac-Impulses mit den erwähnten Nullstellen in der Hüllkurve auf. Deshalb sind dort die Ergebnisse 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:



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 vorschrittmäßig um -6dB pro Oktave.

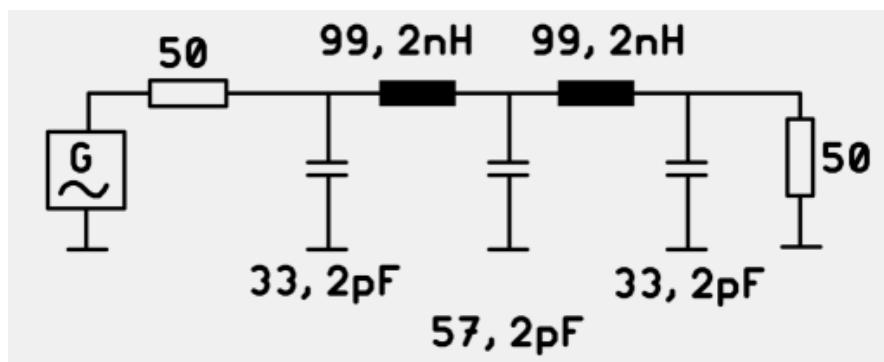
Oberhalb von 15kHz (= etwa der hundertfachen Grenzfrequenz des Tiefpasses) beginnt der Einfluss der Hüllkurve unseres Ersatz – Impuls-Spektrums und ab diesem Bereich darf man irgendwann nicht mehr alles glauben.. Abhilfe bringt da nur eine deutliche Verminderung der Pulsdauer. Das schiebt die erste Hüllkurven-Nullstelle nach oben.

20.2. Zweites Beispiel: Dirac-Untersuchung des 110MHz-Tiefpasses aus Kapitel 14.2

20.2.1. Bestimmung von S21 (= Forward Transmission)

Das waren die damaligen Vorgaben:

„Ripple“-Grenzfrequenz	$f_g = 110 \text{ MHz}$
Spulenarme Ausführung	
Filtergrad	$n = 5$
Systemwiderstand	$Z = 50\Omega$
„Ripple“ (= Welligkeit der Dämpfung im Durchlassbereich) =	0,1 dB.
(



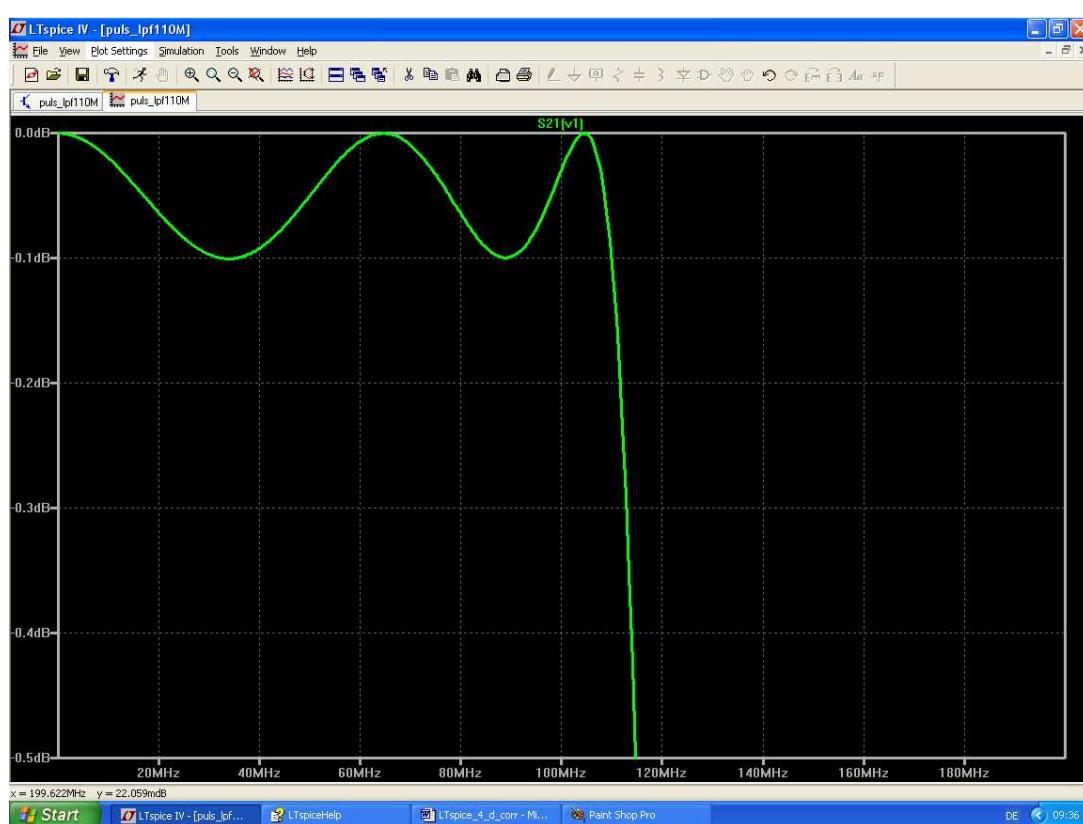
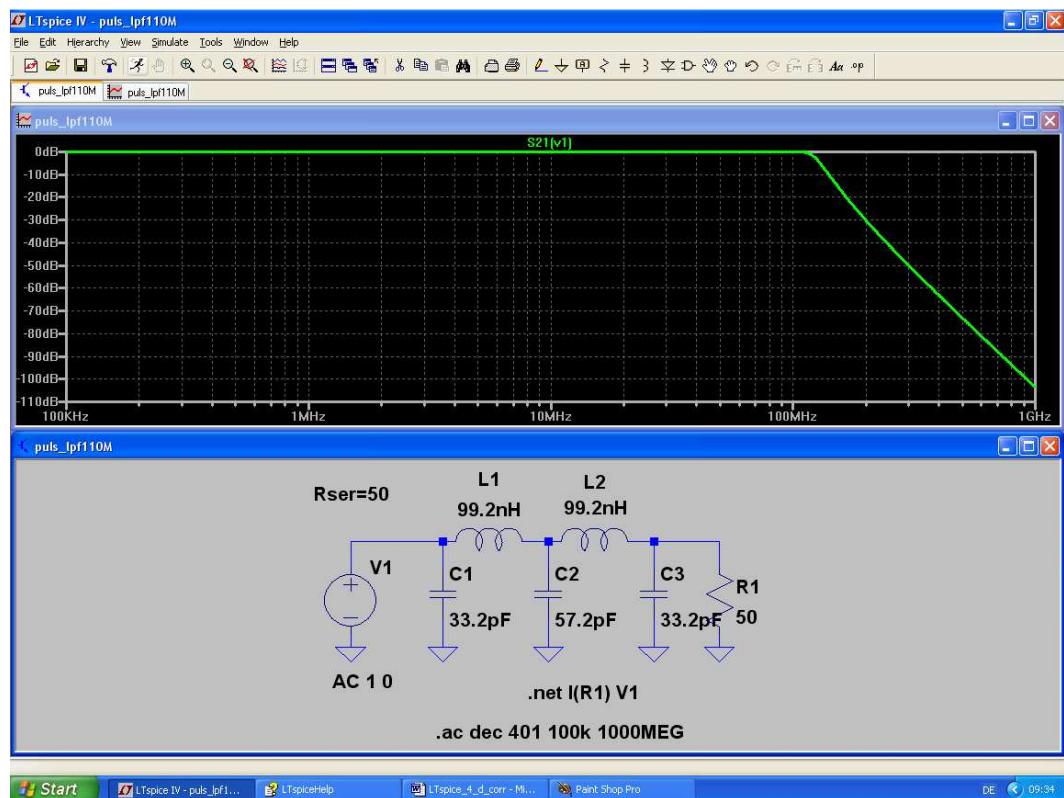
...und das sind die vom Filter-Entwurfsprogramm gelieferten Bauteilwerte für die Schaltung.

Nun stellen wir uns folgende Aufgabe:

- 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.
- 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.
- 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.**
- 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).
- Verwenden Sie zur FFT nur 65 536 echte Samples.** Zeigen Sie am Ergebnisdiagramm 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 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

$$T = \frac{1}{2\pi \cdot f_{Grenz}} = \frac{1}{2\pi \cdot 110\text{MHz}} = 1,44\text{ns}$$

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:

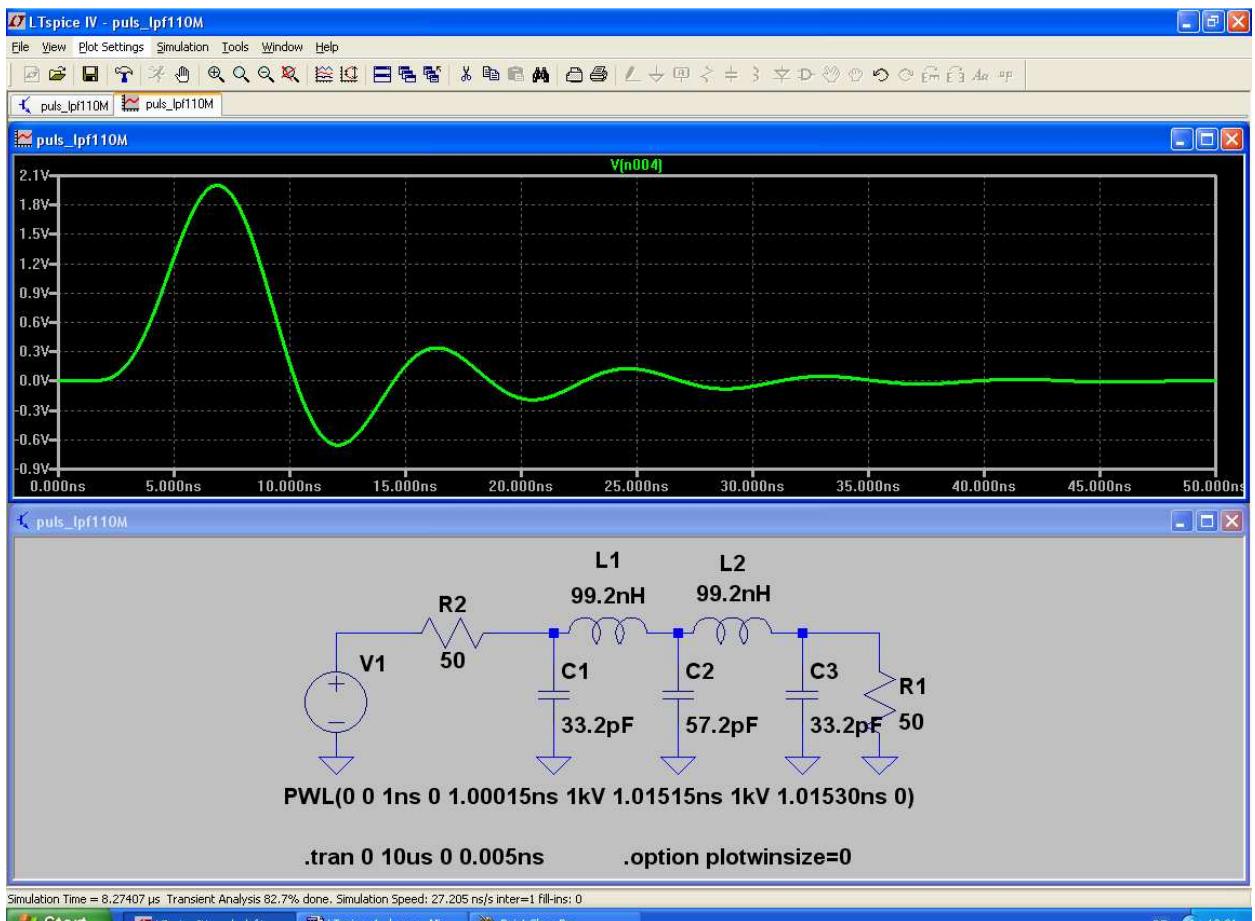
Zeitpunkt T = Null:	U = Null
Zeitpunkt T = 1ns:	U = Null
Zeitpunkt T = 1,00015ns:	U = 1 KiloVolt
Zeitpunkt T = 1,01515ns:	U = 1 KiloVolt
Zeitpunkt T = 1,01530ns:	U = Null

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:

$$t_{max} = 1 / 100\text{kHz} = 10\mu\text{s}$$

Damit LTspice den Dirac-Impuls nicht übersieht, wird der **Maximum Time Step zu 15ps / 3 = 5 Picosekunden** gewählt.

Die Datenkompression ist durch die Direktive „options plotwinsize=0“ abgeschaltet. So sieht das aus, wenn wir uns nur die ersten 50 Nanosekunden anschauen:



Lösung zu d): FFT

Wir simulieren 10 μ s lang mit einer maximalen Auflösung von 5Picosekunden. Das entspricht einer minimalen Anzahl von

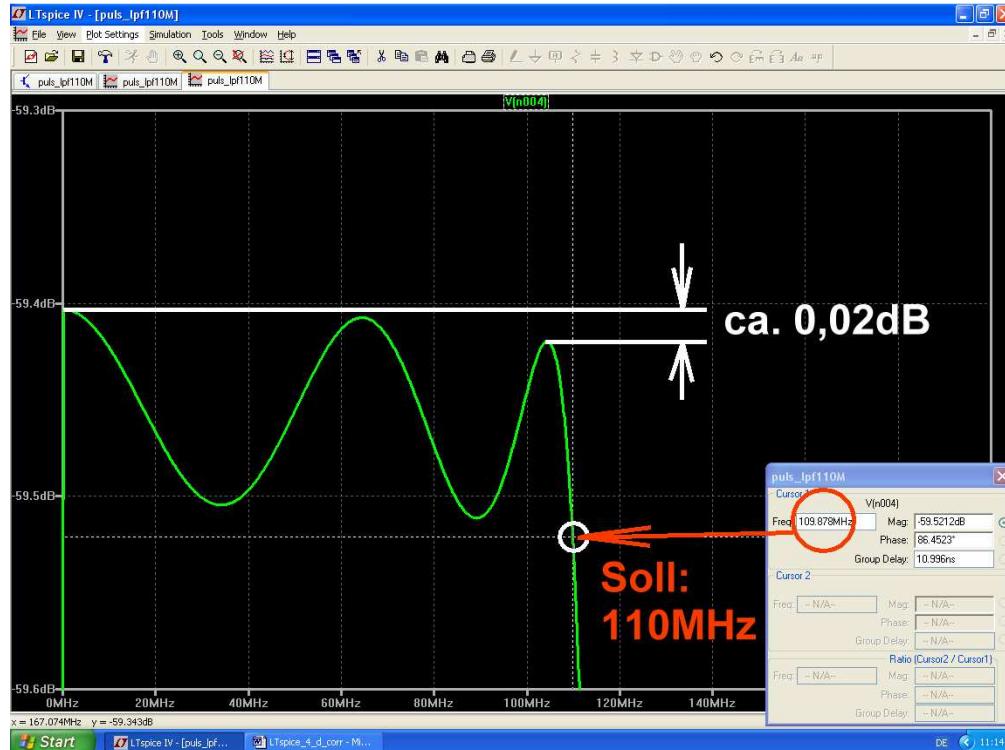
$$10\mu\text{s} / 0,005\text{ns} = 2\,000\,000 \text{ Samples}$$

Also können wir es wagen, bei der FFT mit **1048576 Werten** zu arbeiten.

Das Transformationsergebnis mit linearer Frequenzachse sieht zunächst so aus:

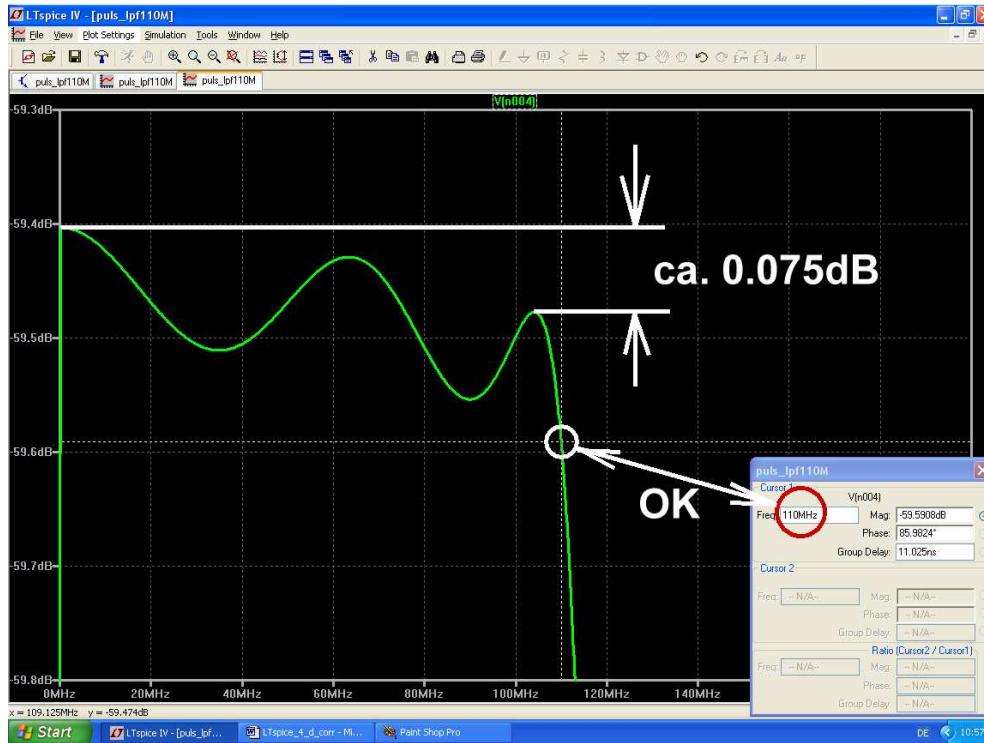


Werfen wir noch einen weiteren Blick auf die Tschebyschef-Wellen im Durchlassbereich und vergleichen sie mit dem am Anfang vorgenommenen AC-Sweep:



Die Abweichungen gegenüber der S-Parameter-Simulation sind fast unbedeutend.

Lösung zu e): Simulation mit 65536 Samples

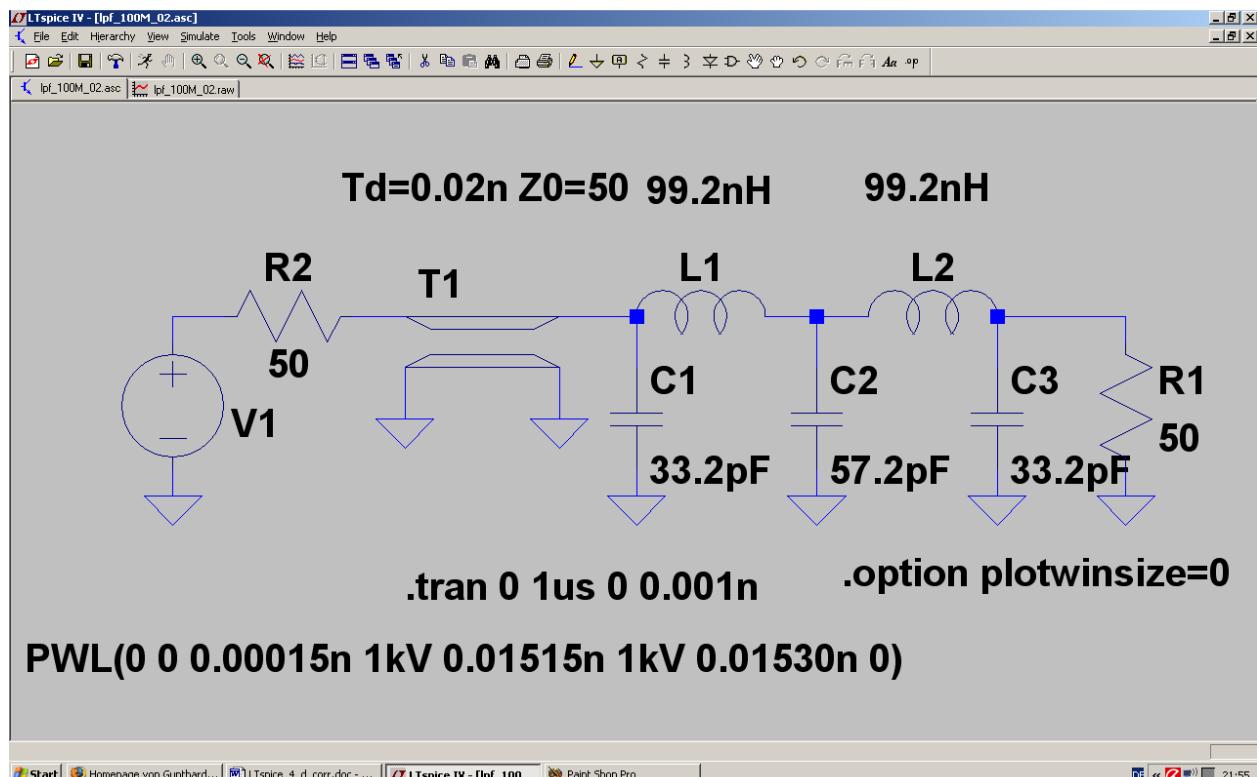


20.2.2. Bestimmung von S11 (= Input Reflection) oder S22 (= Output Reflection)

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 Hinklauenden 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. 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 ein Kabelstück mit $Z = 50\Omega$ 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:

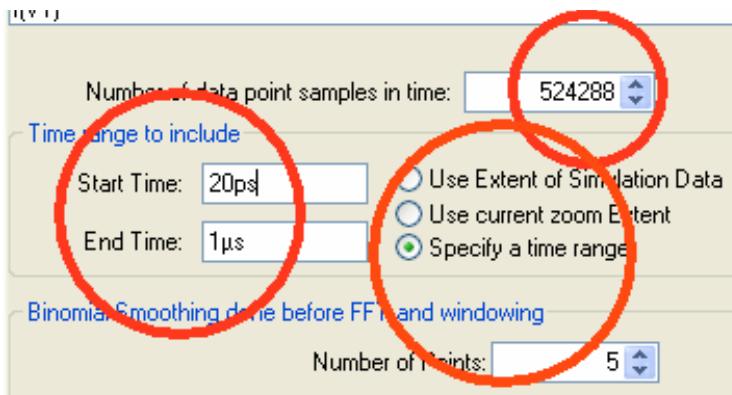
Zeitpunkt T = Null:	$U = \text{Null}$
Zeitpunkt T = 0,00015ns:	$U = 1 \text{ KiloVolt}$
Zeitpunkt T = 0,01515ns:	$U = 1 \text{ KiloVolt}$
Zeitpunkt T = 0,01530ns:	$U = \text{Null}$

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.**

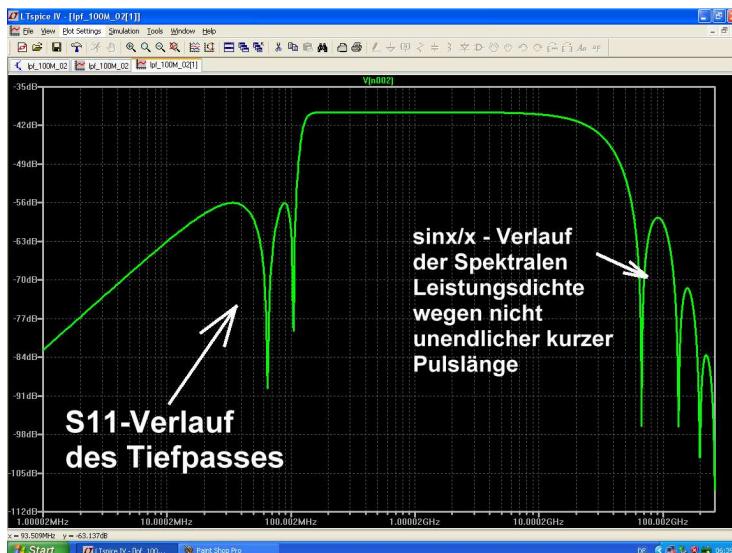
Die Datenkompression ist wie immer abgeschaltet (**.option plotwinsize=0**).

Um die Rechenzeit nicht zu lang werden zu lassen, wurde nur eine Simulationszeit von $1\mu\text{s}$ vorgesehen, was eine Frequenzauflösung von $1 / 1\mu\text{s} = 1\text{MHz}$ 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 $1\mu\text{s} / 0.001\text{ns} = 1\,000\,000$ berechnete Samples zur Verfügung.

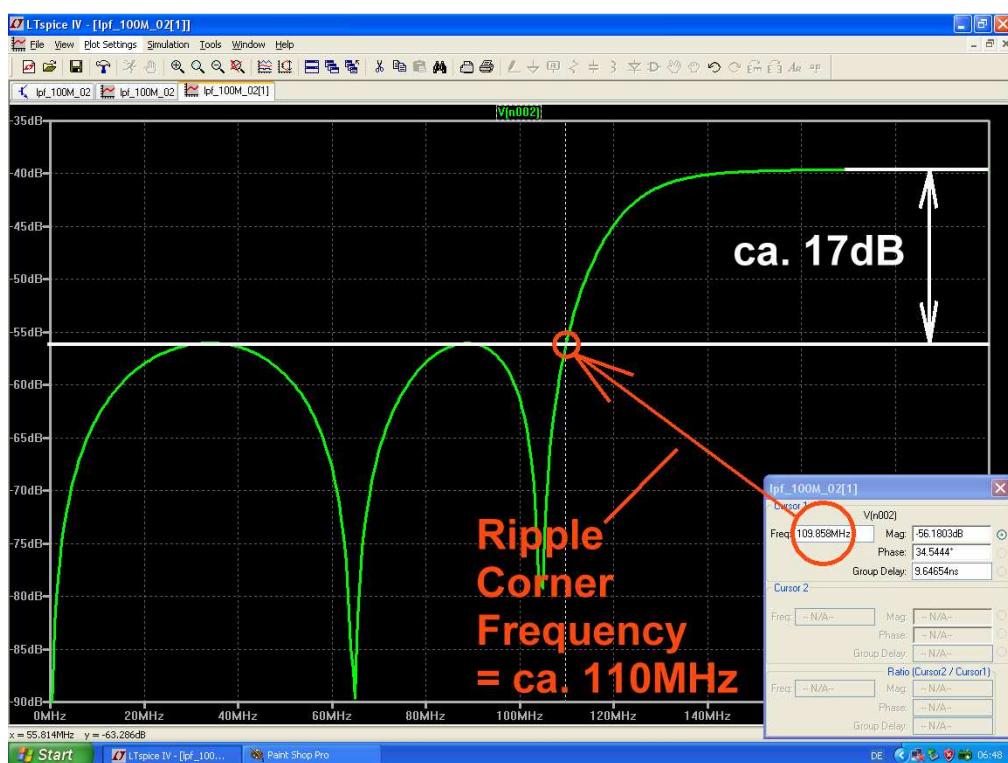


Dann geht es an die Einstellungen für die FFT. Verwendet wird der Zeitbereich ab 20 Picosekunden bis 1μs und die maximal mögliche Anzahl von 524288 Samples.



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:

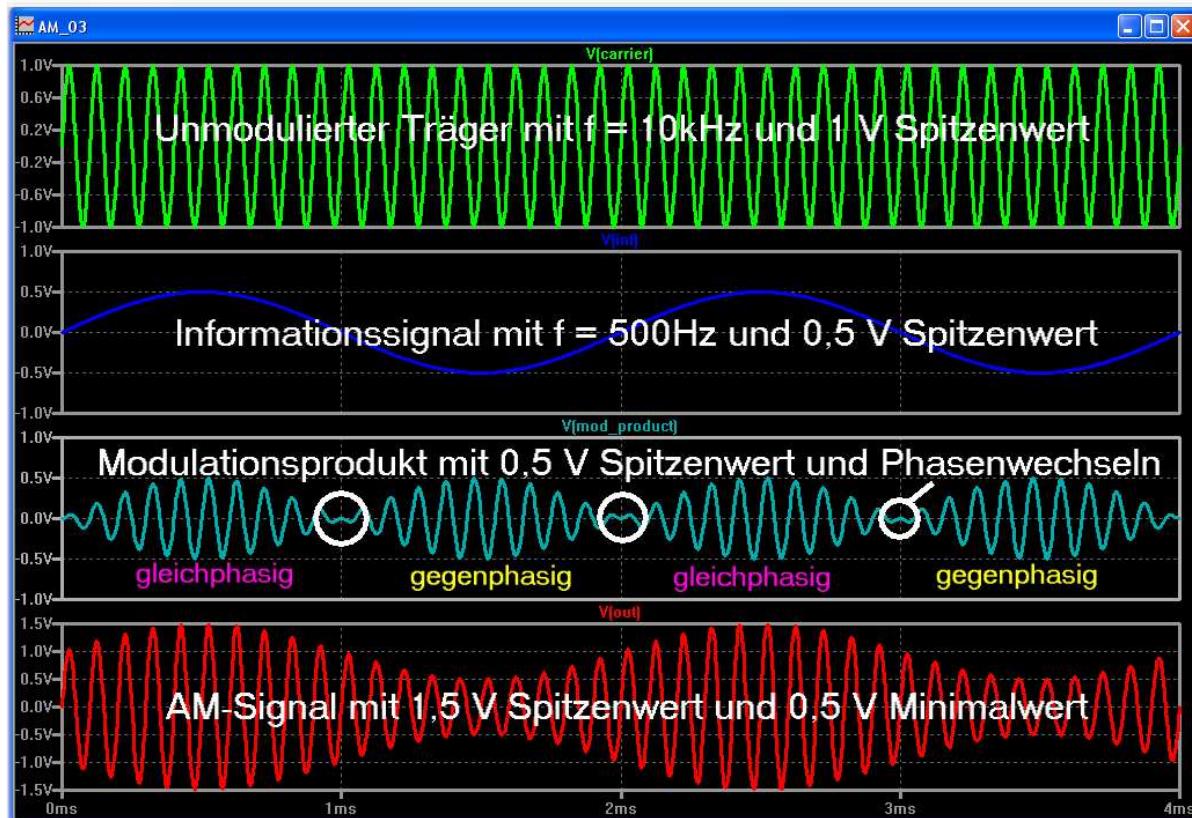


Da brauchen wir uns vor der S-Parameter-Simulation in Kapitel 14.2. nicht zu verstecken.

21. Modulation

21.1. 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 $f = 10\text{kHz}$ und einem Spitzenwert von 1V zu sehen, Darunter folgt die Information als Sinus mit 500Hz und $0,5\text{V}$ Spitzenwert. Wenn wir nun diese beiden Signale miteinander **multiplizieren**, dann muss das Ergebnis ein Signal mit Trägerfrequenz sein, das

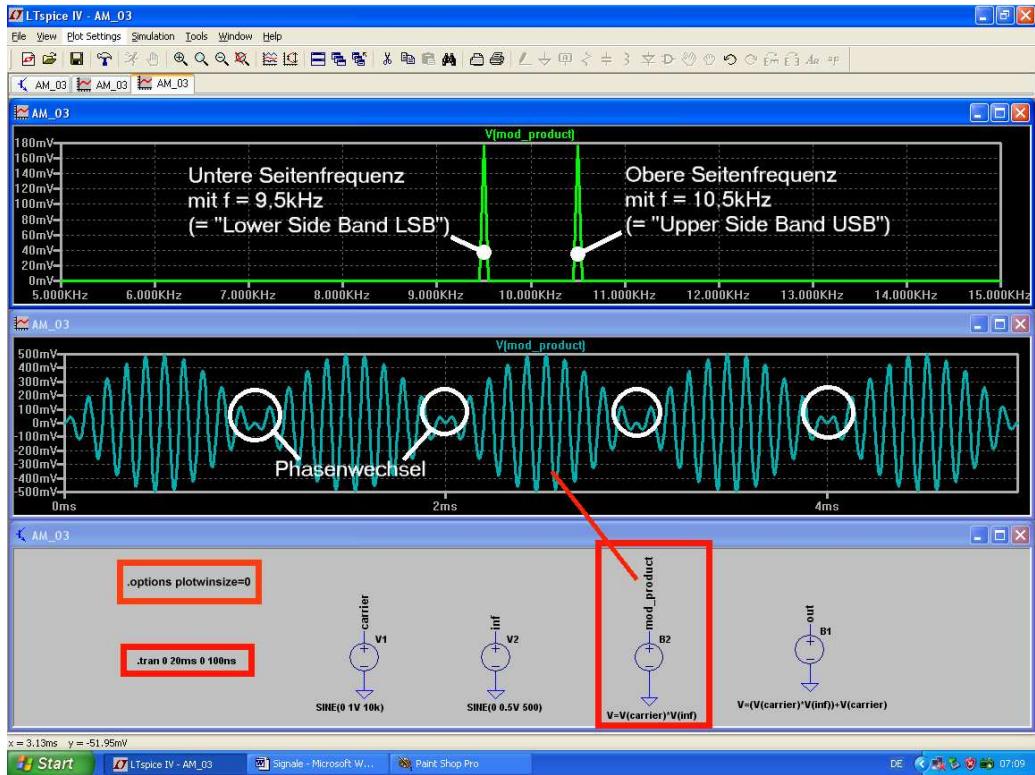
- a) dem Amplitudenverlauf der Information folgt, aber
- b) jedes Mal die Phase umkehrt, wenn die Informationsspannung negativ ist.

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:

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

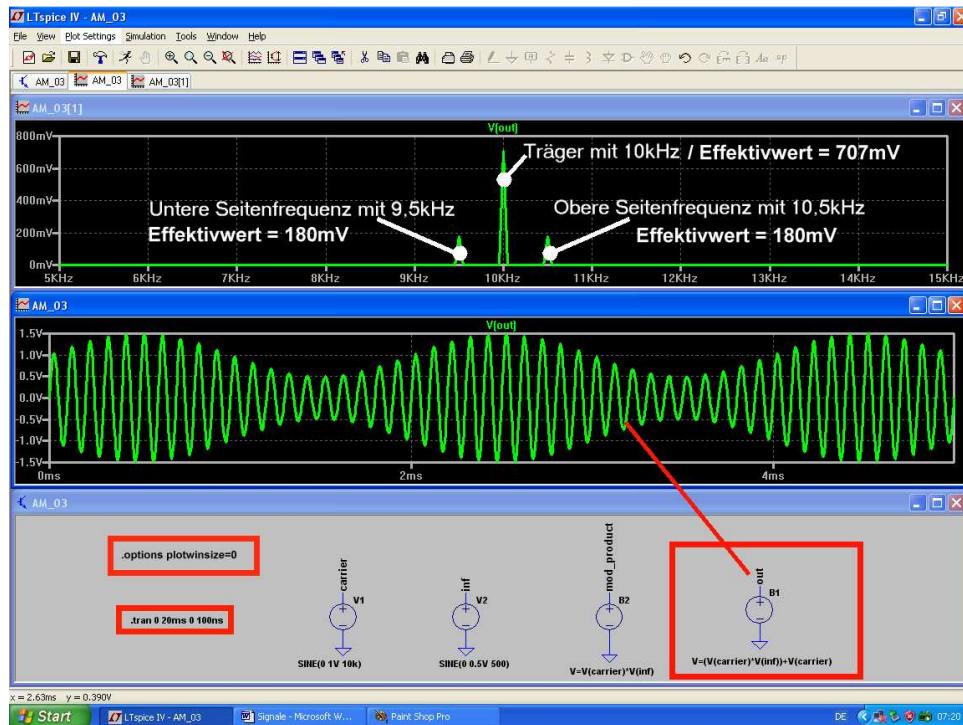
Wir erhalten als Produkt die Summen- und die Differenzfrequenz der beiden beteiligten Cosinussignale. Träger und Information selbst tauchen NICHT MEHR darin auf!



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 $10\text{kHz} + 0,5\text{kHz} = 10,5\text{kHz}$
- b) die **Differenzfrequenz**, also $10\text{kHz} - 0,5\text{kHz} = 9,5\text{kHz}$

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.

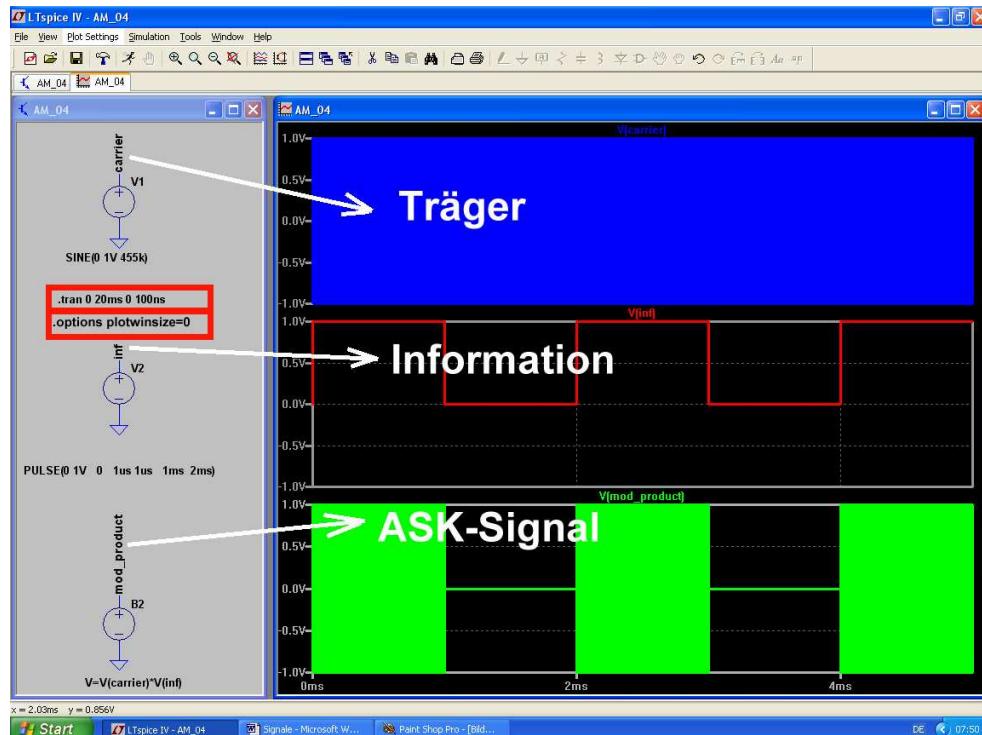


Beim kompletten AM-Signal wird noch das Trägersignal dazuaddiert und das erledigen wir mit einer weiteren „bv-Quelle“. Auch das ist hier gut zu sehen.

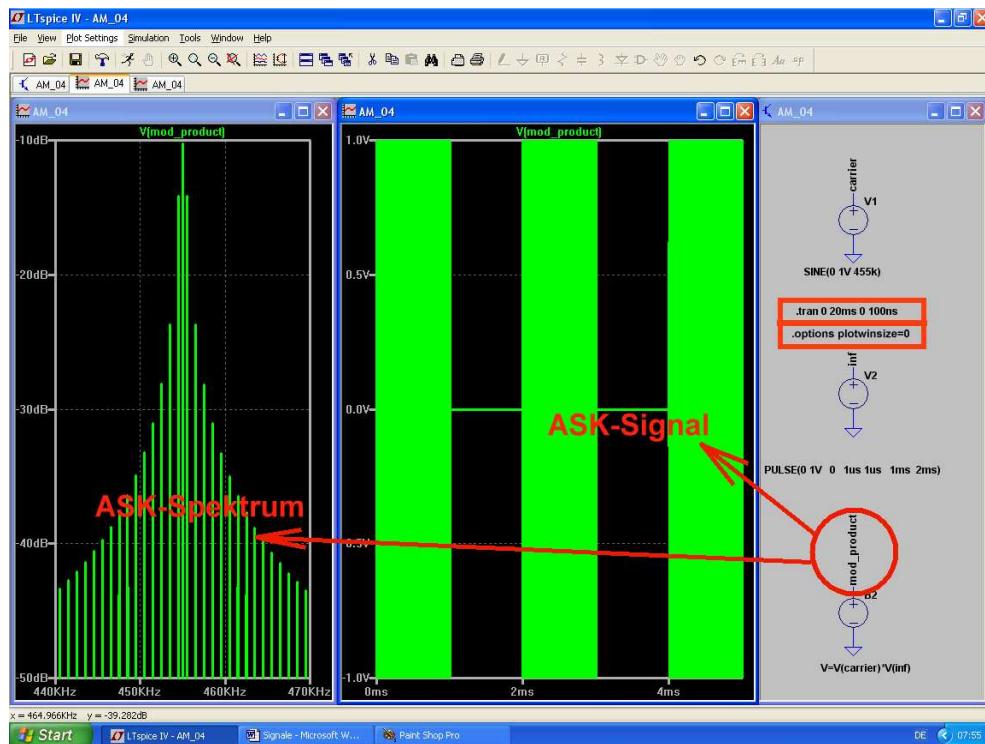
Abschließende Anmerkung:
Meist sind bei der Information mehrere Frequenzen oder ein komplettes Frequenzband beteiligt. Deshalb spricht man in diesen Fällen vom **Unteren und vom Oberen Seitenband, also vom Lower Side Band LSB und Upper Side Band USB** in einem AM-Signal.

21.2. 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 benutzen 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.



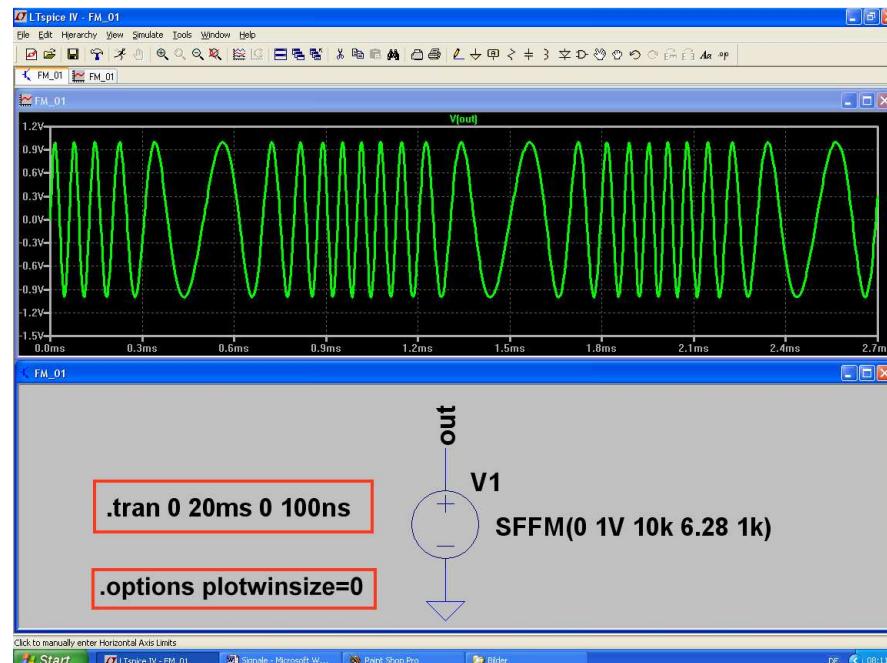
Nun folgt das Spektrum des ASK-Signals und was man da sieht, war zu erwarten. Die symmetrische rechteckförmige Information enthält nämlich außer der Grundwelle noch alle ungeradzahligen Oberwellen von 1kHz und das Ergebnis sind folglich ein LSB bzw. USB mit all diesen Spektralanteilen der Information, die sich nach links und nach rechts vom Träger ausbreiten.

21.3. 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 Informationsspannung**. 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:

- 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.
- 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.
- Und endlich gibt es im gleichen Unterordner noch den FM-AM-Generator „modulate2“, der Quadratursignale besitzt (= zwei getrennte Ausgänge, an denen um 90 Grad verschobene Trägersignale zur Verfügung stehen).

21.3.1. 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)

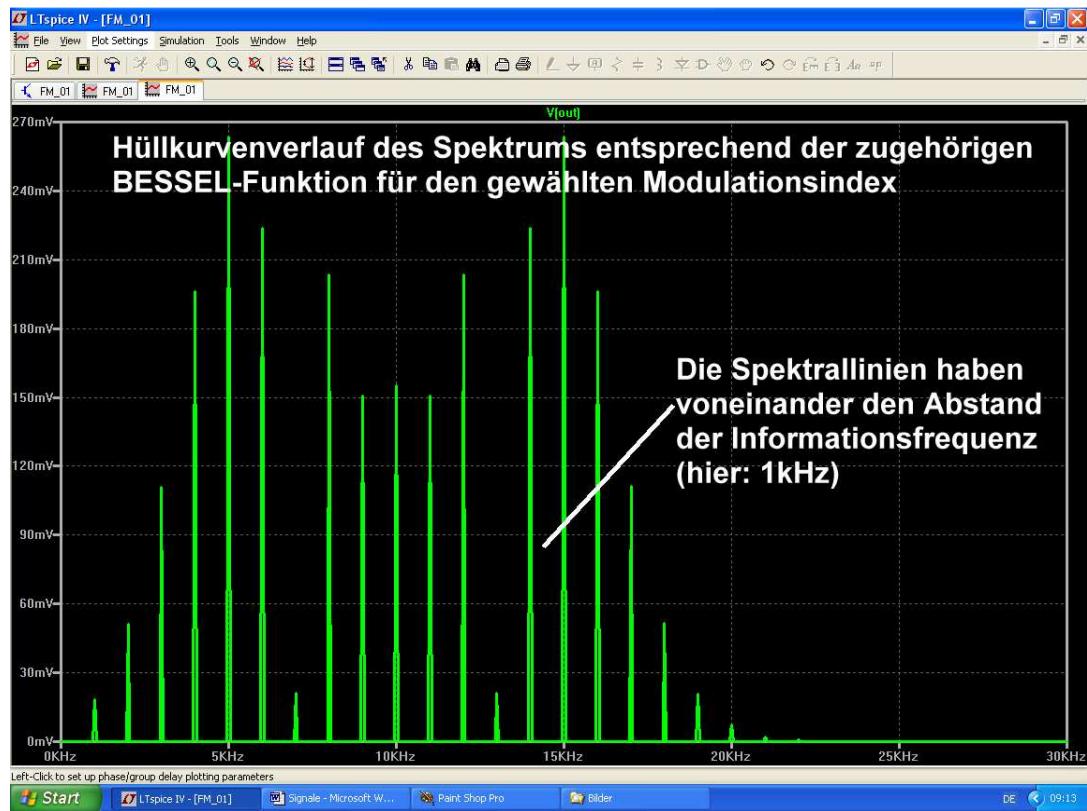
und das bedeutet nacheinander:

0	= kein DC-Anteil
1V	= Spitzenwert des Trägers
10kHz	= unmodulierte Trägerfrequenz
6,28	= Modulationsindex
1kHz	= sinusförmige Information mit $f = 1\text{kHz}$

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“ benutzt 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:

$$\text{Modulationsindex} = \frac{\text{Frequenzhub}}{\text{Informationsfrequenz}}$$

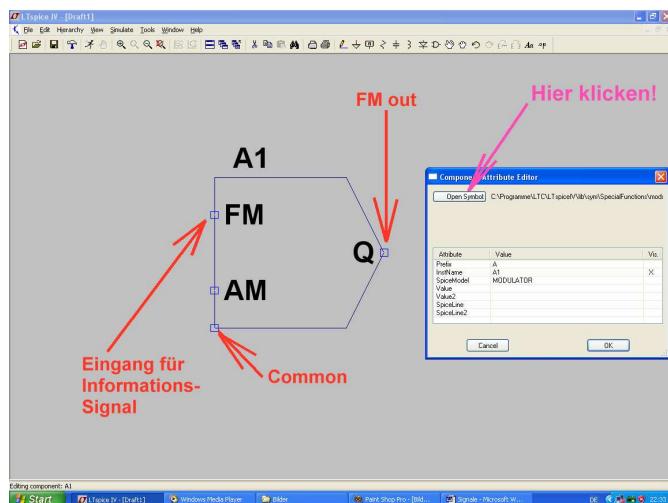
Sieht man sich diese Formel genauer an, dann erkennt man sofort den Unterschied zwischen den beiden Betriebsarten der **Frequenzmodulation** und der **Phasenmodulation**. Es gilt nämlich:

- 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.

21.3.2. 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**

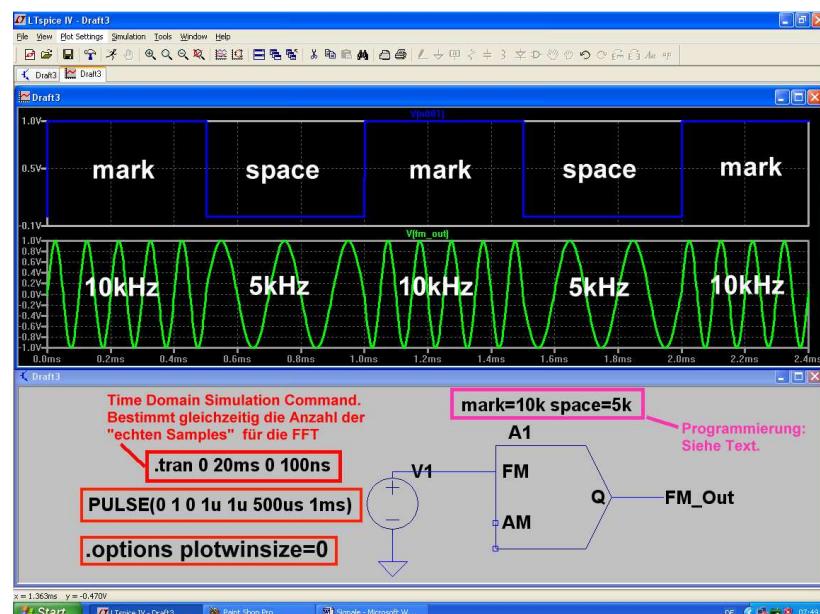
rein, denn so legen wir die Steuerkennlinie des VCO's fest:

Damit erhalten wir mit einer Spannung von Null Volt am FM-Eingang die Ausgangsfrequenz „space“ von 5kHz.

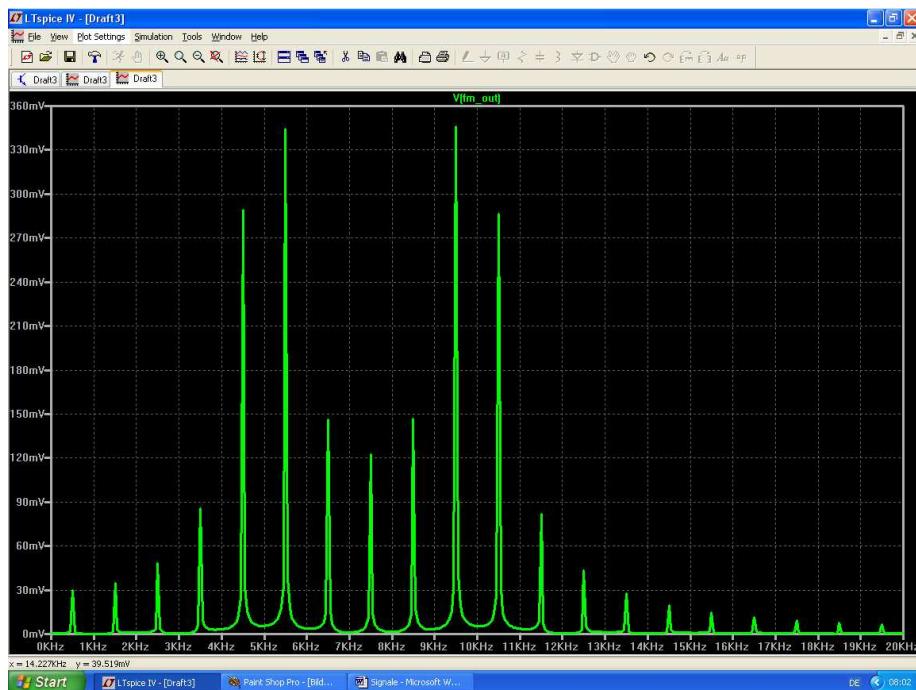
Mit einer Spannung von 1V an FM dagegen wird die „mark“-Frequenz von 10kHz erzeugt.

(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 ($U_{min} = 0V / U_{max} = 1V$) 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:

- Verändern Sie am Informationseingang die minimale und maximale Pulsamplitude. Prüfen Sie, ob auch negative Spannungswerte zulässig sind.
- 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.
- Benützen Sie als Information eine Dreieckspannung mit derselben Frequenz von 1kHz.

21.4. 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!

21.4.1. 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 **einiger 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 hilft 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.

21.4.2. 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 $f = 1000\text{Hz}$ und einem Spitzenwert von $0,5\text{V}$.

Signal B (im Q-Kanal) ist ein Rechteck mit $f = 500\text{Hz}$, einem Minimalwert von 0V und einem Maximalwert von 1V .

Die beiden Trägerkomponenten (= I bzw. Q) haben Rechteckform. Ihre Frequenz beträgt $f = 20\text{kHz}$, der Minimalwert sei -1V , der Maximalwert dagegen $+1\text{V}$. 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 ($f = 20\text{kHz}$) und einer Amplitude von 1V (kein DC-Anteil), aber einem Phasenunterschied von 90° . 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 $12,5\text{ Mikrosekunden}$ bei $f = 20\text{kHz}$ 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 **$t = 20\text{ms}$ mit einem maximalen Timestep von 20ns** . Dann wird der Button mit dem Häckerchen gedrückt, auf die Karteikarte „SPICE“ gewechselt und dort „Gear“ unter „Default Integration Method“ aktiviert (...erhöht die Stabilität der Simulation).

Das gibt (mit $t = 20\text{ms}$) eine **Frequenzauflösung von $1 / 20\text{ms} = 50\text{Hz}$** , 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

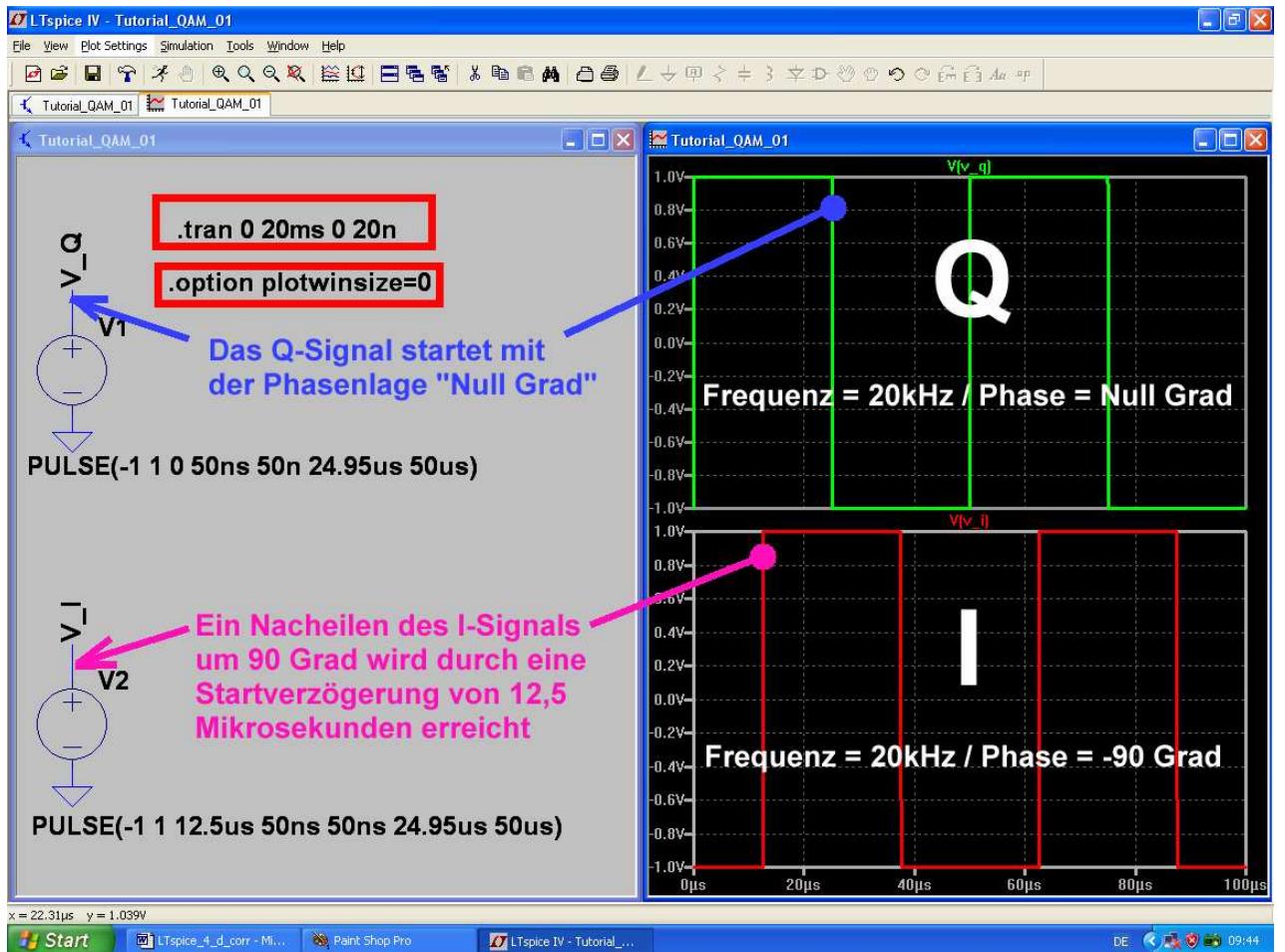
$20\text{ms} / 20\text{ns} = 1\,000\,000\text{ Samples}$

für die FFT zur Verfügung.

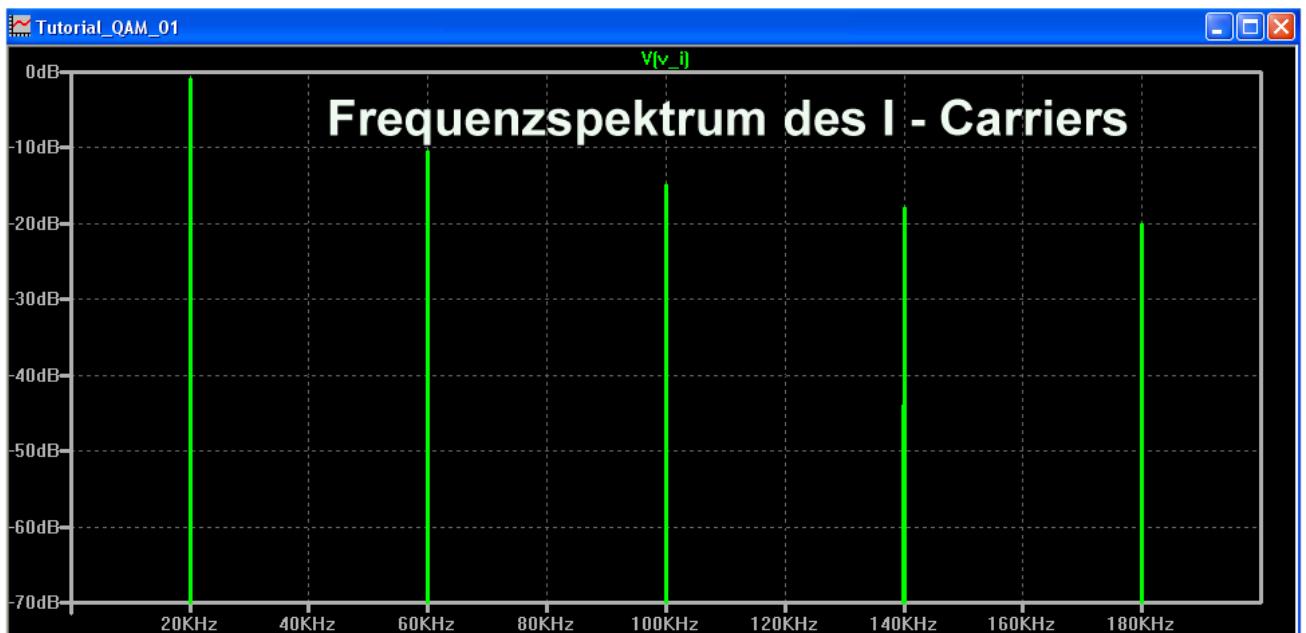
Wer sich über die „**24,95μs**“ statt „**25μs**“ bei der Pulslänge wundert, der möge bedenken:

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 $20\text{ms} / 20\text{ns} = 1000\,000$ 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:

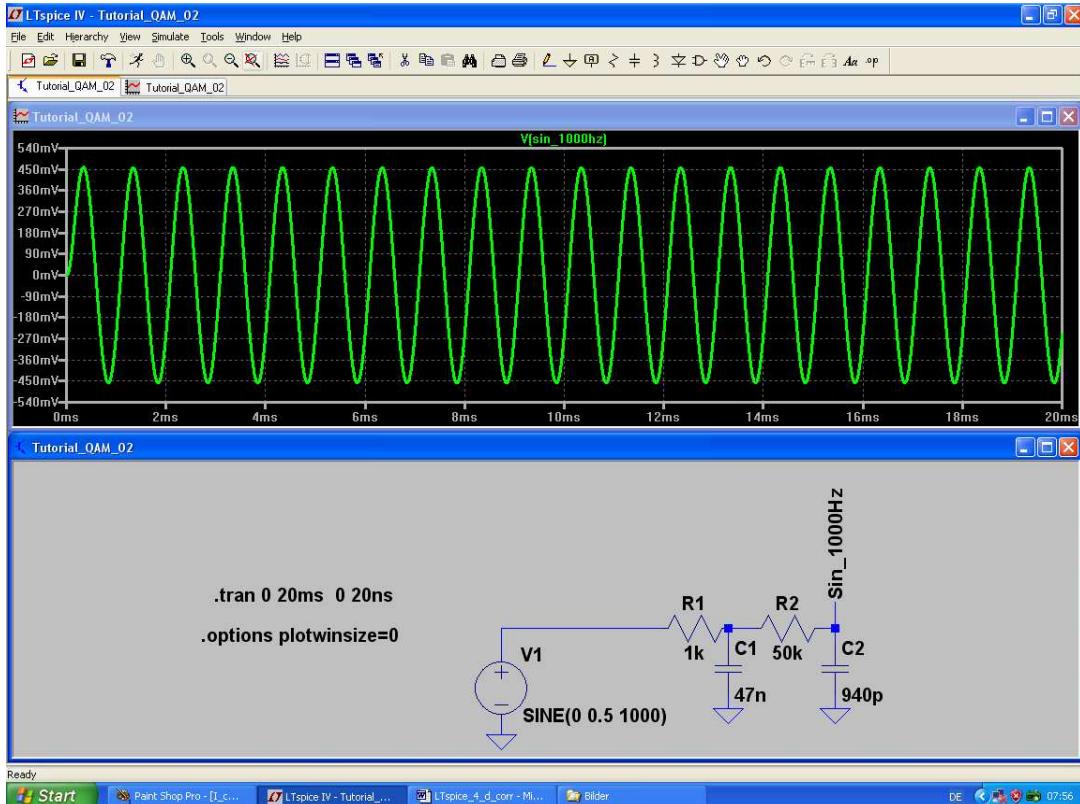


Dazu das Spektrum nach einer FFT mit 524288 Samples (Anleitung: mit der rechten Maustaste auf den Spannungsverlauf klicken, dann „View“ und „FFT“ wählen).

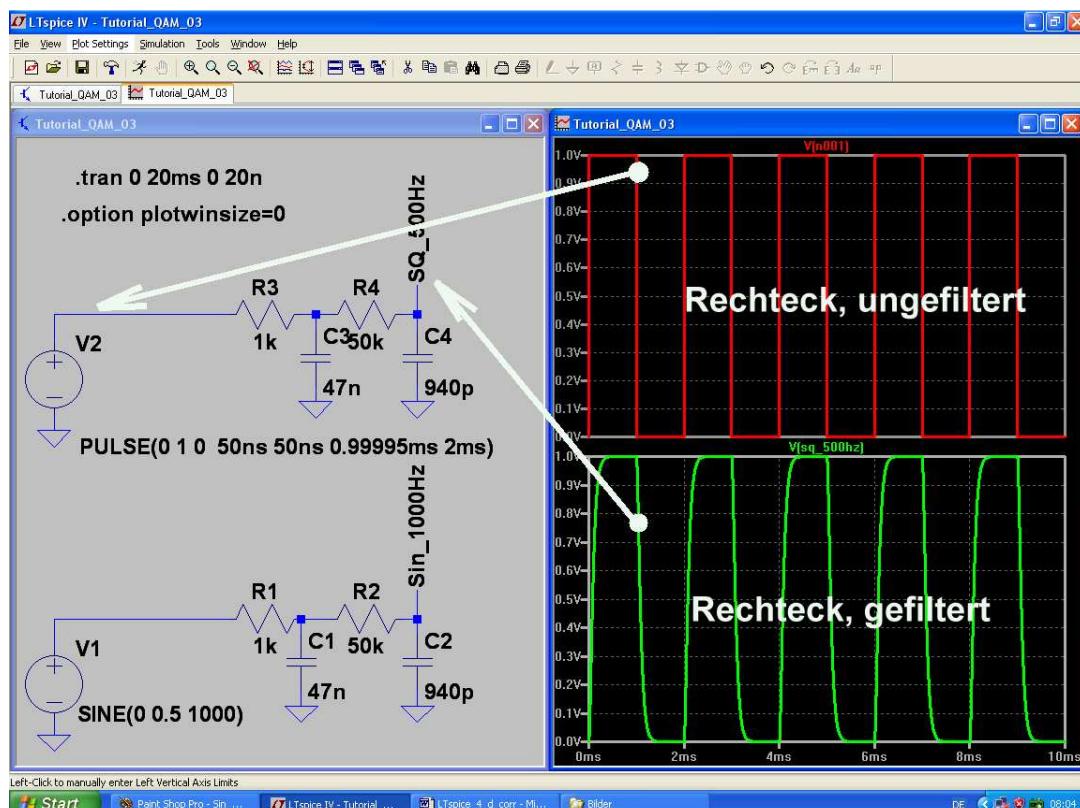


Man sieht, dass nur die ungeradzahligen Harmonischen auftreten und alles Unerwünschte um mehr als 70db gedämpft ist. Das ist ein sicheres Zeichen für nahezu perfekte Symmetrie des Rechtecksignals.

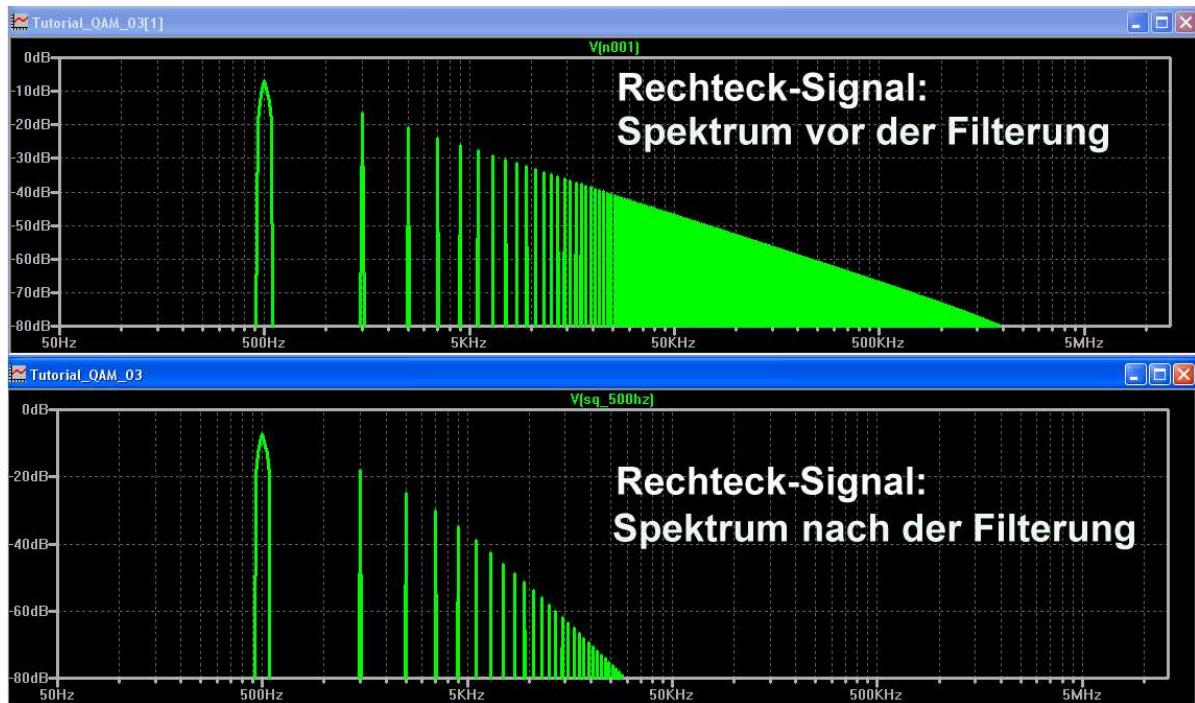
Jetzt programmieren wir die „Basisband-Signale“. Dazu setzen wir zwei weitere Spannungsquellen ein und schalten hinter jede einen einfachen Tiefpass mit einer Grenzfrequenz von ca. 3,4 kHz zur „Bandbegrenzung“ (...das entspricht der oberen Grenzfrequenz eines Telefonkanals). So vermeiden wir Probleme durch Frequenzband-Überschneidungen (Aliasing) bei der nachfolgenden Trägermodulation. **Beginnen wir mit dem Sinussignal ($f = 1000\text{Hz}$ / Spitzenwert = 0,5V) im I – Kanal:**



Im gleichen 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:

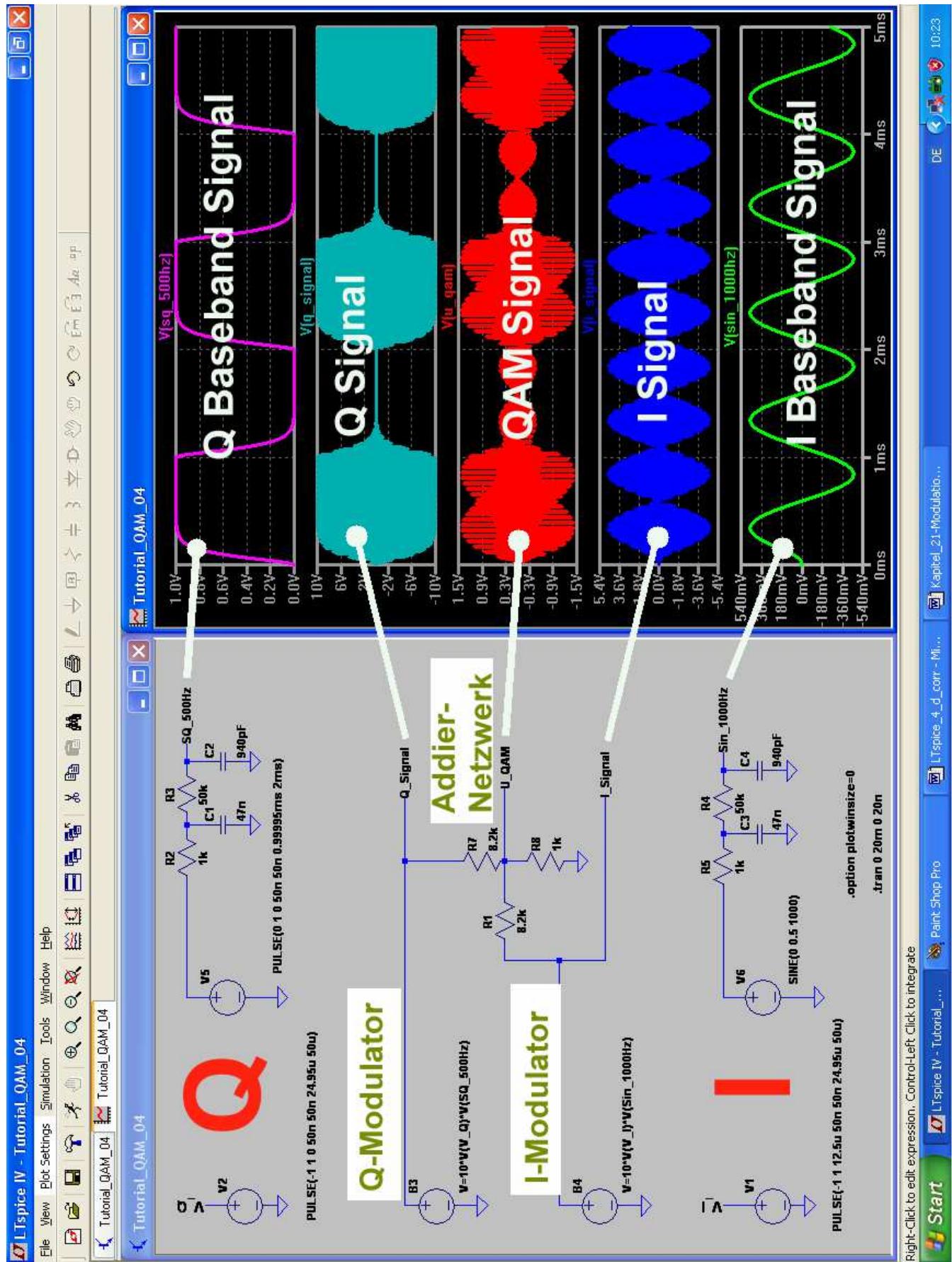
Die „bv“ – Spannungsquelle im Bauteilvorrat von LTspice stellt einen idealen **Analog-Multiplizierer** dar, mit der sich jede Trägerkomponente und ihr Basisband-Signal perfekt multiplizieren lassen. Damit verwirklicht man bei jeder Diagrammachse die gewünschte **Übertragung des Basisband-Signalverlaufs 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.

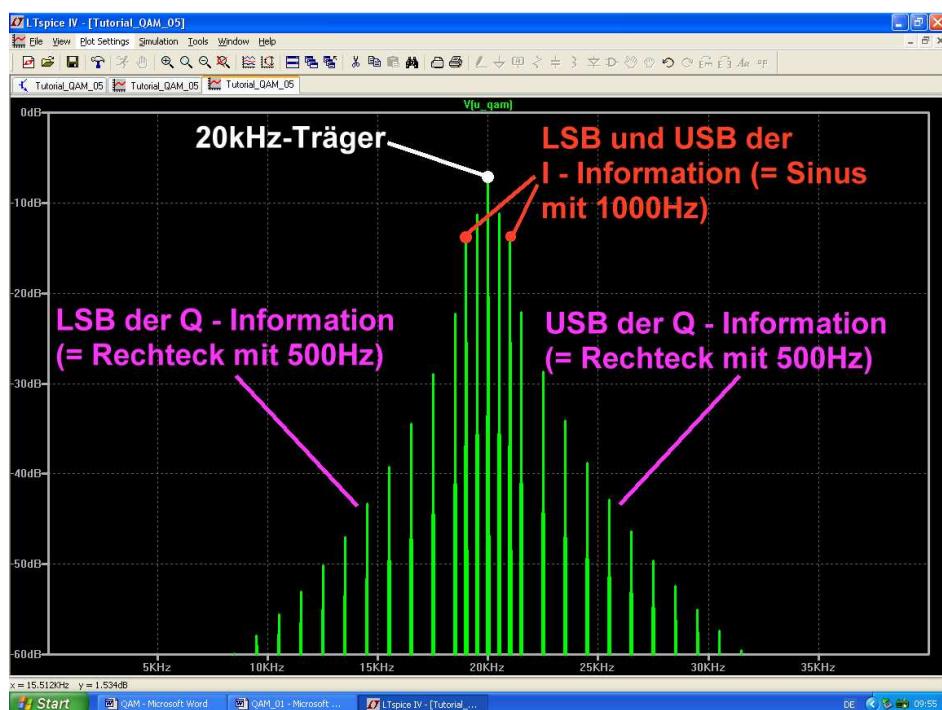
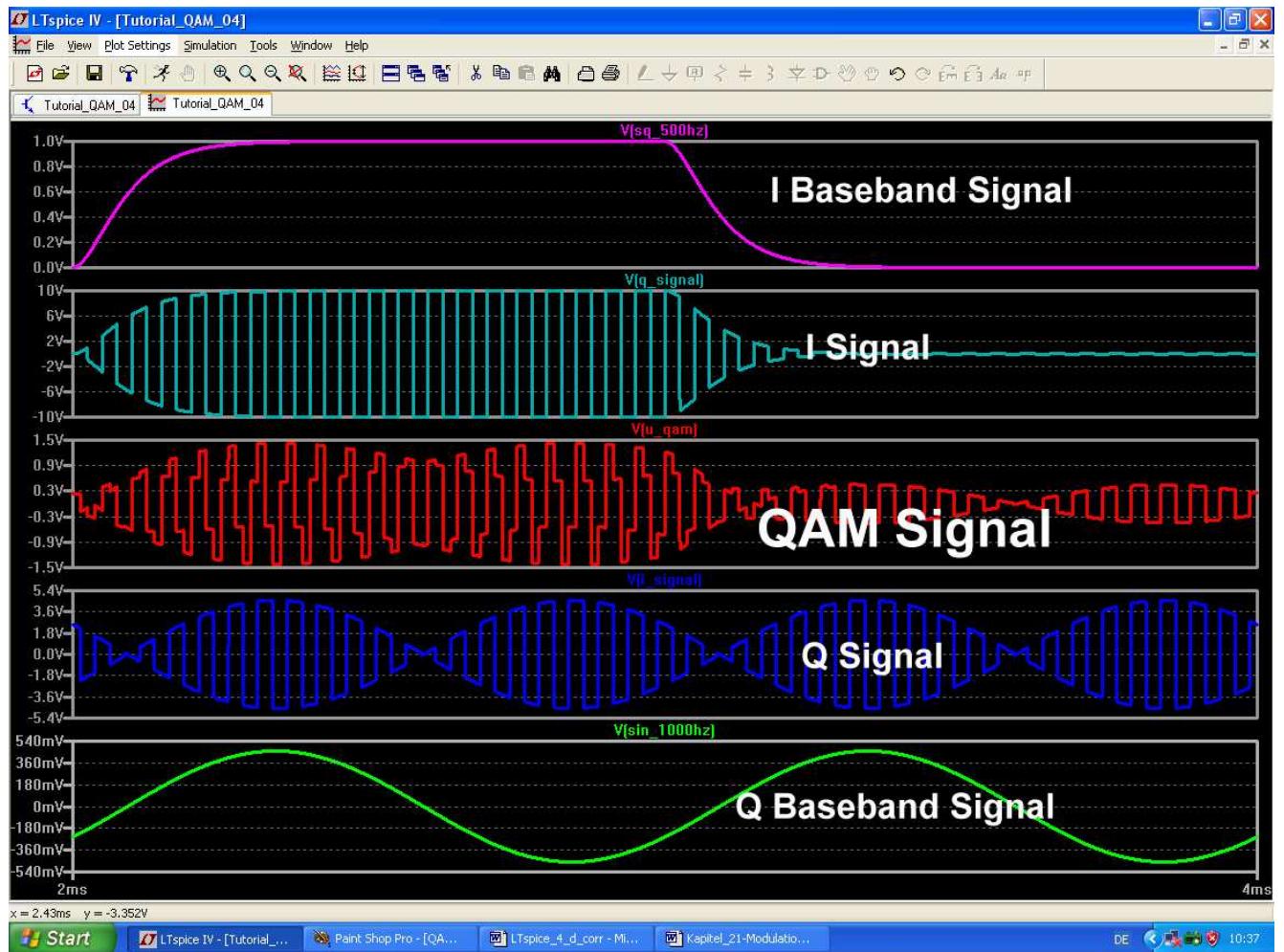
Die erforderlichen Formel-Eingaben bei den beiden Multiplizier-Stufen lauten deshalb

für den „I“ – Pfad: $V = 10 * V(V_I) * V(\text{Sin_}1000\text{Hz})$

und für den “Q” – Pfad: $V = 10 * V(V_Q) * V(\text{SQ_}500\text{Hz})$

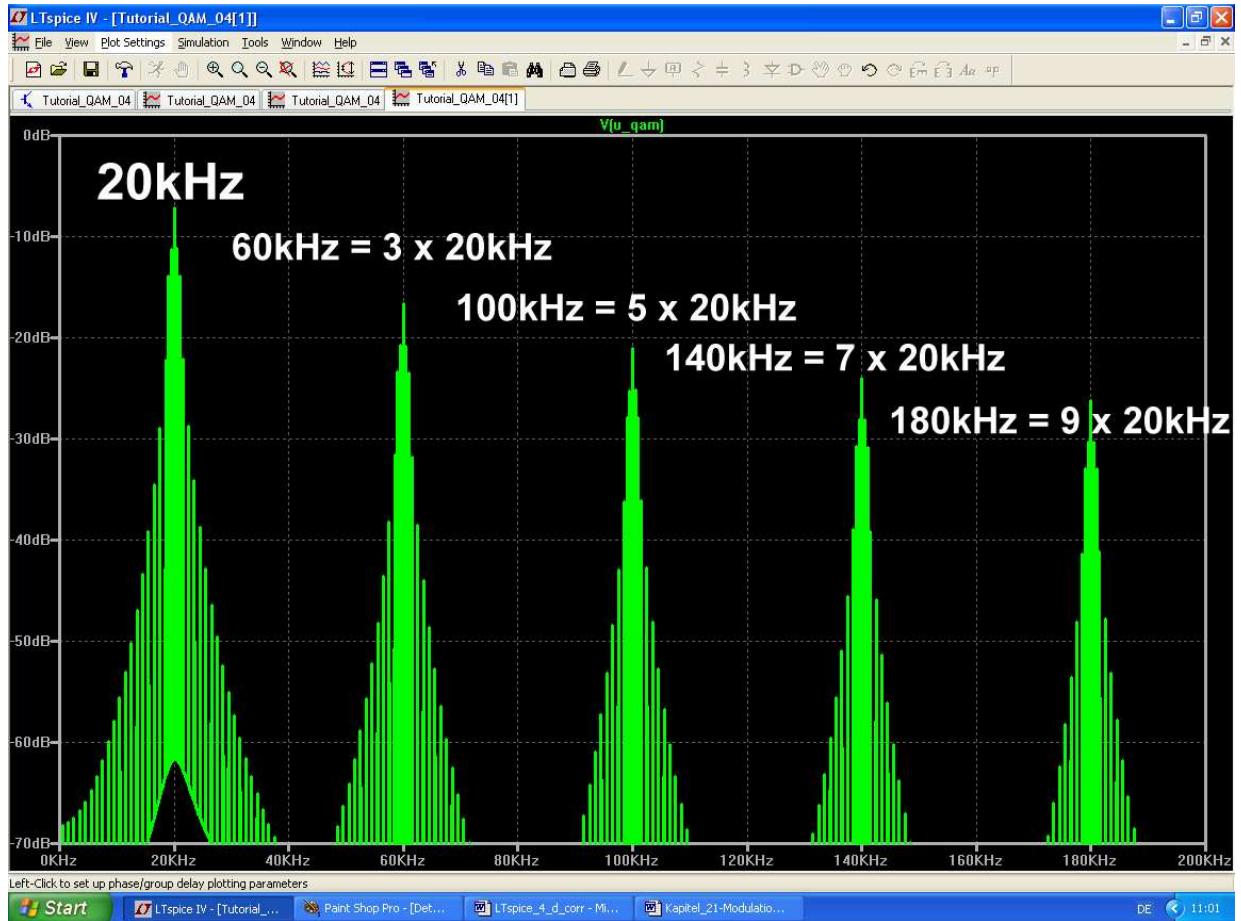


Interessant ist ein Blick auf die Details nach starkem Zoomen der Diagramme. Bei beiden Achssignalen (I-Signal und Q-Signal) ist der Phasenwechsel im Rhythmus der Trägerfrequenz von 20kHz gut zu sehen. Und nach dem Zusammenaddieren der beiden Komponenten wird im QAM-Signal deutlich auch deren Phasenunterschied 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...

21.4.3. 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 $6,8\text{kHz}$ und $R_i = R_a = 50\Omega$** speisen.

Der Bandpass-Entwurf ist mit den heutigen Programmen aus dem Internet ein Kinderspiel und eine Freude. Die Auswahl ist groß, aber hier wurde (wohl aus Gewohnheit und weil es auf andere Weise auch nicht flotter geht!) mit einem alten 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.....

```
Filter Requirements

Use arrows to select the desired option and hit return
LowPass    HighPass    BandPass    BandReject  NarrowBP
Center Freq(MHz):0.02      Poles:3          Ripple:0.3
Bandwidth:0.0068           Zsource:50
0 poles to autocalculate number of poles
Enter the value and hit return
Do you specify 3 dB or ripple width? (r/3)3
```

.....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?
```

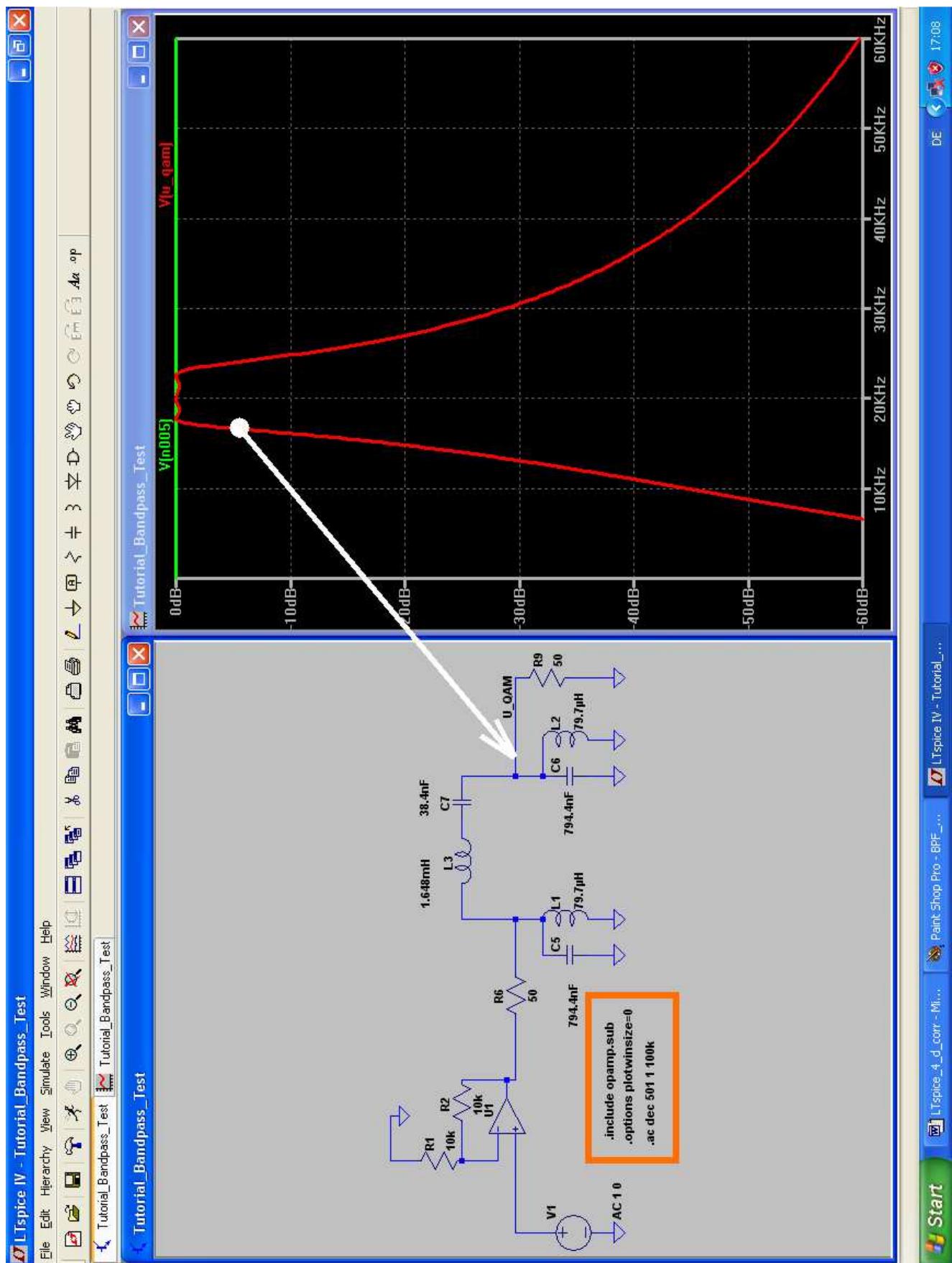
„PLC“ bedeutet „parallel LC resonant circuit“ und „SLC“ natürlich „series LC resonant circuit“...
Nun werden zunächst in LTspice eine Testschaltung aufgebaut und die Filterkurve samt Verstärkung geprüft.

Sie findet sich auf dem nächsten Blatt und folgende Dinge bedürfen einer Erklärung:

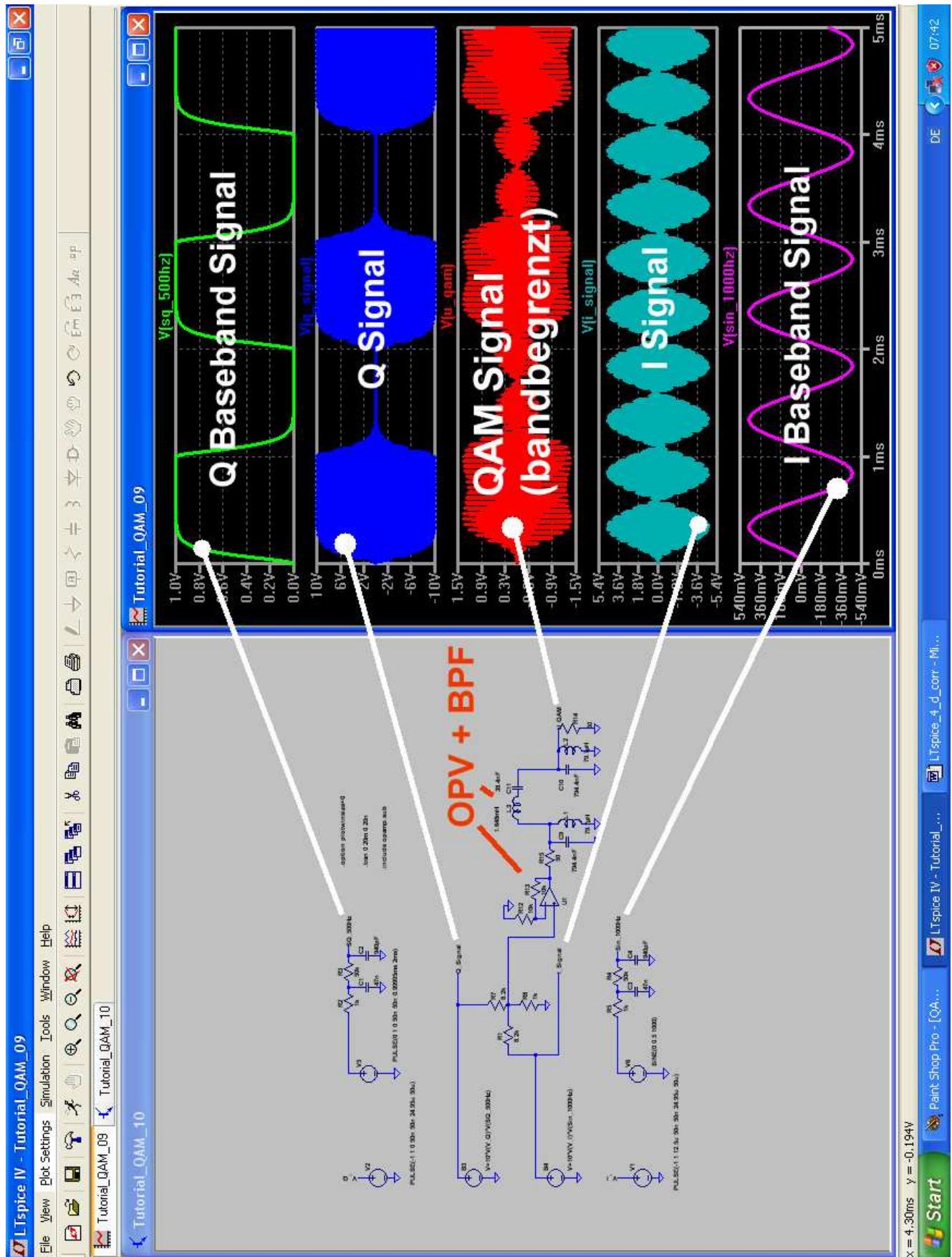
- a) **.options plotwinsize=0** bedeutet, dass die Datenkompression abgeschaltet ist.
- b) **.ac dec 501 1 100k** ergibt einen dekadischen AC-Sweep von 1Hz bis 100kHz und 501Punkten pro Dekade
- c) Für diesen AC-Sweep muss bei der Spannungsquelle V1 die **Amplitude auf „1“ und die Phase auf „0“** gestellt 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.

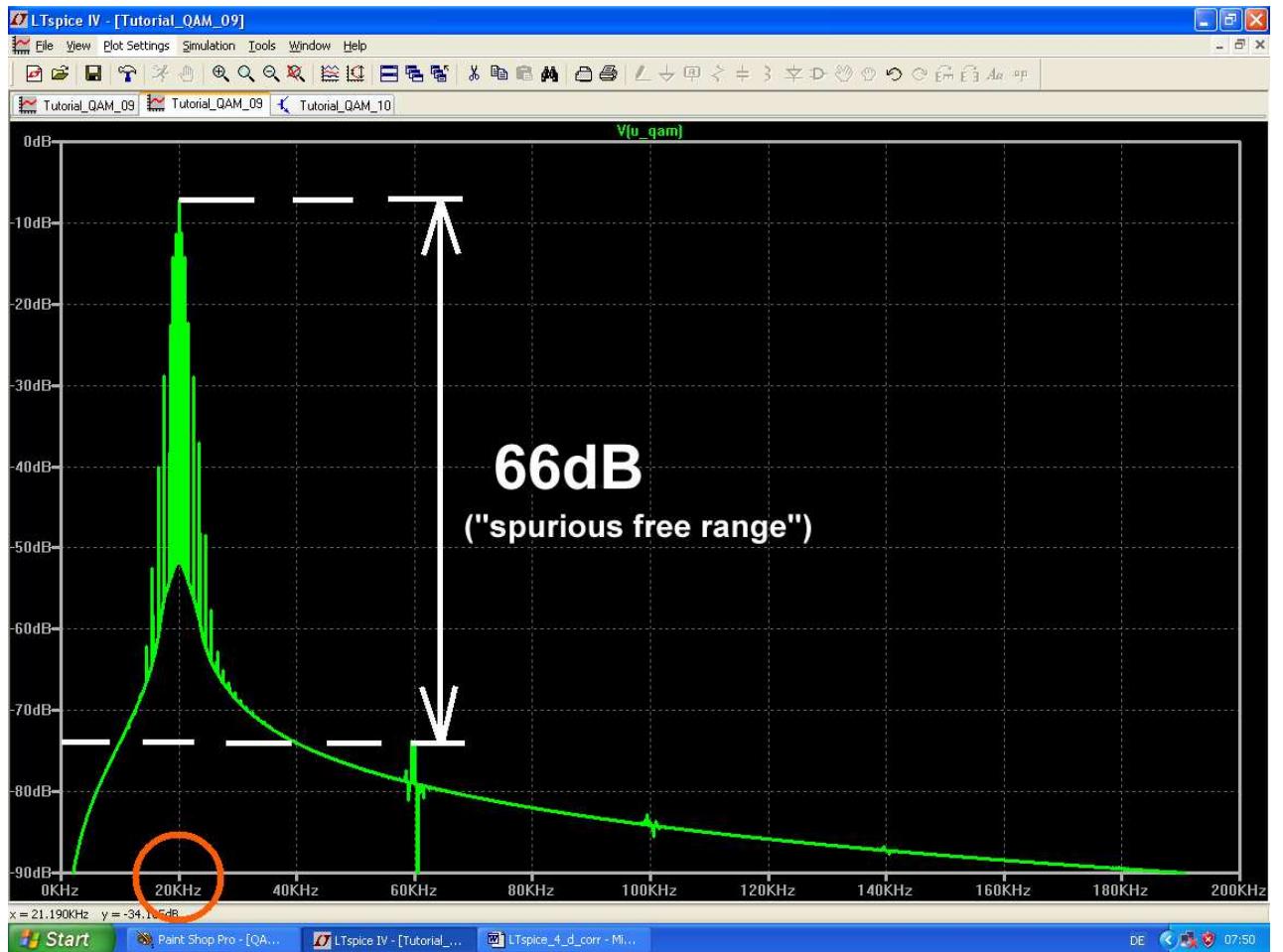
Das nächste Blatt zeigt das Ergebnis des Filtertests.



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....

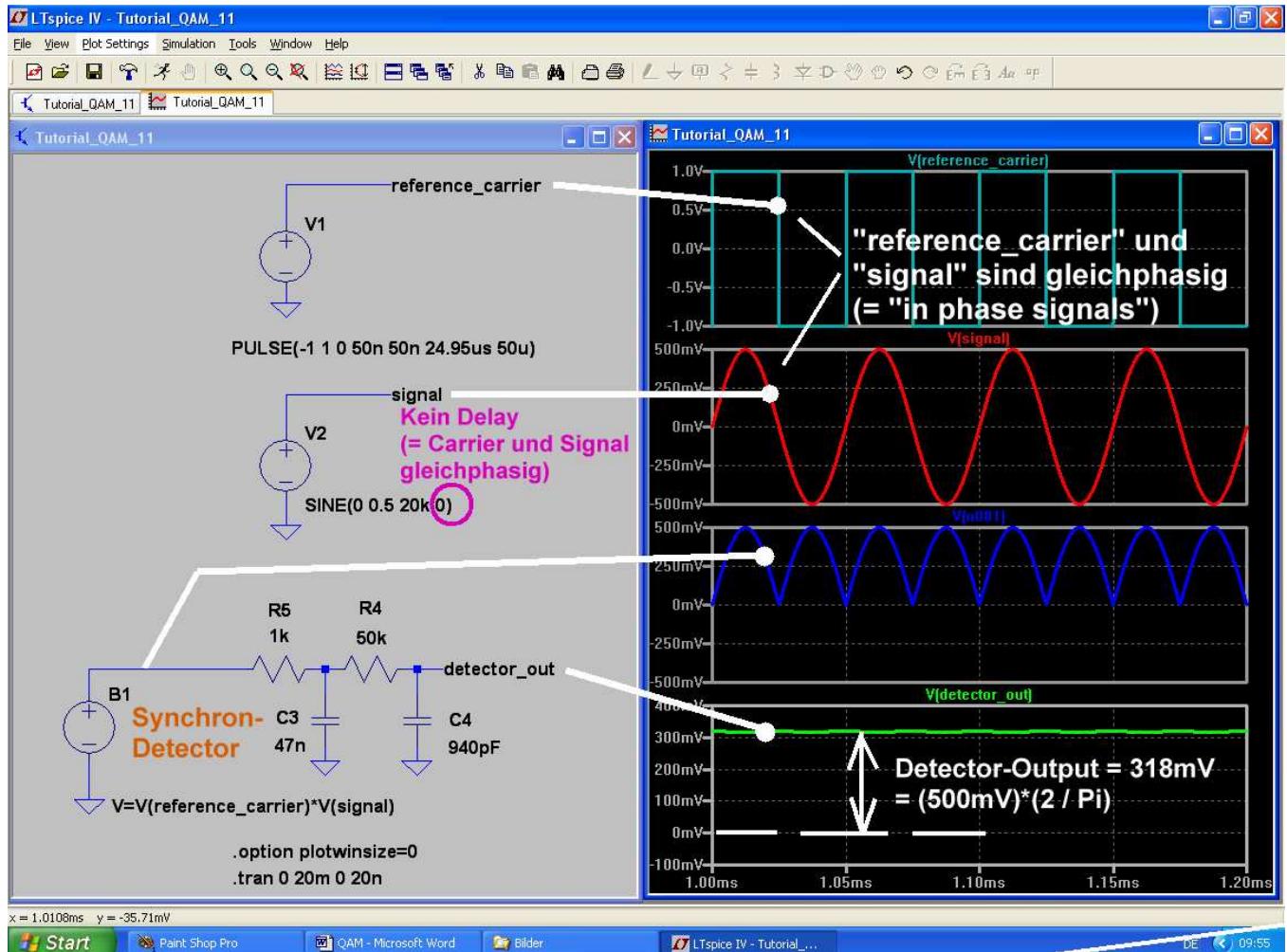
21.4.4. Demodulation des QAM-Signals

21.4.4.1. 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 benutzt 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.

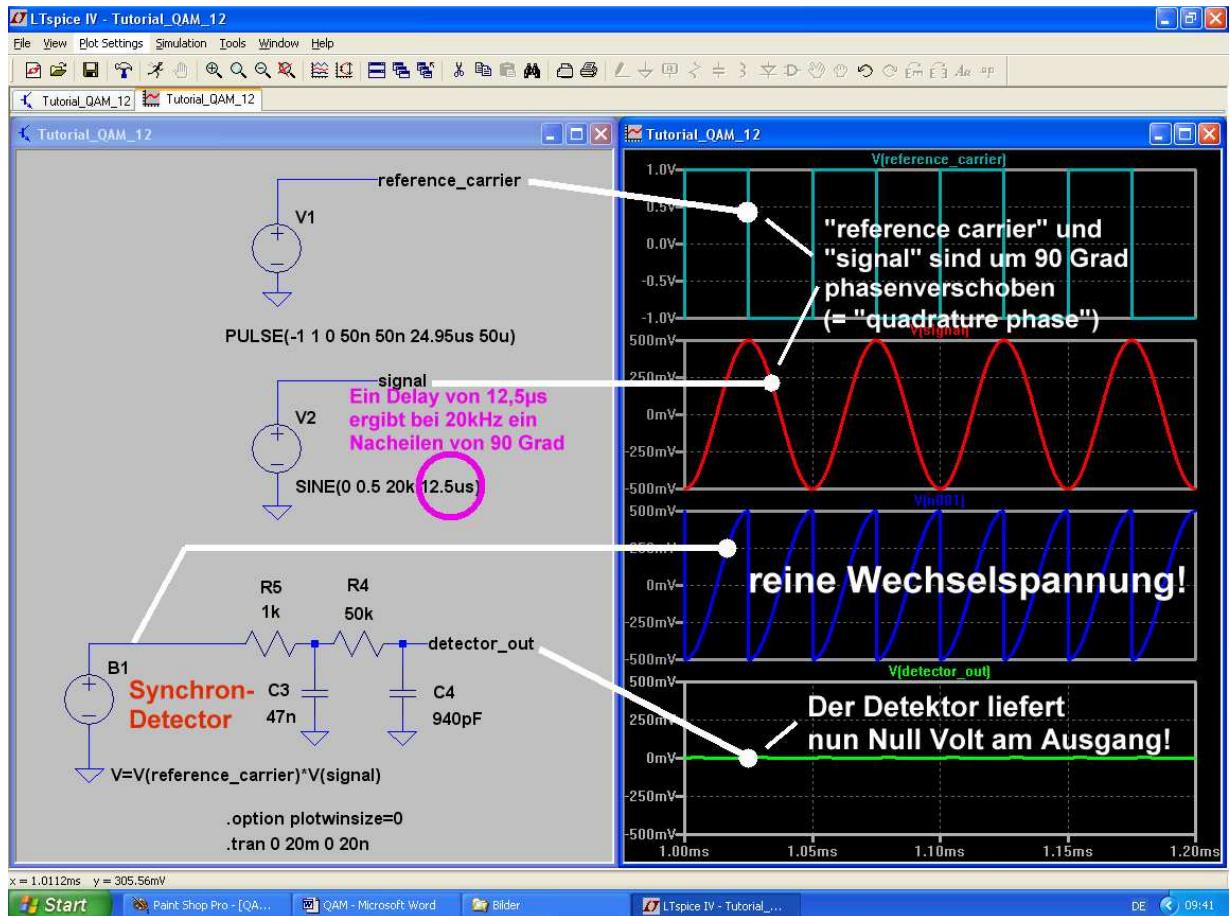
Sehen wir uns an, was bei Gleichphasigkeit beider Signale passiert:



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

$$U_{DC} = \frac{U_{peak} \cdot 2}{\pi} = \frac{500\text{mV} \cdot 2}{\pi} = 318\text{mV}$$

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).

21.4.4.2. 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.

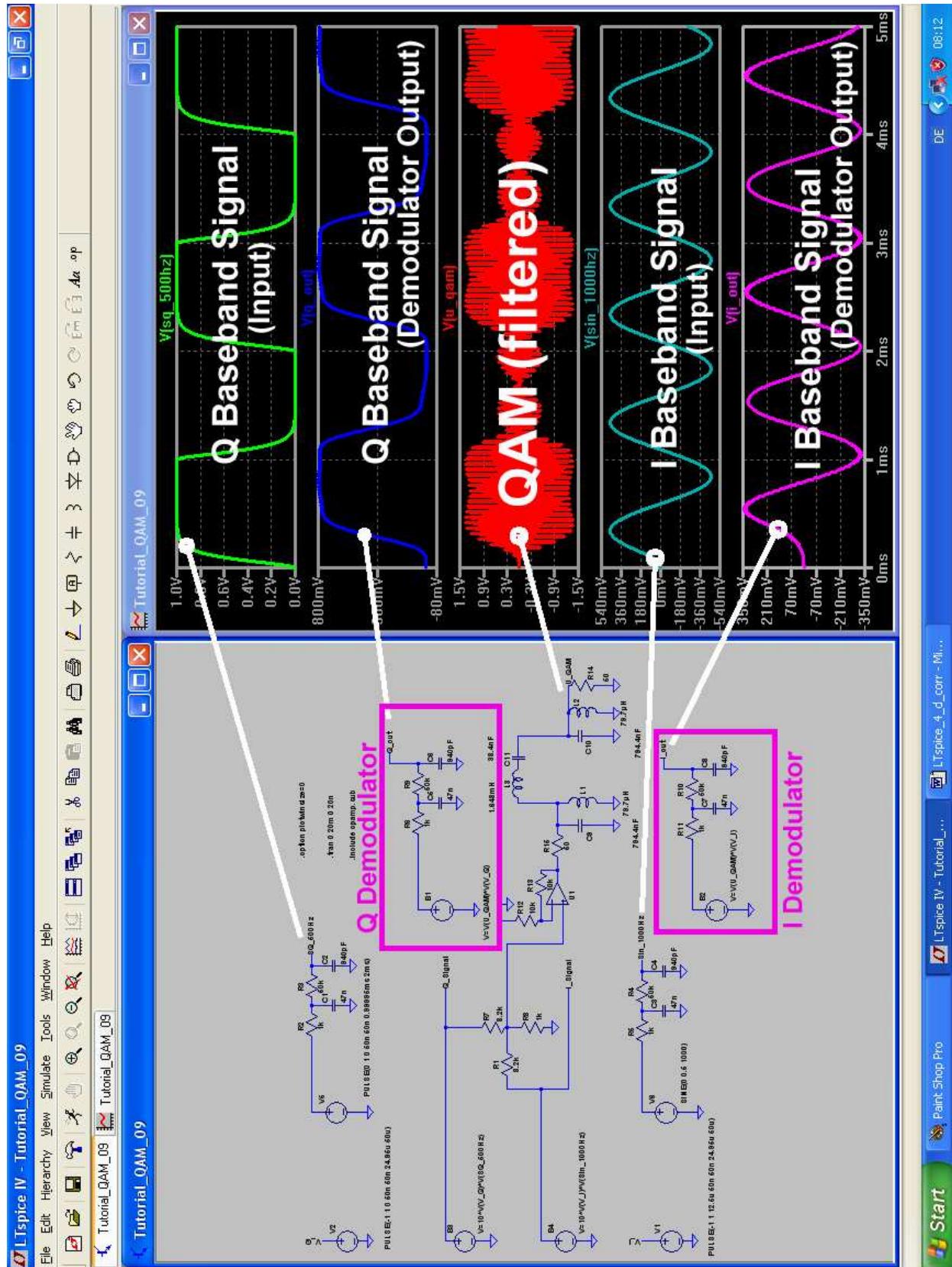
Die erforderlichen Formel-Eingaben bei den bv-Quellen lauten deshalb

$$V = V(V_{QAM}) * V(V_Q)$$

$$V = V(V_{QAM}) * V(V_I)$$

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.



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).