12. Project 8: Thyristor Circuits

12.1. Thyristor Model

A good choice for our experiments is the

2N5171

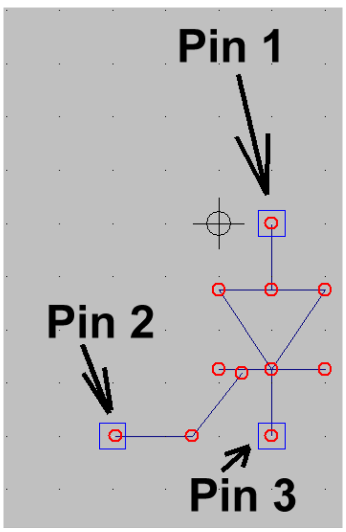
This is a 20A / 600V – thyristor which is used in many circuits. For a SPICE model search in the Internet for the file

thyristr.lib

Then save this library file under **„lib / sub"** **in the LTSpice directory**

But please note:
This library comes as an HTML-file! So open it, select all the text, copy the content to the clipboard and paste it into a new editor file as text. Save this as „thyristr.lib" in the LTspice library.

And now follow the same procedure as before:

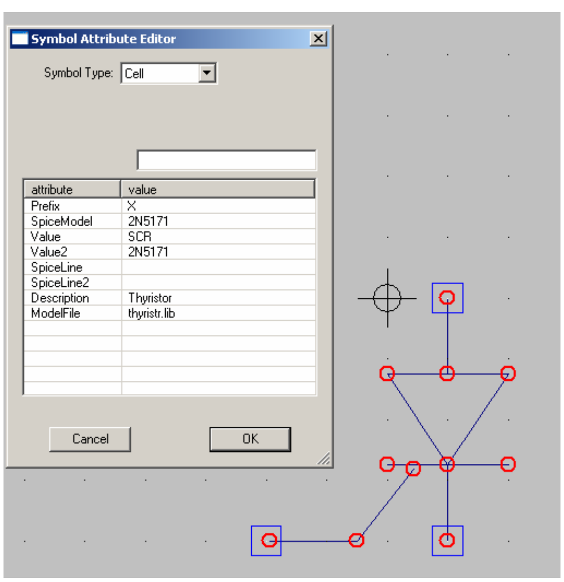


Step 1:
Open „**New Symbol**" in the file menu. Then open the **symbol for a diode („diode.asy" in folder „lib / sym).** Then draw the gate pin for a thyristor.

Warning:
The SPICE model uses the following numbering for the pins

Anode = pin 1
Gate = pin 2
Cathode = pin 3

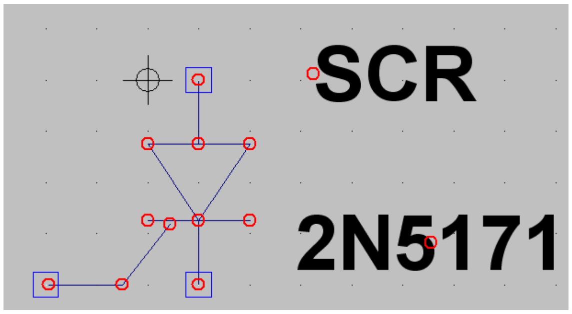
So be absolutely sure that you have the same numbering in your symbol (see left figure). If not, change it.



Step 2:
Now select from the menu bar „**Edit / Attributes**" and choose "**Edit Attributes**"

Enter in the list:

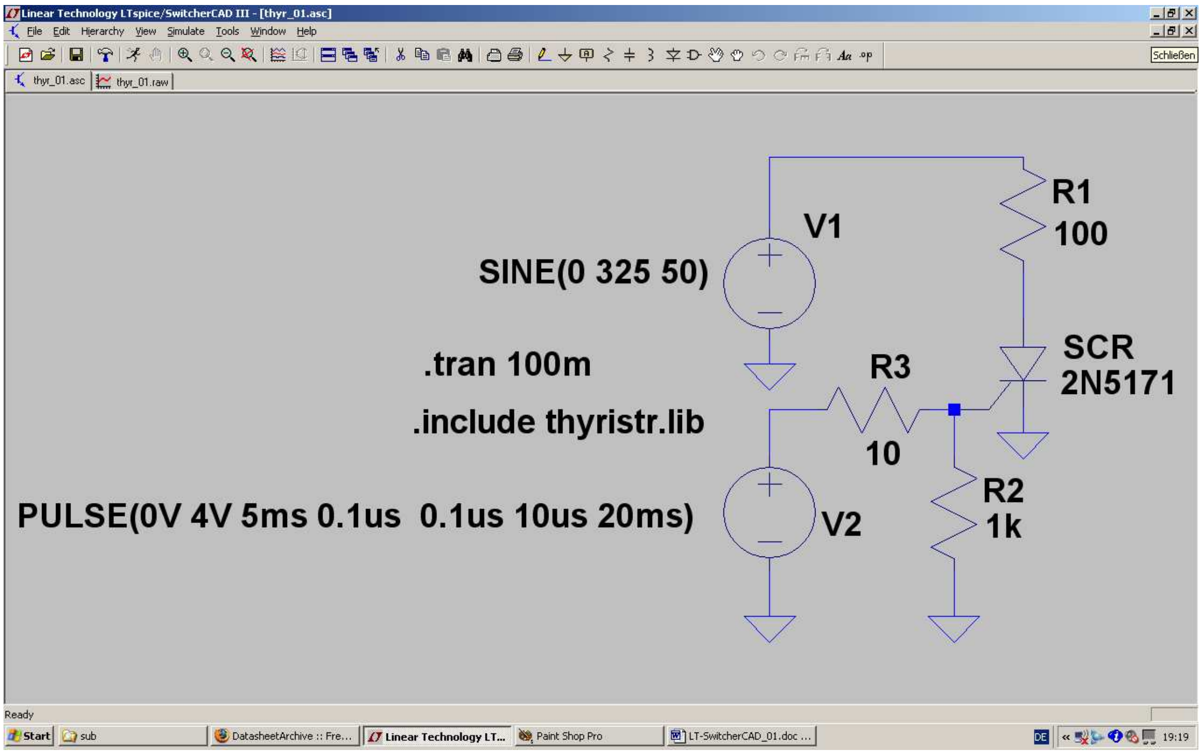
Prefix	**X**
SpiceModel	**2N5171**
Value	**SCR**
Value 2	**2N5171**
Description	**Thyristor**
ModelFile	**thyristr.lib**



Step 3:
Open „**Edit / Attributes**", but now choose „**Attribute Window**". Click on „**Value**" in the list and place "**SCR**" beside the symbol. Repeat the procedure for „**SpiceModel**". This should be the result.

Now the finished symbol must be saved in a new folder (named „**Thyristors**") in **„lib / sym"** using the name **2N5171.asy**".

12.2. Switching Resistive Loads



Description:
A sine wave source V1 (peak value = 325 V, frequency = 50Hz) feeds a series connection of a resistor R1 (100 Ω) and the thyristor 2N5171. Between the gate and the source a resistor R2 (1kΩ) is added to the schematic. A pulse source V2 is connected to the gate of the thyristor using a "**current limiting resistor R3 with 10Ω".**

Explanation of other entries on the schematic:
:

a) **„.tran 100m"** gives a simulation time from 0.....100ms.

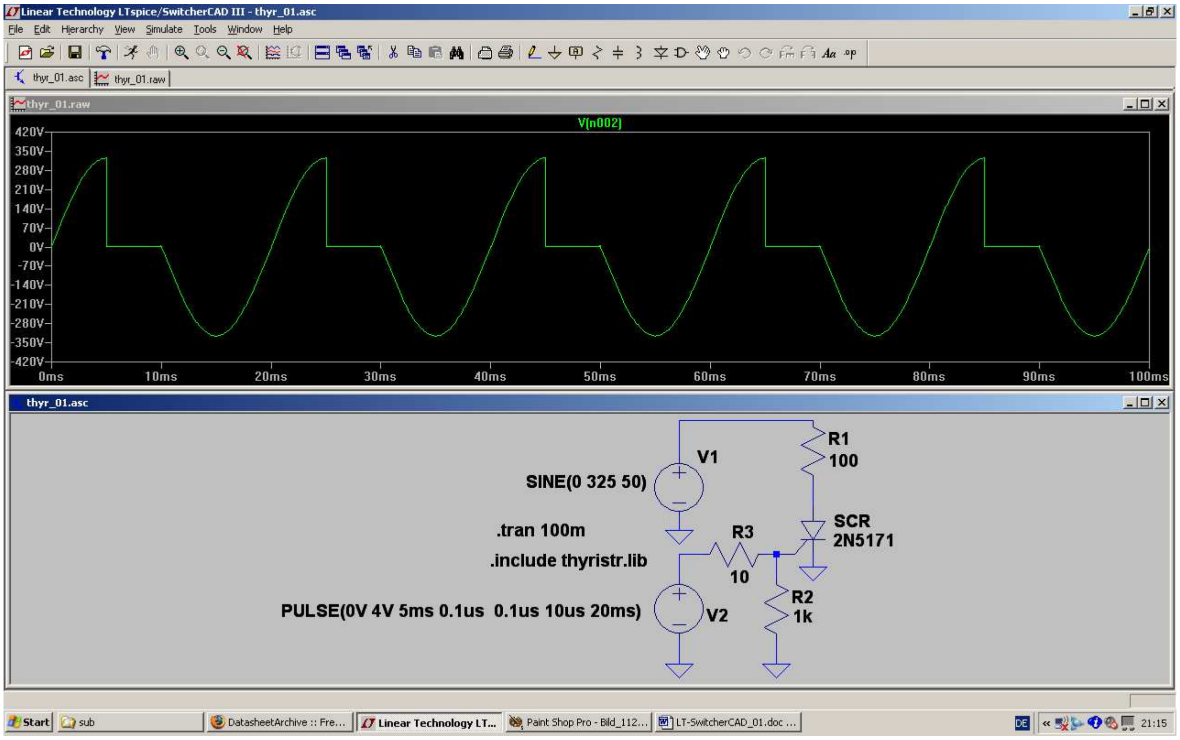
b) „.**include thyristr.lib"** prepares for the usage of the library with the thyristor SPICE-Models.

c) The pulse voltage at the gate is programmed by the line:

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

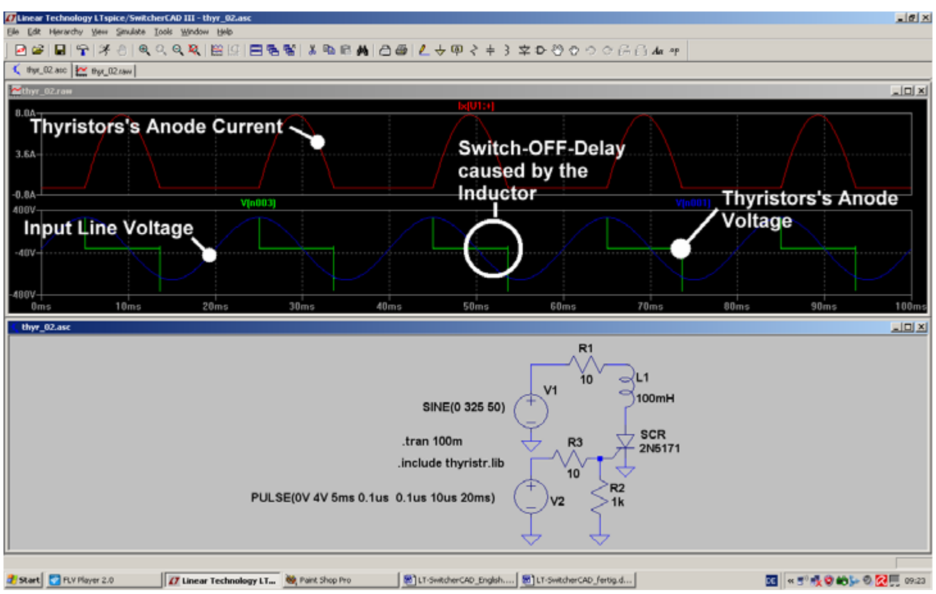
This gives a minimum amplitude of 0V and a maximum amplitude of +4V. The start delay time is 5ms, the rise and fall times are 100ns. Pulse length is 10µs with a period of 20ms.

So the anode voltage looks like:



12.3. Switching Inductive Loads

This is often a problem due to the stored magnetic energy in the inductor. Every time when the line voltage crosses 0V, the thyristor wants to switch off. But the magnetic energy stored in the coil keeps the current flowing in the same direction -- therefore the thyristor stays ON as long as there exists enough energy to do so …and so you get a „Switch OFF – delay" of the thyristor's anode current.



With an inductor of L =100mH and a series resistor of R = 10Ω we get this simulation result and we see that the thyristor is unable to switch OFF at the line voltage's zero crossing.
Current will flow as long as there is any remaining magnetic energy stored in the inductor. After the current drops below the holding current of the SCR the small amount of remaining energy gives a short damped oscillation (= coil + drain capacitance = resonant circuit).

12.4. Circuit with Gate Transformer

Very often practical circuits need complete isolation between the load circuit and the gate circuit. So in this case a gate transformer can be used to fire the thyristor:

Some details:

a) The pulse source to trigger the thyristor is now isolated from the gate by the transformer. But SPICE does not accept „floating circuits or nodes". So resistor R4 (1MEG) has been added to make the necessary ground connection.

b) Also the properties of the pulse voltage must be changed because a transformer cannot transfer DC values from the primary to the secondary winding. So we use "-5V" and "+5V" as minimum and maximum amplitudes.

c) Never connect an inductor without a series resistor to a voltage source……so R5 was added to the primary winding of the transformer.

d) The transformer is our well-known part „**xformer_01**" of the rectifier experiments.

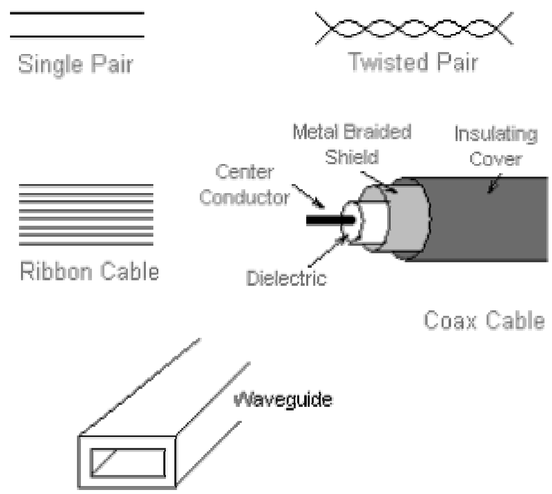
13. Project 9: Echoes on Transmission Lines

13.1. Transmission Lines -- only two Wires?

When considering a simple electrical circuit consisting of a voltage source and a load, everything seems to be easy: on the upper wire the current flows from the source to the load, on the lower wire the current flows back from the load to the source. But where is there a problem with that?

The circuit must be considered as a transmission line when a source transmits energy to a load that is some distance away. In other words, when the distance is greater than about 1/10th of the wavelength the circuit behaves more like a transmission line. At very high frequencies this can be in the cm or even mm range.

Let us at first have a look at the ways we can construct a transmission line:



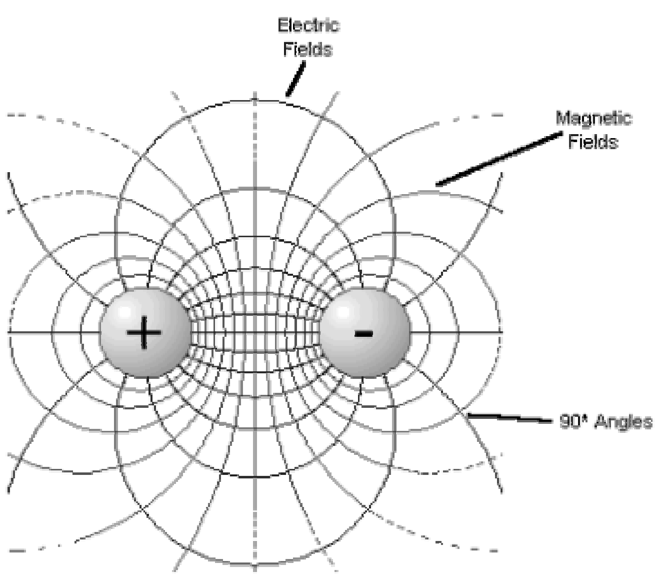
The „**Single Pair**“ is simple: two wires held in parallel.

The „**Twisted Pair**“ is the well - known standard for a LAN, CAN or telephone lines.

The **Coaxial Cable is widely used in the Communications, Radio, TV and Video.**

You can find „**Ribbon Cable**“ used as connection between the PC’s mainboard and the hard disk.

An unusual transmission line is the **waveguide**“, because an empty space serves as the transport media for the electric and magnetic fields.

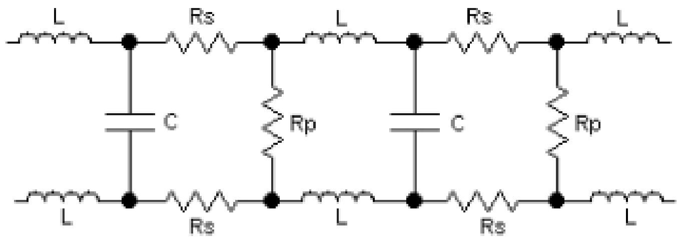


As soon as you apply a voltage to the input of a line (here the single pair is shown) you also get a magnetic field when current flows.
So please take note of the field distribution in the figure on the left.

It is now important to know that the electric field is representative of a capacitor. **This value increases with the line’s length!**

As soon as current flows a magnetic field forms around every wire. **This is representative of an inductor and this inductor’s value increases with the line length.**

To calculate the properties of the line you cut it in pieces -- but you must take very short pieces to simulate the "distributed capacitors and inductors".



Additionally you have to take into account the losses of the inductors (...caused by the skin effect which increases with frequency..) as series resistors. The losses of the insulation medium are representated by parallel resistors.

So you can extend the simple schematic model for a longer line by repeating the circuit.

As soon as a signal is applied to the input of the line, the following effects can the observed:

a) At the first instance of time the generator sees the cable simply as a resistive load with a value of the „characteristic cable impedance", i.e. 50Ω. The applied voltage at the cable input enters the cable and starts to travel along it (= "incident wave"). This characteristic cable impedance can be calculated as follows:

$$\mathbf{Z} = \sqrt{\frac{\textbf{Inductance}}{\textbf{Capacitance}}}$$

The inductance and capacitance values always refer to a defined cable length (mostly: 1m). For the well known RG58-type you have values of

100pF / 1m, 250nH / 1m

b) When the input signal is travelling along the cable length, the small capacitors are charged or discharged through small series inductors. This takes a finite time and so this gives rise to **„cable velocity"** or **"wave velocity** which is slower than the „ velocity of light in air".
We get:

$$\mathbf{v}_{\mathbf{cable}} = \frac{\mathbf{v}_{\mathbf{light}}}{\sqrt{\boldsymbol{\varepsilon}_{\mathbf{r}}}}$$

For a coaxial cable ε_r (....in America sometimes: „k") is the dielectric constant for the cables insulation between inner and outer conductor.

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

RG58 with Z = 50 Ohm uses polyethylene for this purpose. If you look at the table you can see that the wave velocity is reduced to 66.6% of the velocity of light, "C".

c) But what happens to the travelling energy on the cable? (see following chapter).

13.2. Echoes (= Reflections)

At high frequencies it is difficult to measure currents with high precision. Also, the determination of a source resistance (by “Open Loop” and “Short Circuit” Measurements) is nearly impossible. So we have to change our thinking at frequencies higher than 1MHz.

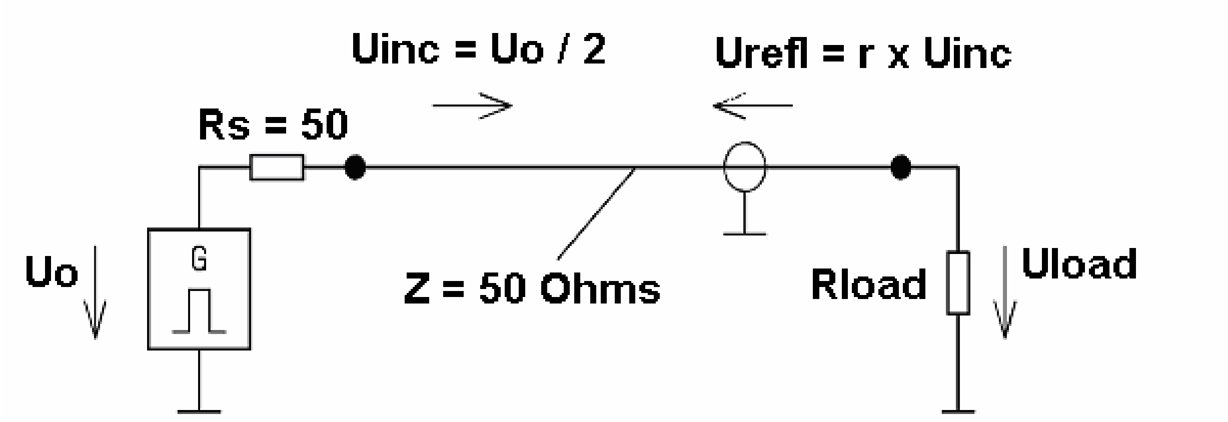
Everywhere in a system we use the same „system impedance“ (75Ω for radio and TV and video, but 50Ω in all other applications). This value is valid for the input- and output Impedance of all used blocks, for the cable impedance and for all terminations in the system.

The reason is very simple: now everywhere in the system you get perfect power matching (Ri = Ra).

With “directional couplers” the differences between ideal and practical situation can be measured and expressed as

Reflection coefficients.

Let us have a look at a pratical example:
A pulse voltage source with a source resistance Rs = 50Ω is connected to a transmission line (coaxial cable RG58 with Z = 50Ω) and generates a short pulse which is sent to the termination resistor Rload at the end of the cable.



Now the following happens:

a) When the cable is very long and the pulse is very short, then the source does not sense the load at the end of the cable. (....You know: the speed of light is 30cm per Nanosecond in the air and 20cm per Nanosecond in a RG58 cable).

b) So the cable’s input resistance is **50Ω** and forms a voltage divider with the source resistance of **50Ω**. This gives power matching and a pulse with amplitude of U0 / 2 enters the cable. The associated power (P = Uo x Uo / 4 x 50Ω) starts to travel at cable speed to the load resistance. This power is called **“incident wave”**

c) When the incident wave arrives at the load, then it can only be absorbed totally if the load resistor has also a value of **50Ω.** Any difference from 50**Ω** causes a mismatch of „perfect power matching“ and so the „power surplus“ is reflected and runs back to source (..at cable speed) as “**reflected wave**”

For the exact description of these effects the **reflection coefficient „r"** was introduced:

$$r = \frac{U_{reflected}}{U_{incident}} = \frac{Z_{Load} - Z}{Z_{Load} + Z}$$

Now the reflected wave can be calculated as

$$U_{reflected} = r \bullet U_{incident}$$

The voltage U_{Load} at the load resistor is then:

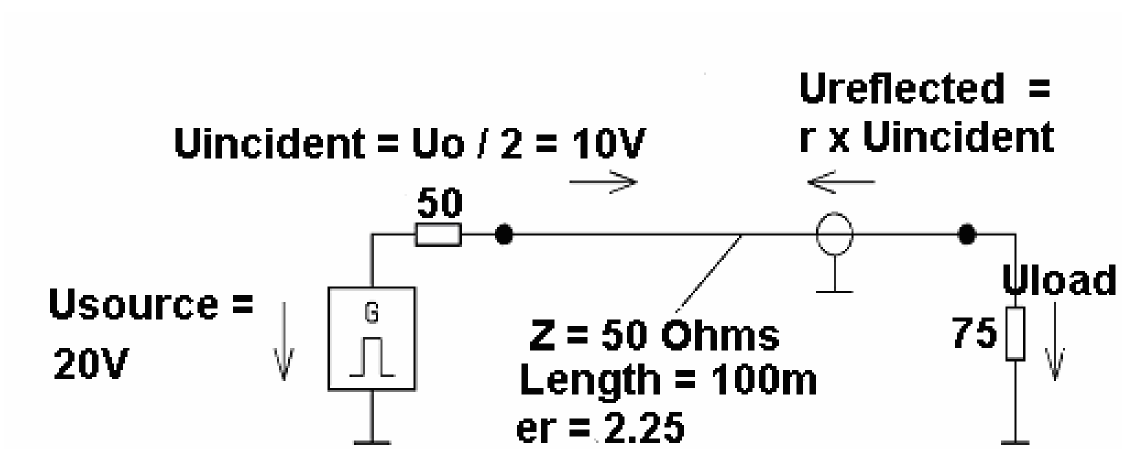
$$U_{Load} = U_{incident} + U_{reflected}$$

Note:
At every point in the cable Ohm's law must be valid, because you have travelling energies. So you can calculate the associated current at any point:

$$I_{incident} = \frac{U_{incident}}{Z} \qquad \text{and} \qquad I_{reflected} = \frac{U_{reflected}}{Z}$$

Example:
A pulse generator (Rs = 50Ω) generates in „no load condition" a pulse amplitude of +20V with a pulse width of 10ns. The pulse repetition frequency is 1kHz. The generator is connected to a RG58-coaxial cable (Z = 50Ω, length = 100m). The cable is terminated by a 75Ω-resistor. The dielectric constant of the cable is er = 2.25.

Calculate and draw the signals
a) At cable's input b) at cable's centre c) at cable's end



Solution:

a) Calculation of the reflection coefficient: $r = \frac{Z_{Load} - Z}{Z_{Load} + Z} = \frac{75\Omega - 50\Omega}{75\Omega + 50\Omega} = +0.2$

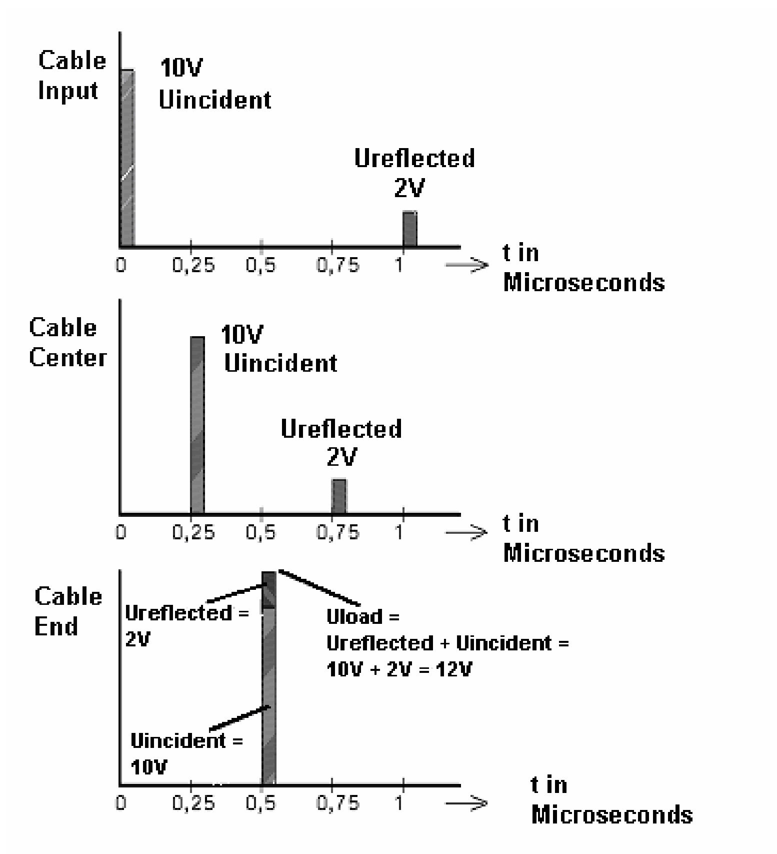
b) Cable velocity: $v_{Cable} = \frac{c}{\sqrt{\varepsilon_r}} = \frac{3 \bullet 10^8 m}{\sqrt{2.25} \bullet s} = 2 \bullet 10^8 \frac{m}{s}$

c) Signal runtime for 100m of cable length: $$t_{run} = \frac{l_{Cable}}{v_{Cable}} = \frac{100m \bullet s}{2 \bullet 10^8 m} = 0.5 \bullet 10^{-6} s$$

d) Amplitude of incident wave: $$U_{incident} = \frac{U_{Source}}{2} = \frac{20V}{2} = 10V$$

e) Amplitude of reflected wave: $$U_{reflected} = r \bullet U_{incident} = 0.2 \bullet 10V = 2V$$

f) Amplitude of load voltage: $$U_{Load} = U_{incident} + U_{reflected} = 10V + 2V = 12V$$



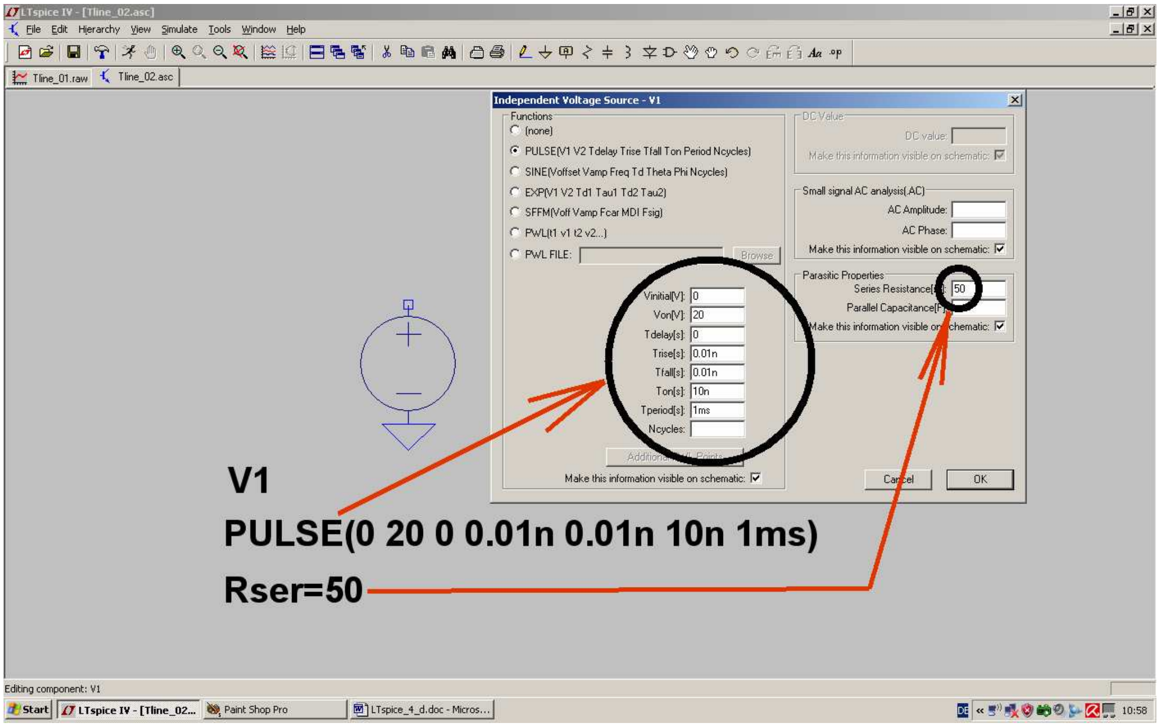
At the cable input we see at first the incident wave (produced by the pulse generator) with an Amplitude of +10V.
After twice the signal runtime for 100m (= 1 Microsecond) the reflected wave comes back from the load and reaches the cable input.

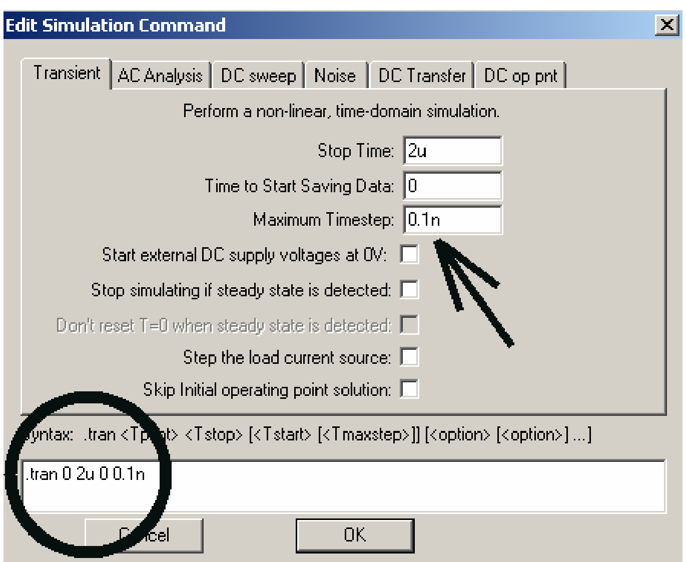
After 0.25 microseconds the incident wave reaches the centre of the cable (= 50m). The echo from the output comes 0.5 microseconds later and passes the centre when travelling back to the cable input.

Exactly after 0.5 microseconds the incident wave reaches the load resistor. Because there is no perfect power match, a part of the arriving energy is reflected and so a reflected wave is generated. At the load resistor we measure the sum of incident and reflected wave = 10V + 2V =12V.

13.3. Simulation of this Example with LTspice

We start with the voltage source from the part library. After placing the symbol on the screen we enter the properties to get a **pulse voltage with a source voltage of 20V, a rise and fall time of 0.01ns, a pulse length of 10ns and a period time of 1 ms. The source resistance must be set to 50Ω:**





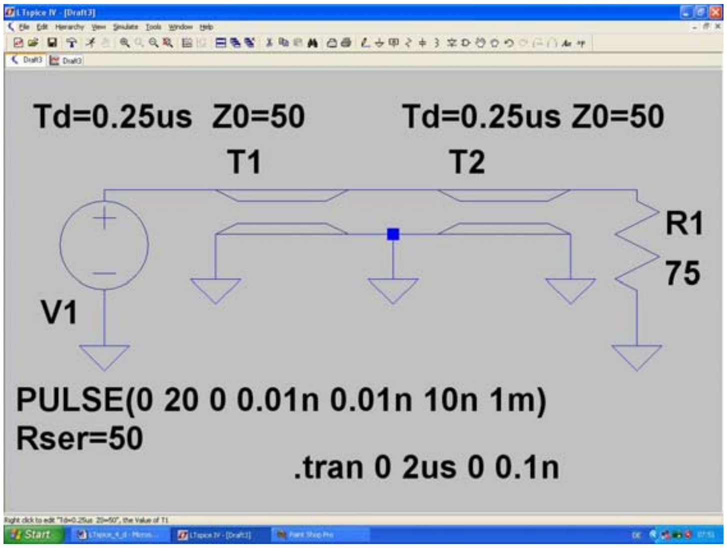
Note:
For such a short rise and fall time of 10 picoseconds it is necessary to use a maximum step size of 100 picoseconds in the simulation settings!

Now we need the **transmission line „tline**“ from the part library. But we must enter the “**signal delay time”** in the properties instead of the cable length!

This delay time can be calculated with the mechanical cable length of 100m and the “cable signal velocity” of 200 000 km / sec as follows:

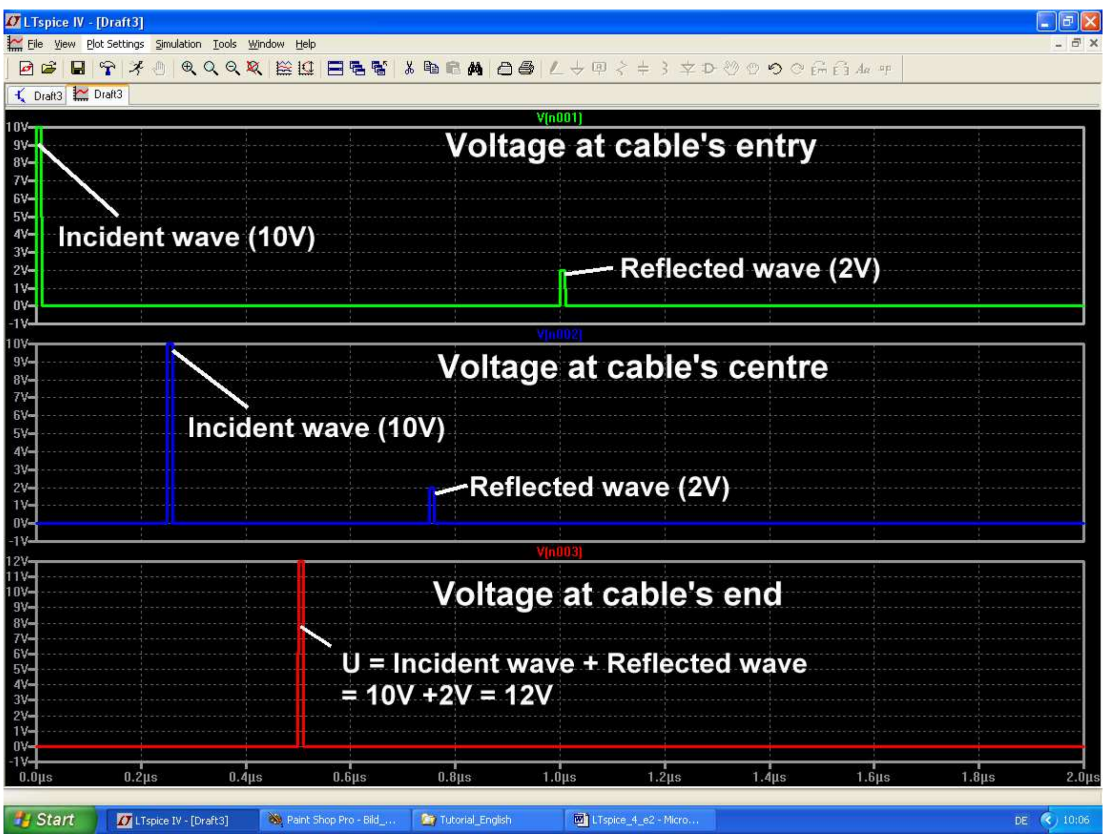
$$t_{delay} = \frac{l}{v_{cable}} = \frac{100m \bullet s}{2 \bullet 10^8 m} = 0{,}5\mu s$$

This delay time of 0.5µs is realized by using two equal pieces of RG58-cable which are connected in series. Each piece produces a delay of 0.25µs and so we get again a total delay of 0.5µs as desired. But now it is possible to view the signals in the MIDDLE of the total cable length.



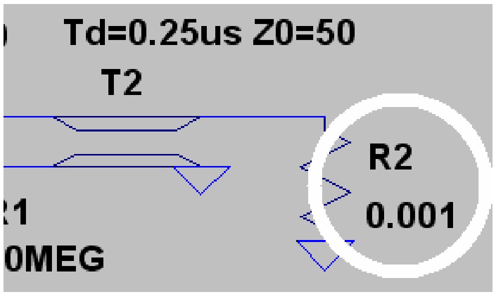
We terminate the end of the cable by a 75Ω resistor and simulate a run time of 2ms (…please check all of the schematic before starting the simulation…):

After the simulation the three voltages (input, middle and output) are presented in three different diagrams. Compare the result with chapter 13.3.:



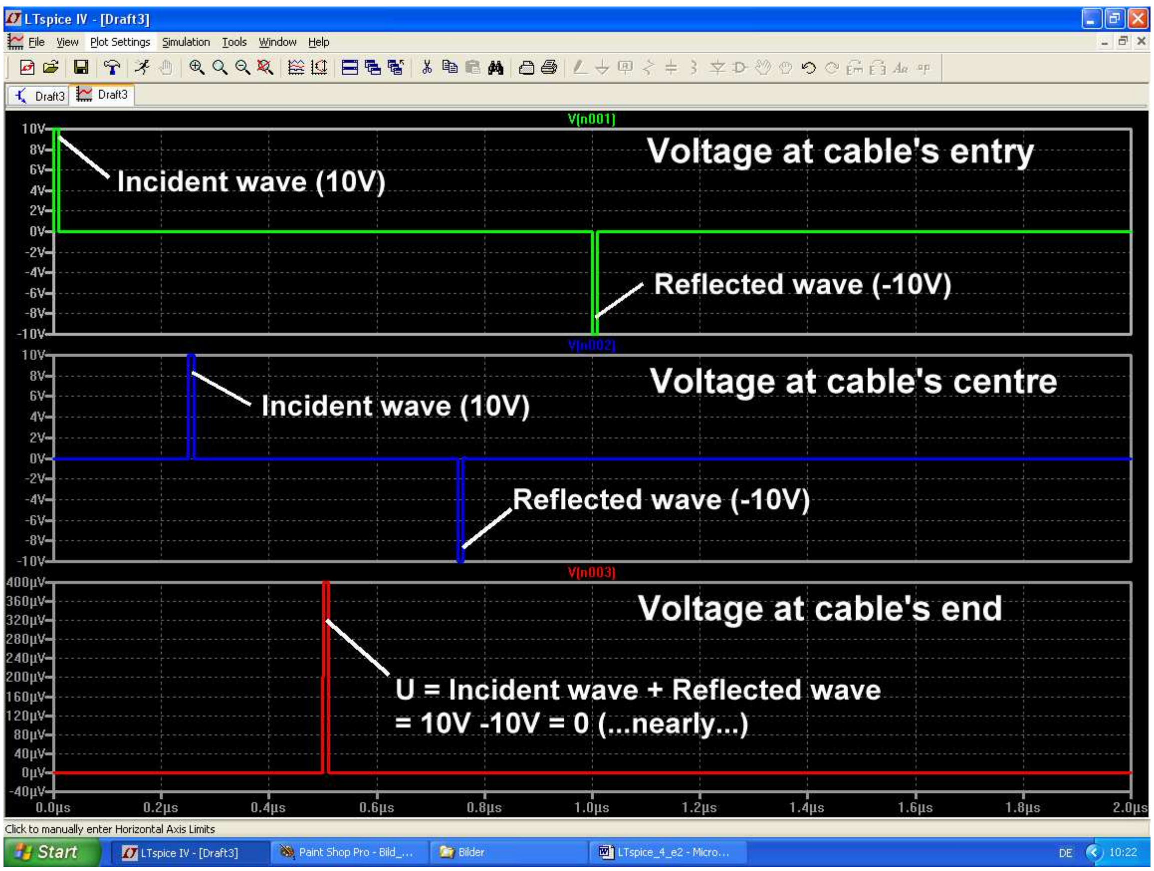
13.4. Open or Short Circuit at Cable's End

This is a very simple task because it is only necessary to change the value of the termination resistance.



a) **Short Circuit at cable's end:**
Choose a realistic value of 0.001Ω (= 1 milli-Ohm) as termination.

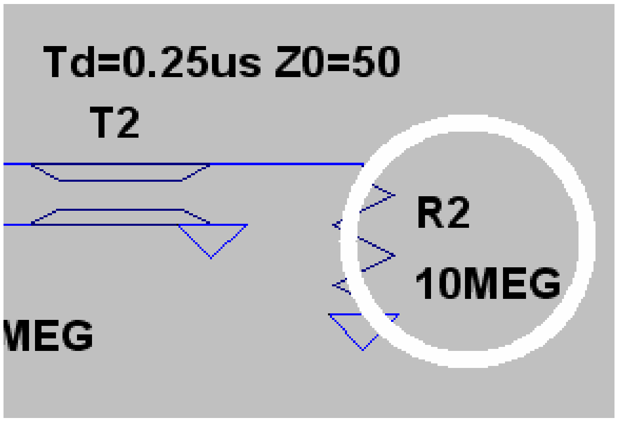
Then you can watch the following signals.



Explanation:
At the short circuit (at cable's end) the voltage must be zero. This means that no power is delivered into this short circuit and the incident wave (= power travelling from the source to the load along the cable) is sent back to the generator. But to produce zero volts at the short circuit the "reflected wave" (= voltage and power which run back) must change its polarity and so we see a negative echo on the cable.

Because we have replaced the ideal short circuit by a 0.001Ω - termination we get a short pulse with an amplitude of 400 microvolts at the output of the cable.

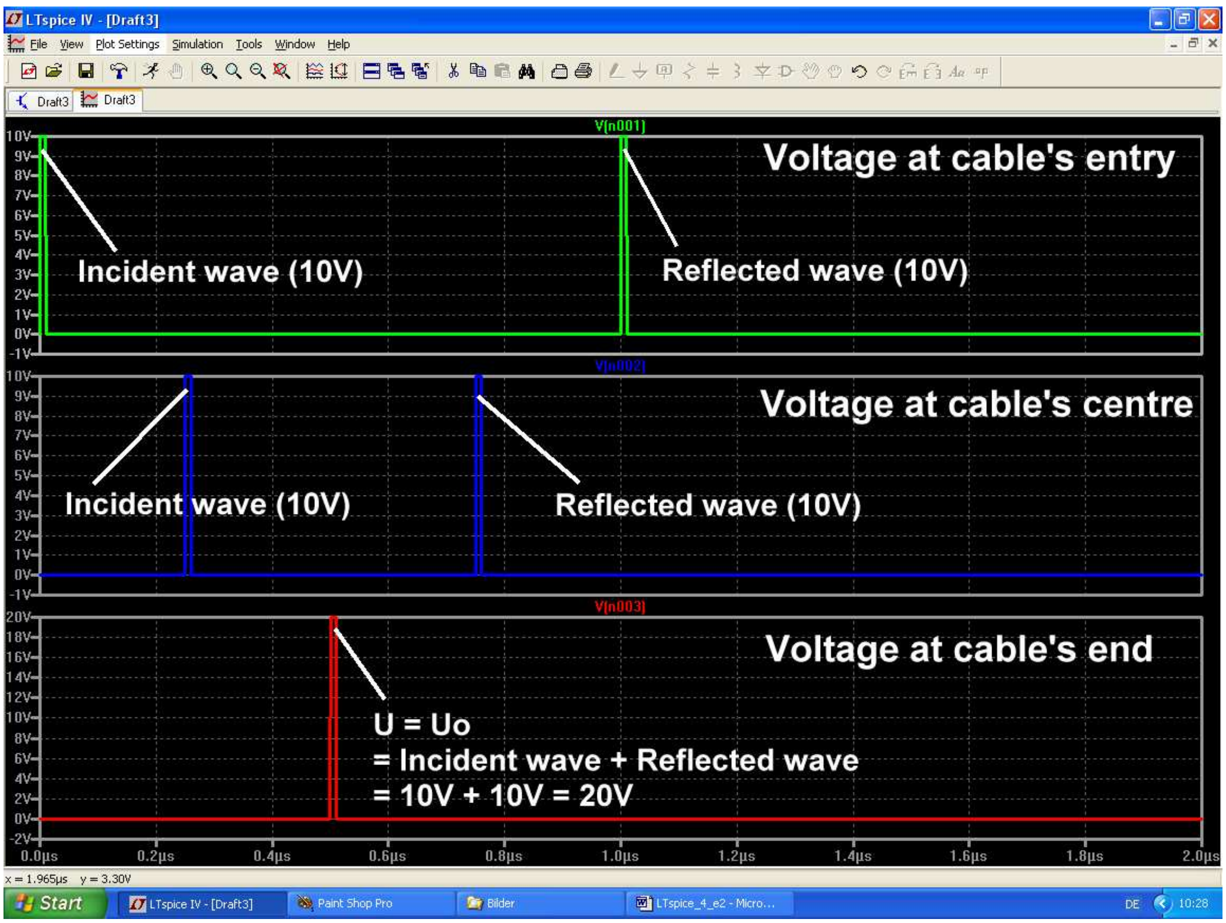
b) Open circuit at cable's end



Change the termination to 10MΩ -- this will do for an ideal open circuit.

Warning:
Never delete the termination R2 from the schematic to get an ideal open circuit. For the program this would be a "floating node" and at once you get an error message and a simulation abort!

And this is what you get as a simulation result:



Note:
Because at the cable's end the current is zero no power is delivered to the termination. So the incident wave is reflected and runs back to the generator. But now -- as always when a power supply doesn't have a load -- the voltage at cable's end rises to the value of the unloaded source voltage (= 20V in our example).

13.5. Lossy Cables (i. e. RG58 / 50Ω)

13.5.1. How can I simulate an RG58 Coaxial Cable?

We need a part from the LTspice – part - library which is named

ltline

(= **lossy transmission line**).

But there is a little problem:

We have to write a special model file with the RG58 properties. These properties can be found in the internet and must be applied to create the desired cable model and symbol.

In the LTspice manual you will find the necessary information which is as follows:

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

Details:

„model“	is the SPICE syntax for a part model
„RG58“	is the name of the new part
„LTRA“	means „lossy transmission line“
„len=100“	means the „unit length” and „len = 100“ says that we have a cable length of 100m if the unit is “1m”.
„R=1.5“	means that for 1 unit length (= 1m) a series resistance of 1.5Ω must be taken into account.
„L=250n“	gives an inductance of 250nH per unit length (here: for 1m)
„C=100pF“	gives a capacitance of 100pF per unit length (here: for 1m)

Write this line with an editor and save it as „**RG58.mod**“ using the path „**LTspiceIV / lib / sub**“.

(The properties „100pF per 1m“ and „250nH per 1m“ come from the internet for RG58.
The value for the losses with „1,5Ω per 1m“ is estimated and will be checked in the next chapter……

13.5.2. Simulation of Cable Loss at 100MHz

At first an overview from the internet:

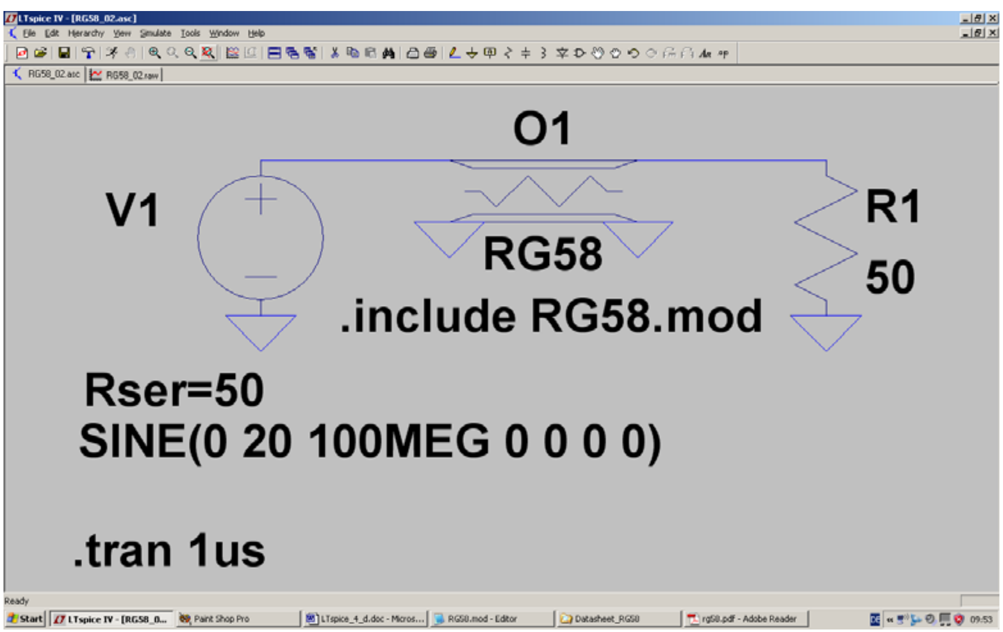
Frequenz *frequency* MHz	Dämpfung typ. Meßwerte *attenuation typ. measured values* dB/100m	max. Dämpfung *max attenuation* dB/100m	min. Rückflußdämpfung *return loss* dB	$P_{40°C,max}$ W	$U_{40°C,max}$ V
50	10,1	11		210	105
100	14,6	16		145	90
200	21,6	24		95	75
300	27,5	30		75	65
450	36,0	40		55	55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

As the table shows, losses increase with frequency. So you must prepare a new simulation for each frequency which you want to use.
Let us test this at 100 MHz.

Step 1:
At 100MHz the table gives a value of 14.6dB per 100m. So the output voltage at cable's end for this length is attenuated by the factor

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

Step 2:
We use a RG58 cable length of 100m (…entered in the RG58.mod – file!) and a termination of 50Ω.
The voltage source works with a sine wave (peak value = 20V) and a source resistance of 50Ω. So we have this schematic:



Note:

a) Right click on „**LTRA**“ under the symbol of the transmission line and replace it by „**RG58**“-

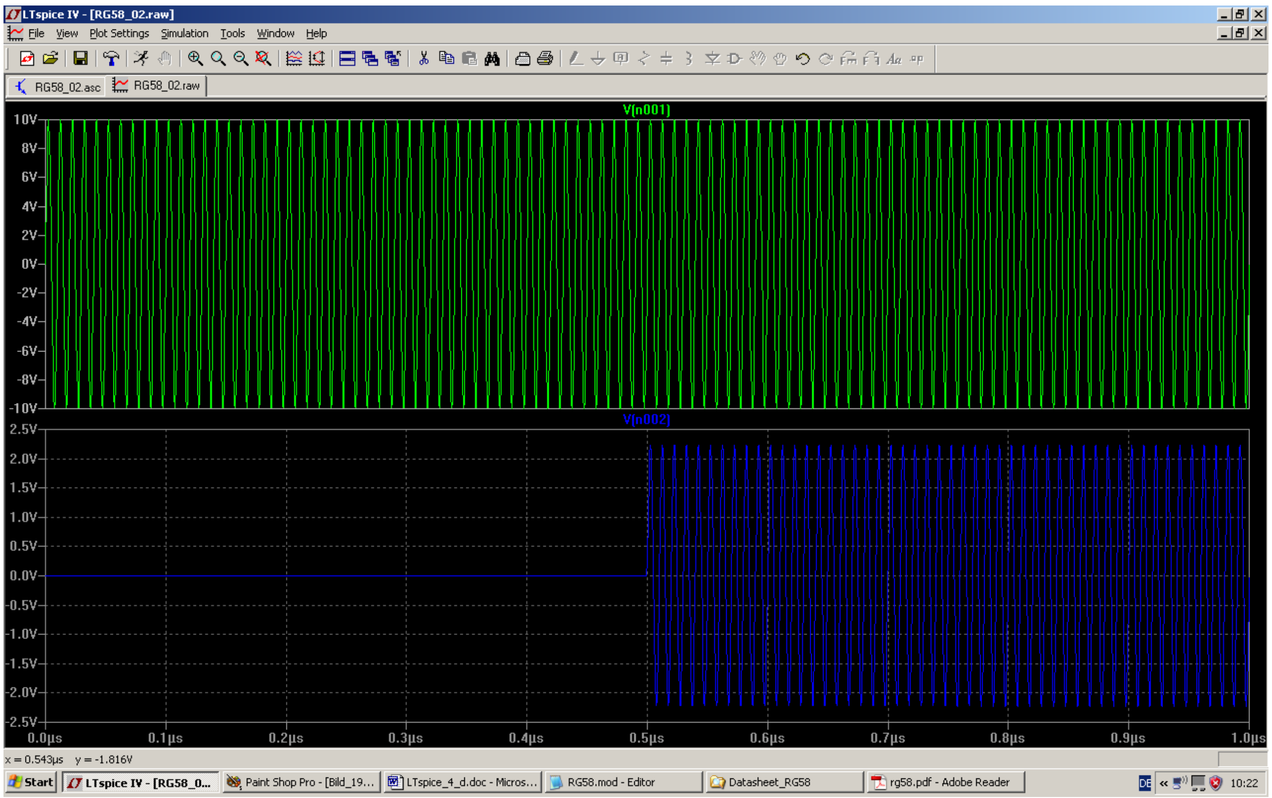
b) Use „**Edit“ and „Spice Directive**“ to get

.include RG58.mod

on the screen..

c) Use a **simulation time from 0….1µs.**

Step 3:
After the simulation use two different diagrams to present the input and the output voltage:

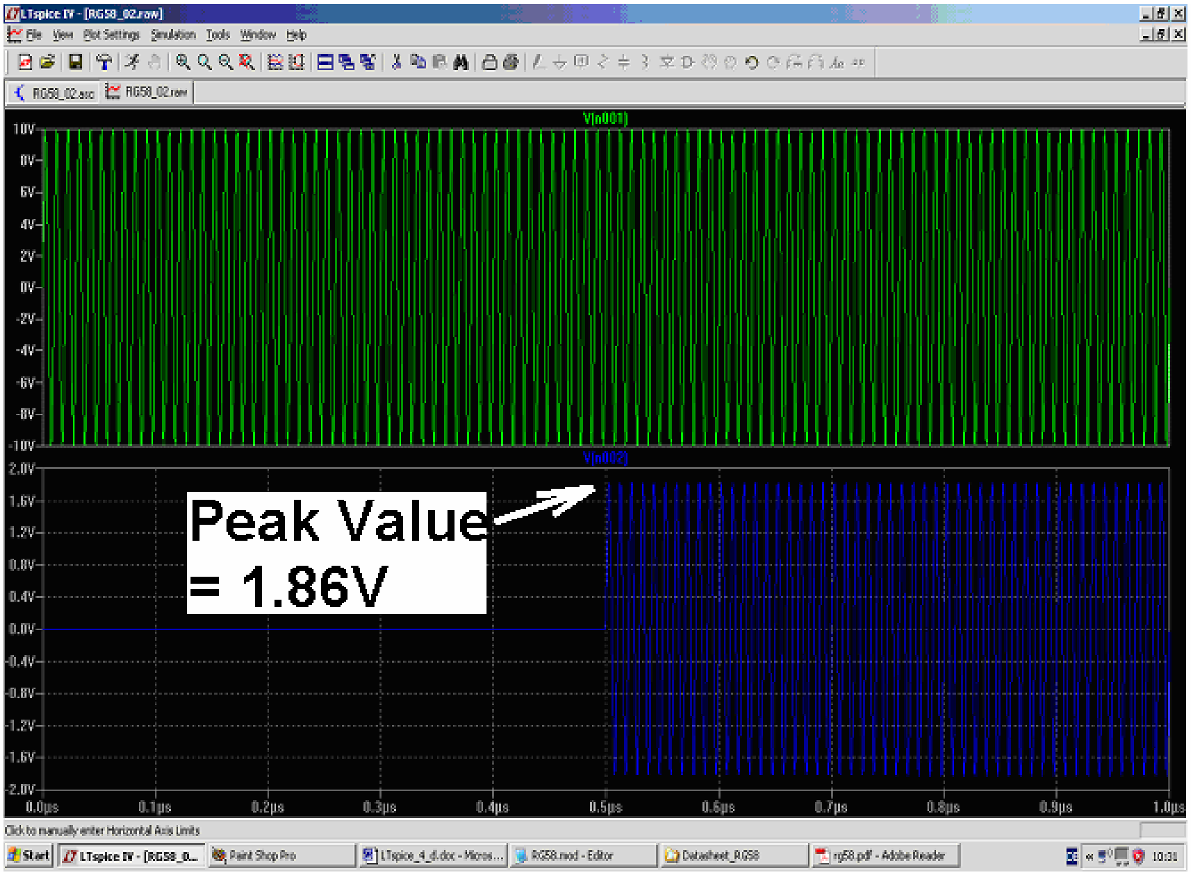


It is nice to determine the „signal propagation time on the cable“ as a starting time difference between the two voltages for a cable length of 100m.

An attenuation of 14.6dB per 100m would decrease the output voltage to 1.86V, but we find 2.25V.
So the series resistance per unit length must be **increased from 1.5 to 1.7Ω** to see the correct value of 1.86V……..

Step 4:
Open the RG58 model file with the text editor and change this „R“-value. Repeat the simulation and the output voltage’s peak value is now at 1.86V.

Compare your simulation with that on the next page.

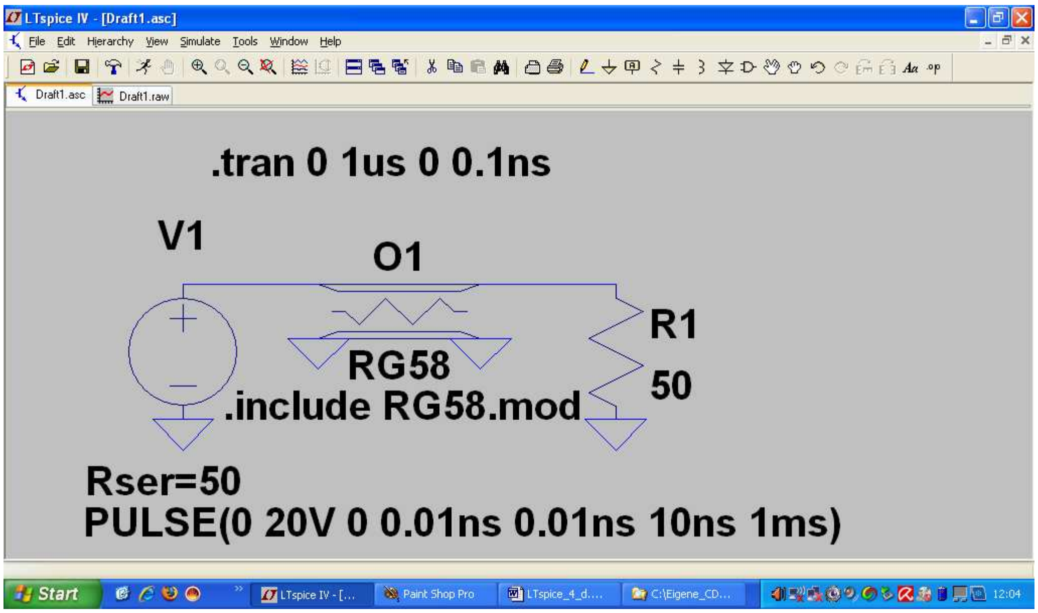

LTspice IV - [RG58_02.raw]
V(n001)
V(n002)
Peak Value
= 1.86V
10V
8V
6V
4V
2V
0V
-2V
-4V
-6V
-8V
-10V
2.0V
1.6V
1.2V
0.8V
0.4V
0.0V
-0.4V
-0.8V
-1.2V
-1.6V
-2.0V
0.0µs
0.1µs
0.2µs
0.3µs
0.4µs
0.5µs
0.6µs
0.7µs
0.8µs
0.9µs
1.0µs

13.5.3. Feeding the RG58-Cables's Entry with a Pulse Voltage

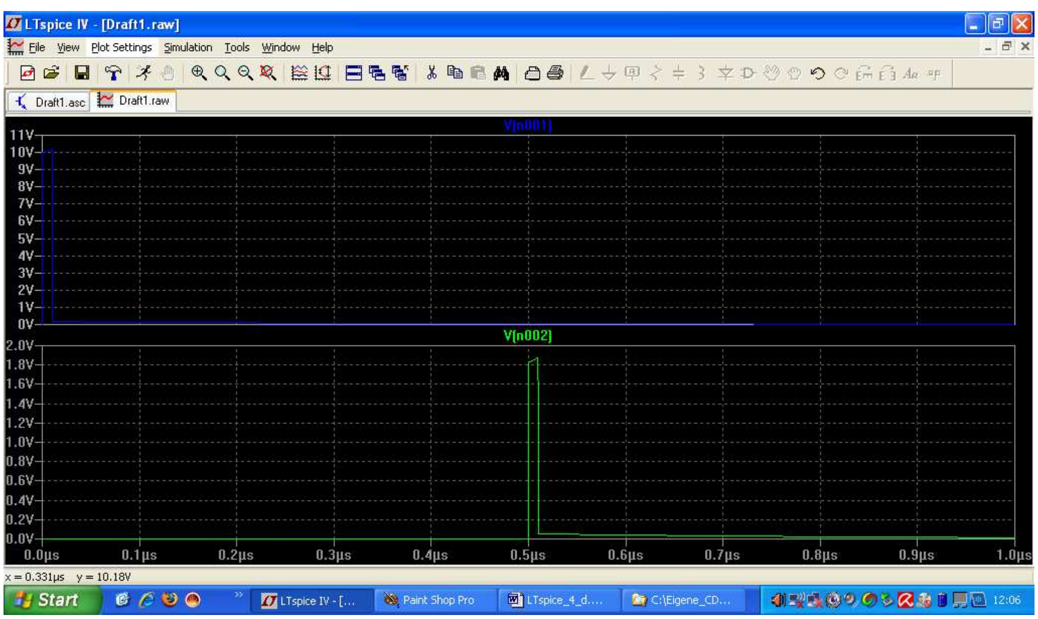
We now test our „lossy line“ with a pulse voltage at the input. Let the source properties be:

Source Voltage Maximum Value	**= 20V**
Source voltage Minimum Value	**= 0V**
Pulse Length	**= 10ns**
Rise Time	**= 0,01ns**
Fall Time	**= 0,01ns**
Periode Time	**= 1ms**

Schematic:



Simulation Result:

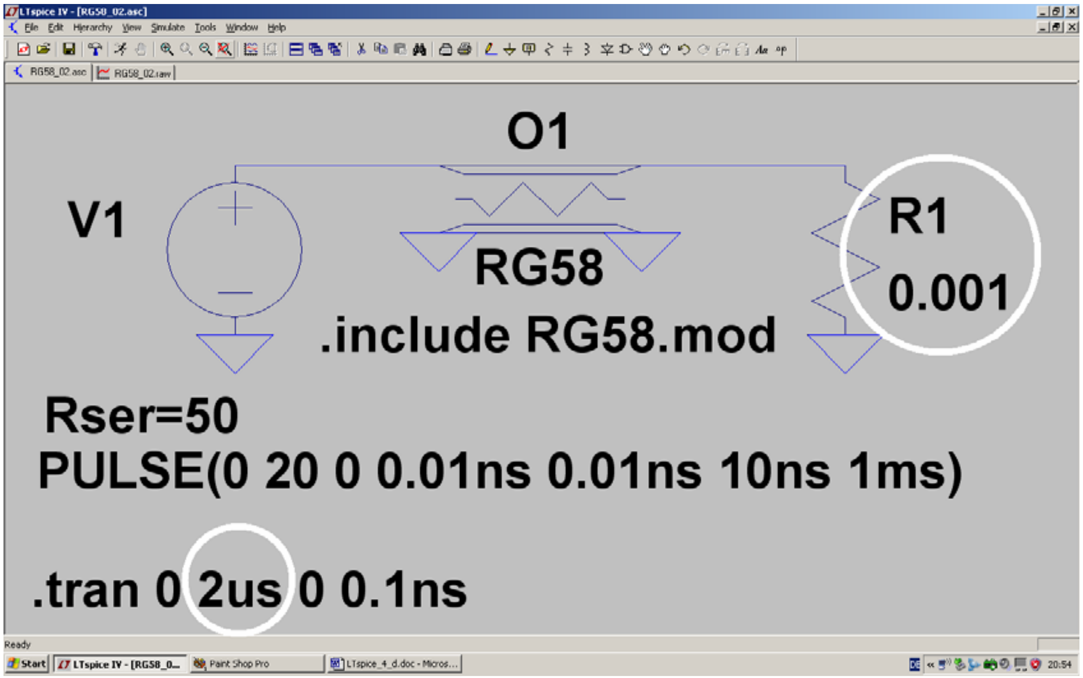


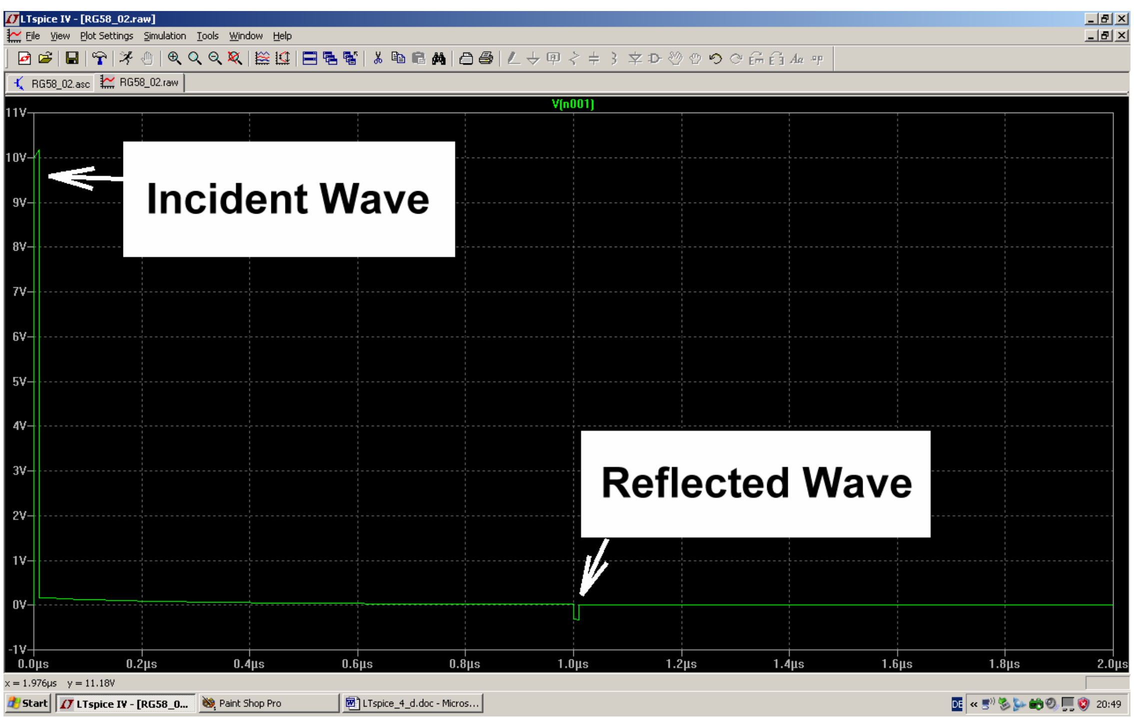
Because the value „R = 1.7Ω / 1m“ does not change with frequency in the simulation (in comparison with the reality: really a pity!), all harmonics are attenuated by the same factor. So the curve form does not change as in practice (= no linear distortion). Only the amplitude at the cable's output is reduced.

13.5.4. A Short Circuit at RG58 Cable's End

Do you remember? This is made using a termination of 0.001Ω, but we must not forget to increase the simulation time to 2ms (otherwise the echo cannot be seen in the waveform viewer).

You need not show the simulated output voltage in a (separate) diagram, because the voltage on a short circuit is "0"….so only the input voltage is of interest.





The echo is very small because of the huge cable attenuation.

14. Project 10: S-Parameters

14.1. Echos once again, but with more System (= S-Parameters)

Energy is always transported on a cable by travelling power. But for easier understanding and calculating we use the square root of the power to get an expression only with **voltage** and we name this result "**wave**".

So we get the „**incident wave a**": -

$$\mathbf{a} = \sqrt{\mathbf{P}_{\text{incident}}} = \sqrt{\frac{\mathbf{U}_{\text{incident}}{}^{2}}{\mathbf{Z}}} = \frac{\mathbf{U}_{\text{incident}}}{\sqrt{\mathbf{Z}}}$$

And the „**reflected wave b**":

$$\mathbf{b} = \sqrt{\mathbf{P}_{\text{reflected}}} = \sqrt{\frac{\mathbf{U}_{\text{reflected}}{}^{2}}{\mathbf{Z}}} = \frac{\mathbf{U}_{\text{reflected}}}{\sqrt{\mathbf{Z}}}$$

The connections of a part are now named "ports". A simple block (= filter, amplifier, attenuator...) with input and output is then a **TWO PORT**, but a resistor or a diode is a **ONE-PORT.**

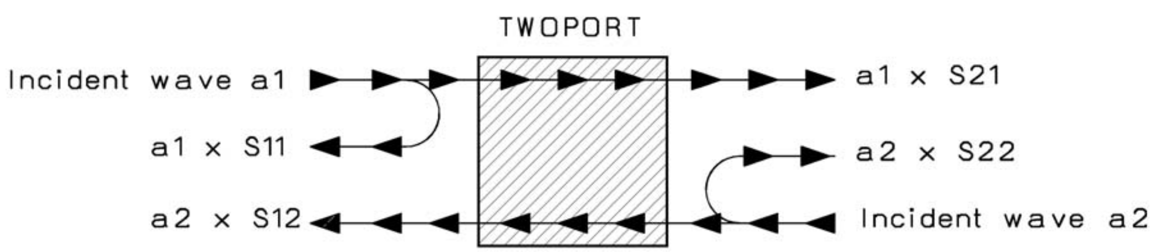
.

TWOPORTs have additionally a gain or an attenuation. This means:

The incident wave at a port will therefore produce a signal at the other port.

This effect can be measured and expressed as „transmission".

If we look at an amplifier as a typical TWOPORT, then we get the following **signal flow chart**:



Input side = port 1:

The incident wave a1 (coming from the signal source) feeds the input (port 1) of the TWOPORT. In general we find then a wave „b1" running back in direction of the signal generator. This "**backrunnning wave b1**" can be separated from the incident wave a1 by a directional coupler and consists of two parts:

1) **The reflected wave „a1 x S11"** due to the fact that the input impedance at port 1 isn't exactly 50Ω and

2) a **second signal** which is normally caused by „**feedback"** (= reverse transmission) from the output to the input.

So the total echo b1 on Port 1 can be expressed as $b_1 = a_1 \bullet S_{11} + a_2 \bullet S_{12}$

If the output (port 2) is terminated with exactly Z = 50Ω, then no echos come back from the termination resistance in direction of port 2. Now the incident wave a2 at the output is zero and we get a simple solution for S11:

$$S11 = \frac{b_1}{a_1} \qquad \text{for } a_2 = \text{zero}$$

But…this is the well – known **Input Reflection Coefficient "r"** of the preceeding example! In most cases „r" is complex and so the part manufacturers always list MAGNITUDE = MAG and ANGLE = ANG in their S-parameter-files for different frequencies.

S12 is then the „reverse transmission coefficient“. In this case we terminate the input (Port 1) by 50Ω and disconnect the input signal generator. The signal generator is now connected to the output (port 2) to measure the feedback effects at the input (port 1).

$$S_{12} = \frac{b_1}{a_2} \qquad \text{for } a_1 = \text{zero}$$

Output side = port 2:
One signal generator is connected to the output (port 2) and sends the wave a2 to this port. The other signal generator feeds the Input of the Twoport.

With a directional coupler we can at the output separate the incident wave a2 and the “echo b2” coming back from the TWOPORTs output:

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

If we now disconnect the signal generator from the input and terminate with 50Ω, then the output’s echo consists only of the „reflected wave “$\mathbf{a_2}$ **x S22**” (if the output impedance differs from 50Ω). So S22 is simply the „output reflection coefficient for perfect matching at the input (= port 1)”.

$$S_{22} = \frac{b_2}{a_2} \qquad \text{for } a_1 = \text{zero.}$$

Now let us look at the rest of the formula:
If the input signal generator is switched ON and the output signal generator disconnected and replaced by a perfect 50Ω-Match, we get only the wave

$$\mathbf{b_2 = a_1 \times S21}$$

at the output. So S21 = forward transmission coefficient means simply the gain or attenuation of a TWOPORT (with perfect match at the output)!

$$S_{21} = \frac{b_2}{a_1} \qquad \text{for } a_2 = \text{zero.}$$

Please remember:
The S-Parameters are **„wave“ ratios** -- that means: voltage ratios. So if you want to calculate the power gain, you must use $\mathbf{(S21)^2}$

How can we get an input or output impedance value from S11 resp. S22?

Use the following formulae, but remember: normally all the numbers are complex……:

Input resistance: $$\mathbf{Z_{Input} = Z \bullet \frac{1+S11}{1-S11}}$$

Output resistance: $$\mathbf{Z_{Output} = Z \bullet \frac{1+S22}{1-S22}}$$

It is also possible to calculate the reflection-S-parameters from measured impedance values, but therefore you need a modern microwave CAD-program (like the free ANSOFT DESIGNER SV) to save lot of efforts and sweat.

And lastly let us have a look at the S-Parameter-file of a modern MMIC (= “Touchstone” or “S2P” file), downloaded from the Internet:

```
!.....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
```

And this is how to interprete these files:

a) A line starting with an exclamation mark is a comment and therefore ignored for calculation

b) The line **# ghz S ma r 50“** means:

The first value is the **measuring frequency in GHz.**

Then follow the 4 S-parameters in the range **S11 S21 S12 S22**

“ma” says: every S-parameter is given as its **magnitude**, followed by the **angle**.

“r 50” tells us that the system resistance (for measurement and application) is real 50 Ohm.

Example line for **f = 1GHz**:

S11 = 0.32 / 144 degrees (Magnitude of the input echo is 0.32 x a1, the phase of the echo is 144 degrees -- caused by an input impedance **not equal to 50Ω**).

S21 = 18.18 / -72 degrees **When the output (= Port 2) is terminated by 50Ω, then we get a voltage gain of 18.18 (= ratio of output wave a2 referred to input incident wave a1)**

S12 = 0,016 / 21 degrees The feedback voltage measured at the input (= port 1) is 0.016 =1.6% of the output wave a2

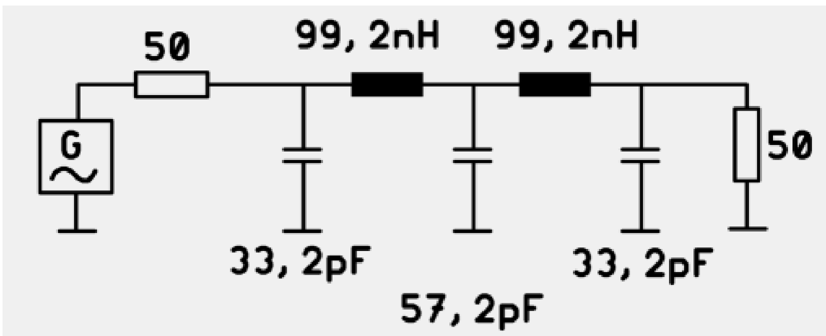
S22 = 0.5 / -20 degrees Magnitude of the output echo is 0.5 x a2, phase of the echo is 20 degrees -- caused by an output Impedance **not equal to 50Ω**

14.2. Example: 110MHz – Tchebyshev Lowpass Filter (LPF)

Let us now simulate the S-parameters of a 110MHz Tchebyshev lowpass filter. This type of filter suffers from “passband ripple” but provides a sharper transition from the passband to the stopband.

Properties:

„Ripple“ corner frequency	fc = 110 MHz
Filter degree	n = 5
System resistance	Z = 50Ω
„Passband ripple“ (This ripple causes a maximum value of S11 = -16,4 dB)	0.1 dB.



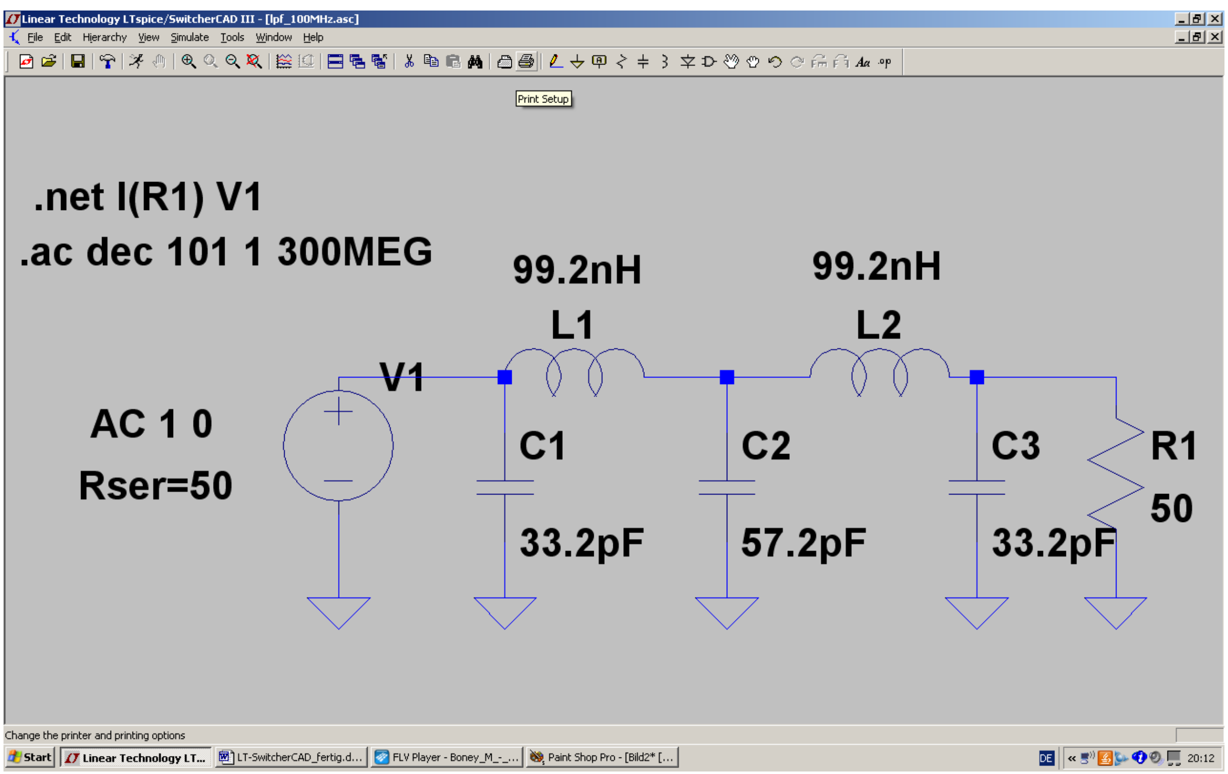
For the filter part calculations use the free microwave CAD software (Ansoft Designer SV) with the integrated filter designer and you get:

C1 = C3 = 33.2 pF

C2 = 57.2 pF

L1 = L2 = 99.2 nH

And this is the simulation schematic for LTspice:



Explanations:

a) The directive **.net I(R1) V1** starts a network calculation with the output voltage referred to the input voltage (given by source V1).

b) Then program a decadic sweep from 1Hz to 300 MHz with 101 points per decade by the directive

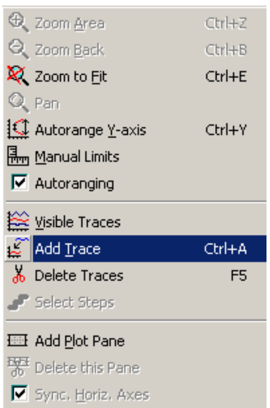
.ac dec 101 1 300MEG

c) For a sweep the properties of source V1 must be set to „AC amplitude =1 and AC phase = 0“:

AC 1 0.

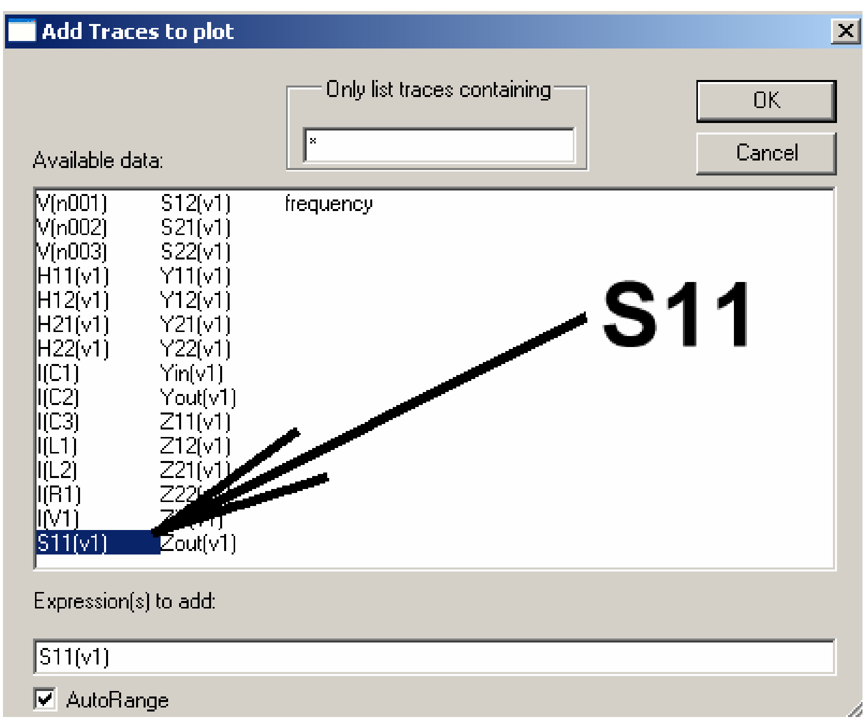
d) After a right mouseclick on the symbol of source V1 enter the value for the necessary source resistance of 50Ω in the system:

(Rser=50)



Now press the simulation button and follow these steps:

Step 1:
The diagram for the results is still empty. So click right on it with your mouse and choose **„Add Trace“**



Step 2:
Find

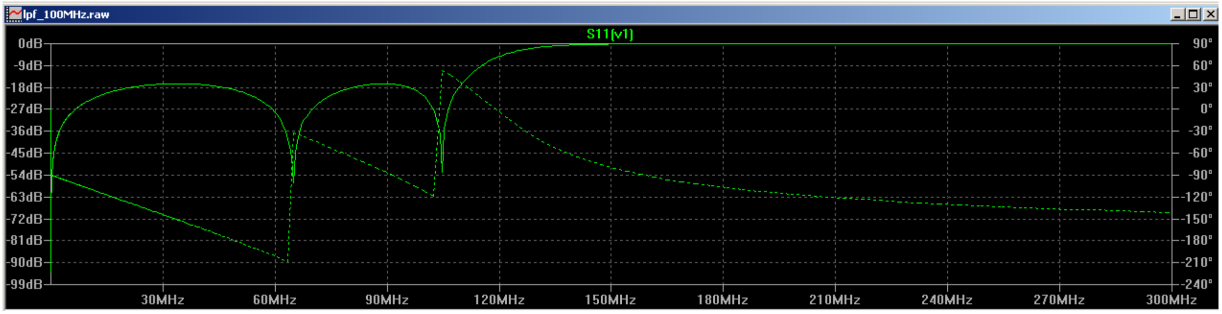
„S11(v1)“

in the list, mark it and

press OK.

For professionals:
In this list also the Y- and Z-parameters can be found...

Step 3:

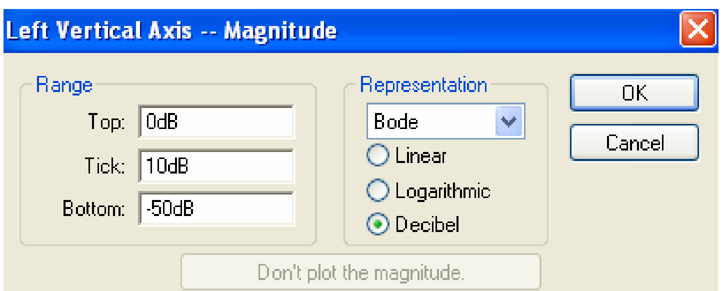


The result is not so bad, but the presentation…….
The phase curve might be interesting, but we want to delete it. The dB-scaling of the vertical axis must be improved.

So move the cursor to the scaling of the **right vertical axis** and click on it. Choose in the menu

Don't plot phase.

The phase curve disappears at once.

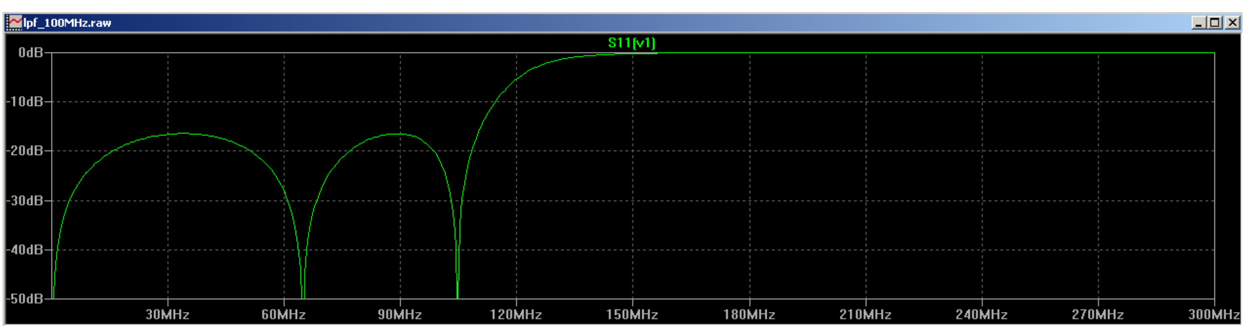


Now we repeat this procedure with the left vertical axis and enter the values of the left figure (= range from 0dB to -50dB with a tick of 10dB).

At last and once more click right on the diagram to activate the option

GRID

Now you have this result:



Step 4:
We complete the work by adding the S21 curve. Let's go:

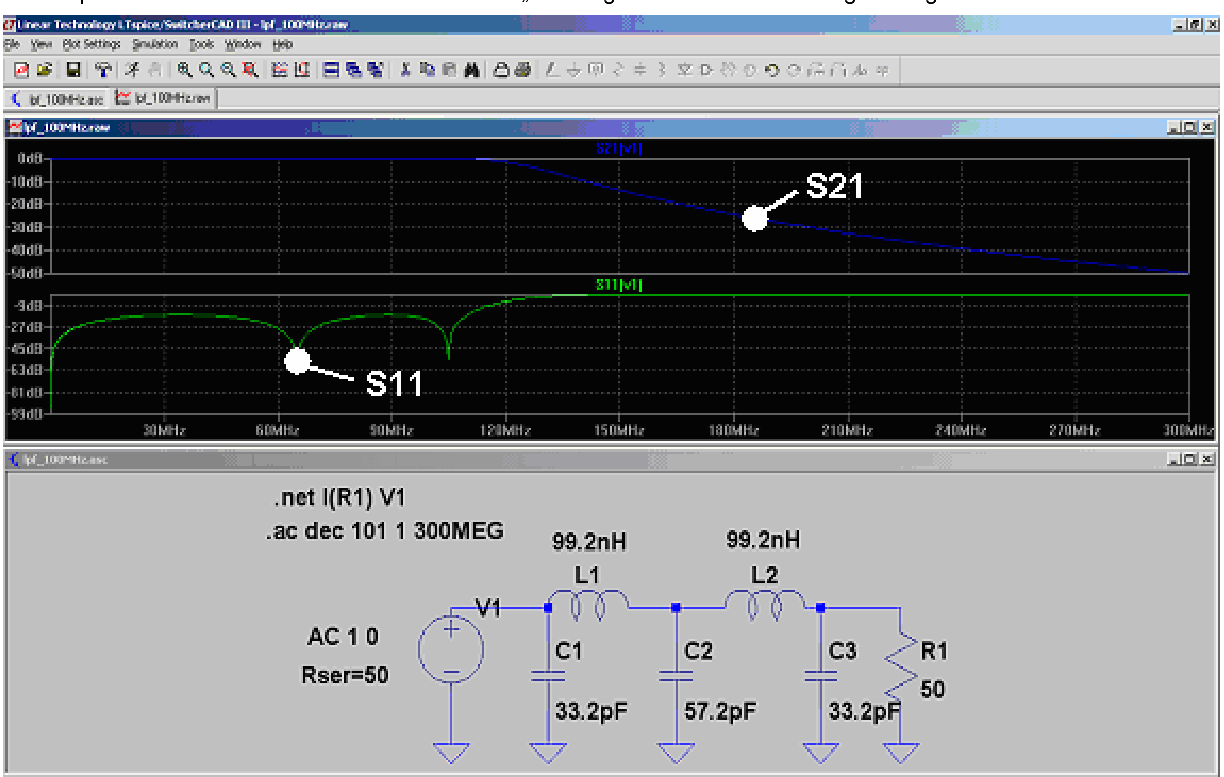
Right mouseclick on the diagram / Add Trace / choose S21 and press OK.

Sorry, but it is now better to repeat the procedure with the scaling of the left vertical axis to show a range from 0....-50dB. This gives the final result on the next page.

For a separate presentation of the two S-parameters right click on the diagram and choose

Add Plot Plane

for the parameter S21. Delete the S21-curve in the „old“ diagram and scale both diagrams again:



15. Project 11: Double Balanced Mixer (= DBM = Ring Modulator)

15.1. Fundamentals and Definitions

This well known circuit is used for converting frequencies without changing the information contents.

We have two inputs and one output:

a) RF = radio frequency input. Here the applied signal amplitude must be small (typically below 50...100mV)

b) LO = local oscillator input. Here the signal must be high (typically up to several volts)

c) IF = intermediate frequency output. Here we find the converted signals.

Note:

This operates a multiplier circuit for the two input signals!

When two sine signals are multiplied we get the following result:

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

This means for our circuit:

At the output of the DBM the two input signals have disappeared and are now replaced by their sum frequency and difference frequency!

When using a square wave signal as the LO-input, then this signal consists of the fundamental frequency f1 and the harmonics (with 3 x f1 , 5 x f1 , 7 x f1 , 9 x f1......).

Not only the fundamental frequency but also the harmonics are now multiplied by the RF-signal in the DBM and produce sum and difference frequencies with amplitudes that decrease linearly with the “harmonic’s degree”.

So you get at the DBM’s output a lot of „signal pairs“ (= sum- and difference frequencies). Every pair is located at a harmonic frequency.

In practical applications the desired new frequencies must now be picked out with a lowpass or a bandpass filter by the user.

15.2. The Ring Modulator

This part is used everywhere in communication systems and is offered by a lot of manufacturers -- as an active or passive device. We'll examine the standard passive version.

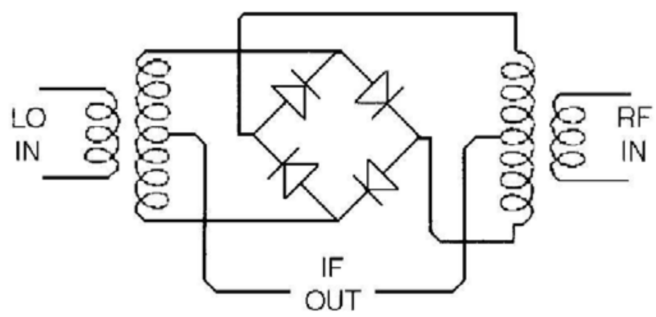


Figure 2. A popular diode-ring mixer topology.

A ring of four schottky diodes is connected to two transformers. Each transformer uses 3 identical windings. The two secondary windings are connected in series.

The use of superfast schottky switching diodes and special microstrip circuits enables cutoff frequencies higher than 50....100GHz in special applications.

Explanation:

„**LO-IN**" means „**local oscillator input**" for the converting signal.

„**RF-IN**" means „**radio frequency input**" for the information which shall be converted.

„**IF-OUT**" means „**intermediate frequency output**" and delivers the multiplication result.

Principle of operation:

The LO-signal at the secondary winding of the left transformer (amplitude: up to several volts….) switches the right diode pair ON at the positive halfwave and the left diode pair ON at the negative halfwave.So either the upper secondary winding or the lower secondary winding of the right transformer with the RF signal is connected to the IF output by the conducting diode pair. But the RF signal which finally reaches IF-OUT is changing its polarity rhythmic with the LO frequency. That is the same effect as multiplying the RF – sinewave with a LO-squarewave (amplitude values: +1 and -1).

Advantages: Very low distortion. Useful for broadband applications up to many GHz. Needs no power supply, but correct 50Ω-matching at all ports. Produced in huge quantities at reasonable prices by a lot of manufacturers.

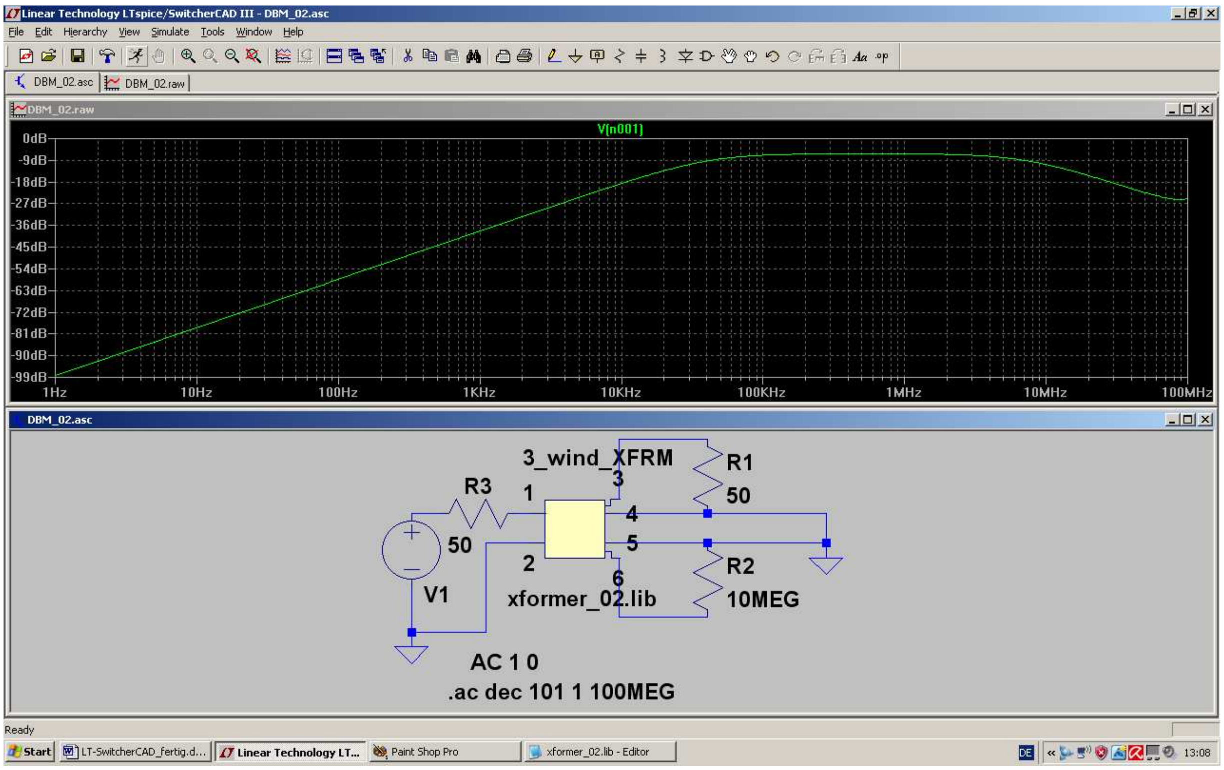
Disadvantages: Due to the missing power supply the complete „switching energy" must be delivered by the LO signal source. It demands high LO levels for rapid diode-switching and low distortion. There is no gain but always an attenuation of 5….8dB between the RF input and the IF output. This increases the noise figure in a system and the attenuation must be compensated by more „pre-amplifying".

15.3. The necessary Transformers

The „weak points“ of this circuit are always the 2 broadband transformers …and the special construction is a top secret guarded by the manufacturers! The transformers will in any case limit the bandwidth and the upper frequency.

So let us study an example with LO frequency = 1MHz and RF frequency = 100kHz. The lower cutoff frequency of the transformer should therefore be below 100kHz and the upper frequency limit min.10….20 MHz.

We use our well known „xformer_02“ with 3 identical windings and vary the properties to reach these specifications.
But remember: when the mixer is operating, always only one of the two secondary windings is in action, while the other is “idling” as long as the active diode pair is switched ON! So in the transformer simulation only one secondary winding is terminated by 50Ω while the other winding is simply feeding a 10MΩ-Resistor. This avoids the error message “node is hanging free”.
The main winding inductance is 100µH, copper resistance is 1Ω and winding capacitance is 1pF:



Now let’s tackle the complete mixer!

15.4. DBM Simulation with ideal Transformers

From the left comes the LO signal to switch ON and OFF the Schottky diodes (sine wave, peak value = 2V, frequency = 1MHz). From the right comes the information as RF signal (sine wave, peak value = 20mV, frequency = 100kHz.

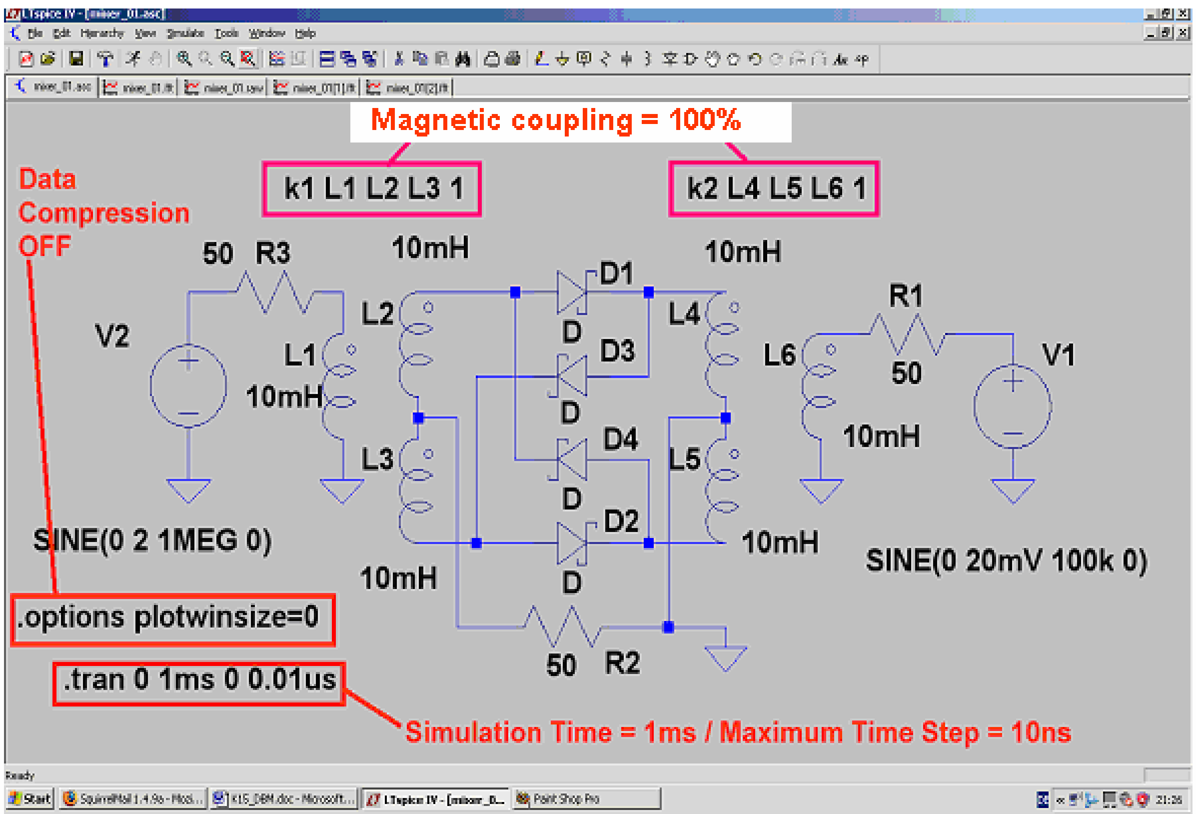
Both ideal transformers use 3 separate windings(each L= 10mH) which are magnetically coupled. The coupling factor is 1 (= 100%).

The Schottky diode comes from the LTspice part library. The symbol is named “schottky”.

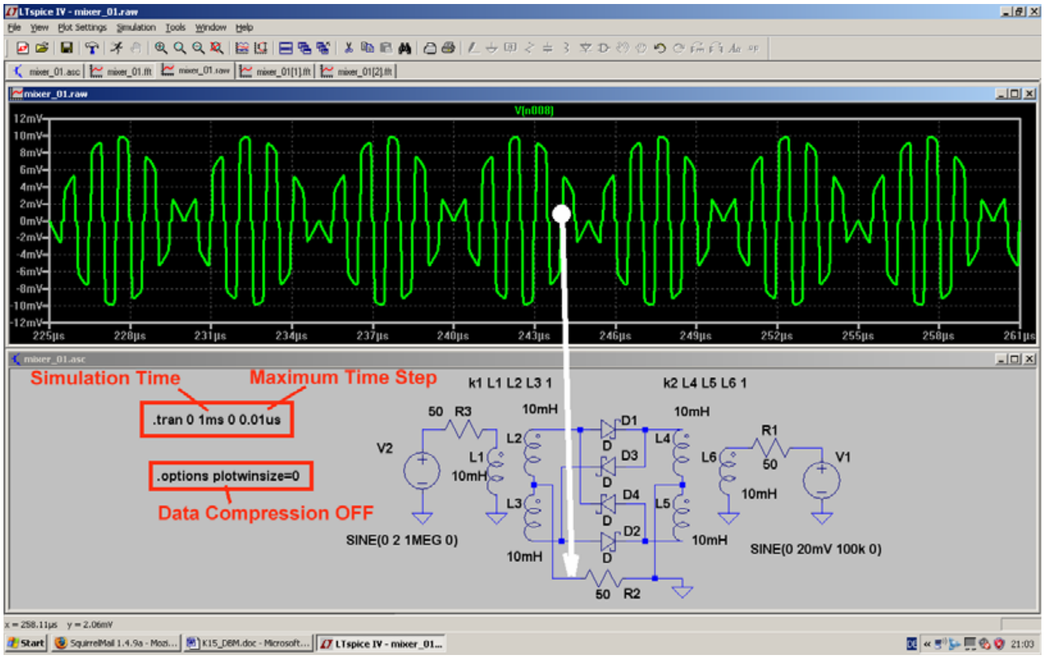
A simulation time of 1ms gives a frequency resolution of 1/1ms = 1000Hz. So for a number of 100 000 true samples a maximum time step of 1ms / 100 000 = 10ns must be used. This time step corresponds to a minimum sample frequency of 1/10ns = 100MHz and so no aliasing effect should be found up to a signal frequency of 50MHz.

Data compression is switched OFF by the spice directive **.options plotwinsize=0**.

This is the screen before starting the simulation:



And so the screen looks like this after the simulation of the output voltage at resistor R2 and zooming

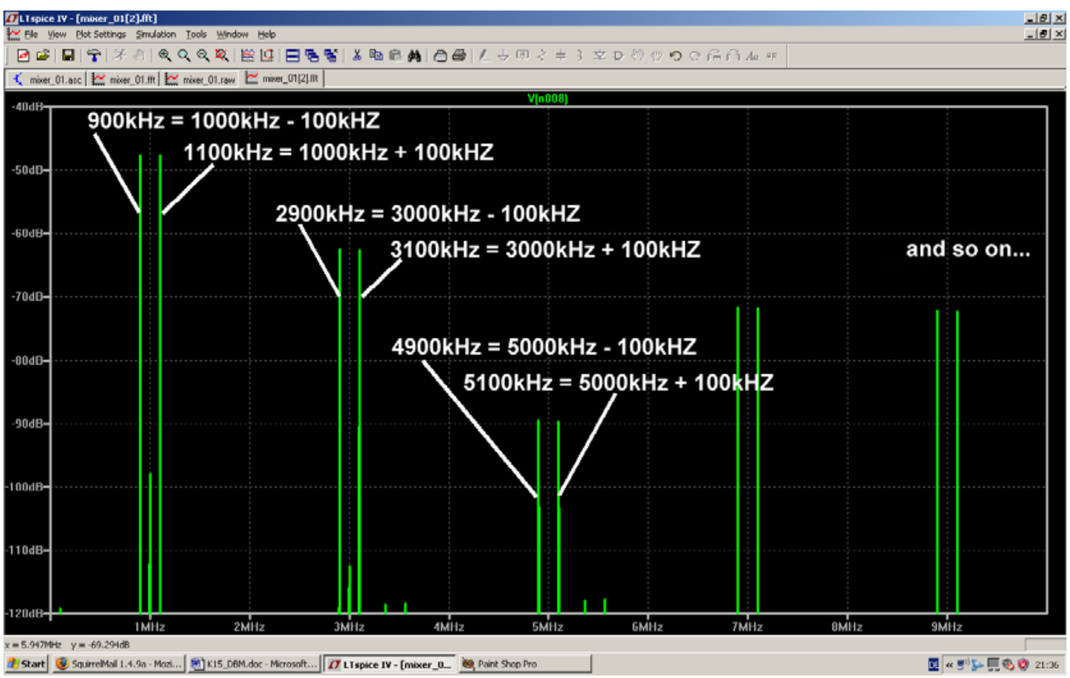


The polarity changing of the RF signal (caused by the LO signal) can clearly be seen. But: the „ON" voltage of 0.3V of every Schottky diode causes distortion at every zero crossing.

Lastly let us look at the FFT result for the output voltage. So right click on the time domain curve, open "View" and "FFT" and use **65536 samples** in time.

Press OK and modify the scaling of the diagram axis:

Frequency axis: **linear ; start frequency = 0Hz ; stop frequency = 10MHz ; tick = 1MHz**
Vertical axis: **dB scaling from -40dB …..-120dB with a tick of 10dB.**



Here comes the confirmation of theory:
At every LO harmonic frequency (3MHz ; 5MHz ; 7MHz ; 9 MHz…) we find a signal pair consisting of the sum and the difference frequeny of the RF signal and the harmonic.

Neither the RF signal nor the LO signal nor the LO harmonics should be found at the output.

16. Project 12: Simulations with Digital Circuits

16. 1. What you should know before you begin

Sorry, but this information is not described in the online help or the included examples. So you need to gain some of your own experience (..also gained through the use of other SPICE-programs) by exploring the problems and to master these non-intuitive constraints .

The most important rules for the „normal user“ are:

The included library **„[Digital]“** contains only **„ideal fundamental blocks“.**
Each of them always has 8 pins!

The reason for this is:

a) You always have several input pins (e.g. an AND-Gate has 5). Please leave all non-used pins open-circuit (unconnected). They will be ignored for the simulation.

b) Normally you will get **not only the desired output signal** but also the **inverted signal**.

c) Logical levels are „**0 Volt“ for „Logical Zero” and „+1 Volt“ for „Logical 1“**. The internal threshold voltage is 0.5V.

If you need to work with other voltage levels (like TTL = 0V /+5V), then right click on the part symbol, open the line „value“ and write the following entry:

Vhigh=5V   Vlow=0V

d) **Output pins should not be hanging free in the air.** So please **add a label** or terminate them by a 10kΩ – resistor to ground for a troublefree simulation.

e) The supply voltage pins are completely missing….you are working with “ideal parts“.

Use the „PULSE-“ or the PWL voltage as input signal.
Minimum value is 0 Volt, maximum Value is = +1Volt (…other values are handled as discussed above…)
As rise and fall time use 1ns.

To see your simulation results open separate diagrams by right clicking on the result screen and choosing

Add Plot Plane

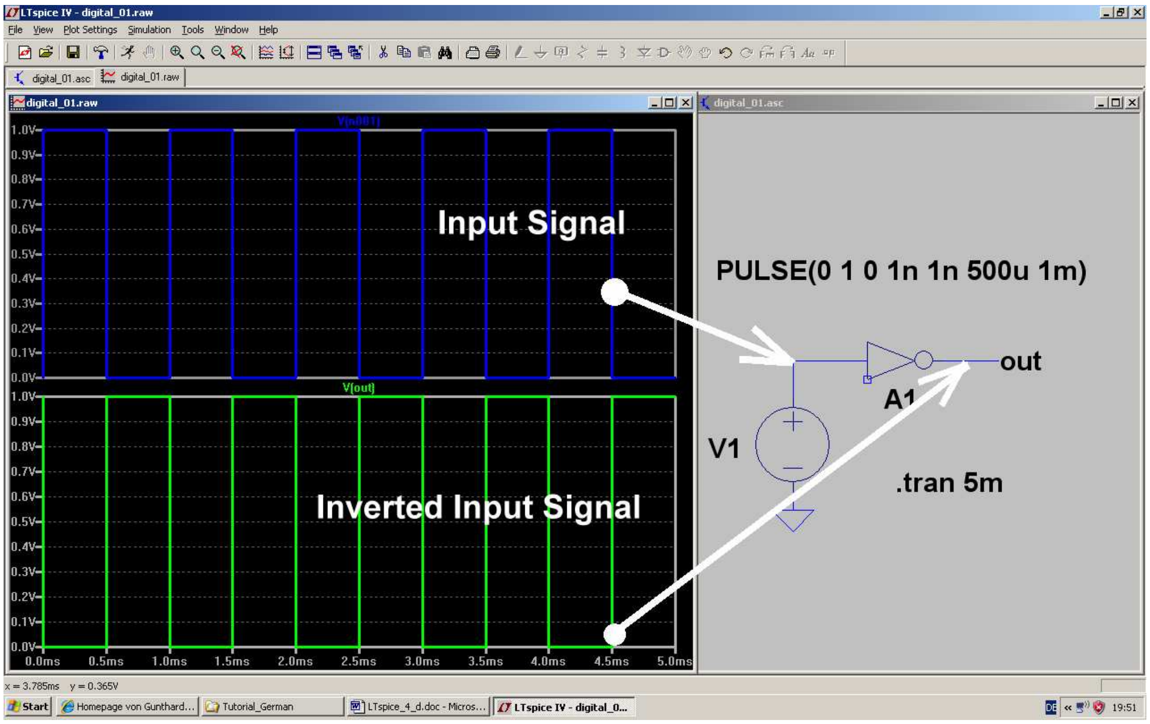
Use as many separate diagrams as you need to present signals -- otherwise you go crazy after a short time. Filling such a plot pane is very simple: click right on the diagram and choose “Add Trace”.

16.2. Simple start: the Inverter (= NOT)

Get the part as „**Digital / INV“** from the part library and use as the input signal a symmetric pulse voltage (Umin = 0V / Umax = +1V / Rise Time = Fall Time = 1ns / Pulse Length = 500µs / Period Time = 1ms). Simulate for 5ms.

Connect the Voltage Source to the upper input pin and leave the other input pin open-circuit.

Now simulate, open two diagrams by using „**Add Plot Plane**“ and view the input and the output voltage. You will get the following screen when selecting „**Tile vertically**“ in the „**Window**“ menu:

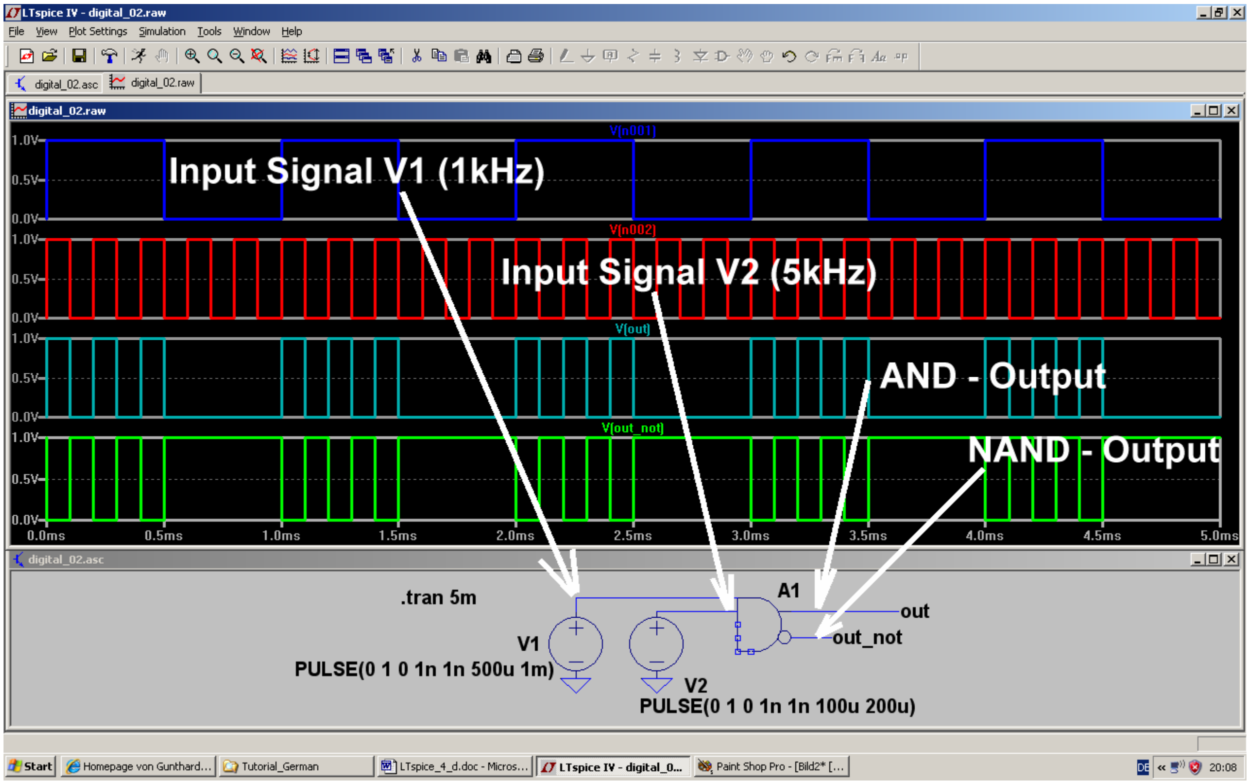


16.3. AND- and NAND Gate

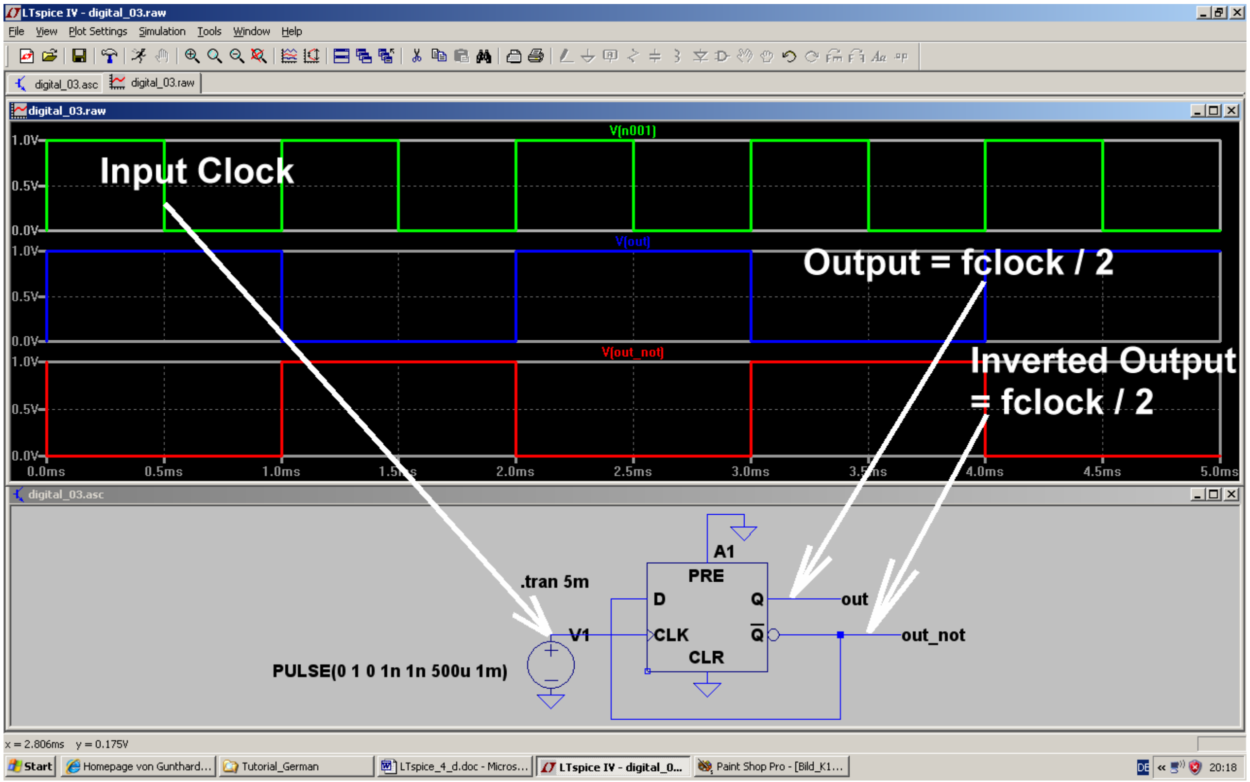
Let us simulate an AND Gate with 2 inputs. One input is fed by a symmetric pulse signal of 1kHz, the other input with a symmetric pulse signal of 5kHz. Simulate for 5ms.

As already mentioned, leave all unused input pins open-circuit and use a label for the output pin.

Here is the simulation result:



16.4. D Flipflop

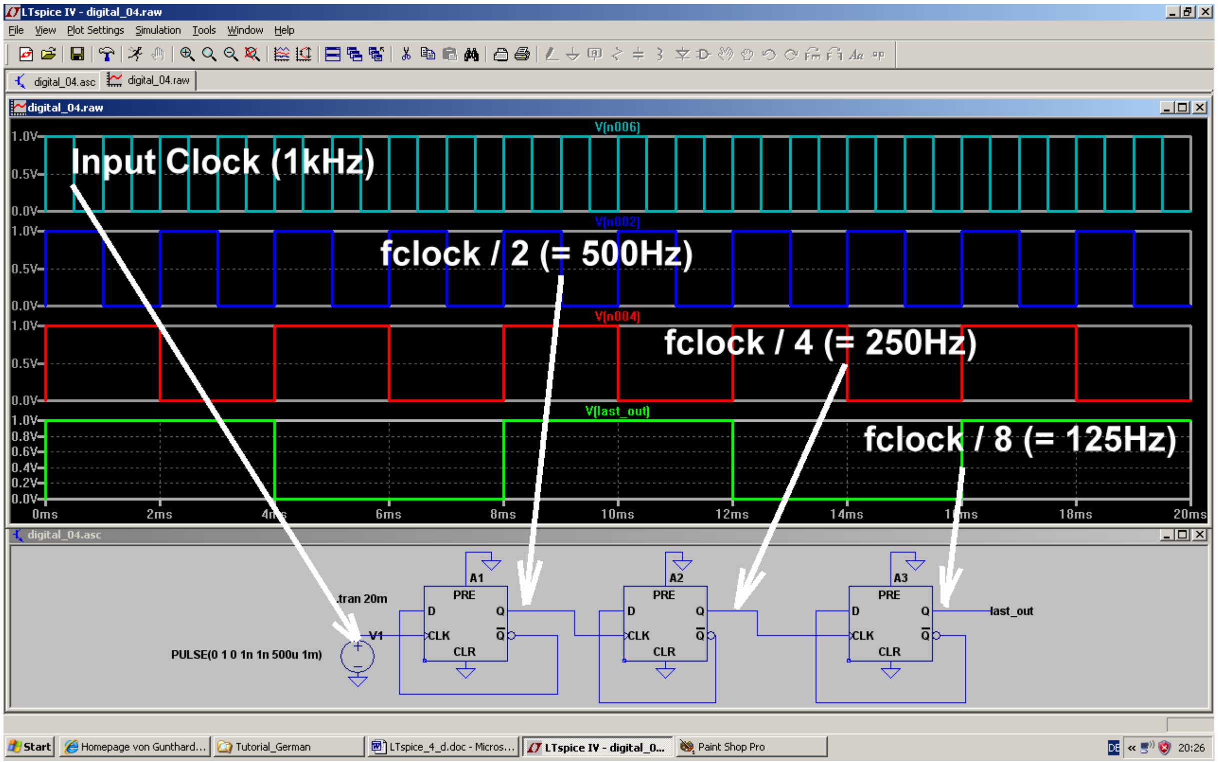


This is the well known „Binary Divide by Two Frequency Divider". At the output you get one half of the input clock frequency. The inverted Q-output is connected to the D-input pin.

„Preset" and „Clear" must not be active and so they are connected to ground.

16.5. Three Stage Frequency Divider with D flipflops

This is a very simple task because you only have to connect three D-stages in series and wire them.
Simulation time is 20ms, the input clock is a square wave of 1kHz.



If anyone has noticed that this chapter is short in comparison to the others: you are right. The reason is simple: in the LTSpice library the complete TTL and CMOS part collection is missing. This is really a pity when you consider that all these interesting circuits and schematics and problems could be simulated.

But there is a way -- this link:

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

It's the address of the LTSpice usergroup and there you find all the missing parts and symbols of the 74xx-families.
Thanks to Helmut Sennewald who has done the work of creating these important parts. Once more: Thanks, Helmut, for all your personal effort!

But:
This is a community and so you have to register first to get your ID and password. (no problem, it is free). If you do this you will be astonished and pleased: you'll find extensive Online discussion consisting of calls for help, questions, answers, remarks, advice, offers, problem solutions....and a lot of nice and kind people, always ready and willing to help.

So why wait? Go and see.

17. Project 13: Noise Simulation

17.1. Fundamentals

Please note
The following 3 chapters are part of my article

„Practical Project: Noise factor measurement with older Spectrum Analysers, Part 1 and 2“.

Published in „VHF –Communications”, Issues 2008-Q1 and 2008-Q3

17.1.1. „Noise“ -- where does it come from?

That can be answered very quickly and precisely: in each electrical resistance where a current flows and electrons move, there exists noise. As soon as heat comes into play (that is always the case above absolute zero), electrons have an independent existence. They move ever more randomly and not directly from minus to plus. They collide, rebound, are hurled forward or off to the side….. This makes the current vary irregularely by small amounts due to the influence of heat. This effect is called thermal noise. Even if no outside voltage is applied these independent movements of the charge carriers, due to heat, develop a small open circuit voltage V_{noise}. This can be computed as follows:

$$V_{noise} = \sqrt{\frac{4hfBR}{e^{\frac{hf}{kT}} - 1}}$$

Where
h = Planck's constant
k = Boltzmann's constant = 1,38 x 10^{-23} J / Kelvin
T = absolute temperature in Kelvin
B = bandwidth in Hz
f = center frequency of the band in Hz
R = resistance value in Ω

This seems terribly complicated and not practical, but the following simplification can be used without problem to at last 100GHz and temperatures down to 100K:

$$V_{NOISE} = \sqrt{4kTBR}$$

Changing this formula around it suddenly looks much simpler:

$$\frac{\left(\frac{V_{NOISE}}{2}\right)^2}{R} = kTB$$

a) This is a simple indication of power! Each resistance, independent of its resistance value, produces an „Available Noise Power“ proportional to kTB.

b) The resistance should be regarded as voltage source consisting of V_{NOISE} and a noise free *internal resistance* R to generate the noise power. A noise free load resistance with the same value R is connected to this source. Because voltage across the load resistance is then half V_{NOISE} the load resistance receives the Available Noise Power kTB.

c) This noise power increases linearly with the absolute temperature of the circuit and the voltage with the square root of the power. The spectral power density (power per Hz) is independent of the frequency. This is called “White Noise”.

Important:
Receiver systems are nearly always specified in terms of „power levels“ instead of voltages. These are logarithmic measurements so that the gain can be calculated by adding levels instead of multiplication. The most familiar unit is “dBm” -- that is not a voltage but a power rating in relation to the system reference resistance:

P_0 = 1 milliwatt at the system resistance

Thus

Power level in dBm = 10log (power value / 1mW)

If the noise power “kTB” is considered in more detail, an interesting simplification can be introduced:

kTB = (kT) x B = (Noise Power Densitiy) x (Bandwidth)

The Noise Power Density „kT“ represents the power in every Hz; this must be multiplied by the bandwidth in order to calculate the noise power produced. Converting this to a power level calculation the following formula should be committed to memory:

Each resistance produces at ambient temperature (T_0 = 290K) an available noise level and thus an available noise power density of

-174dBm per Hz of bandwidth

For a bandwidth larger than 1 Hz:

$$\boxed{\textbf{Available noise power level in dBm = -174dBm + 10log(bandwidth in Hz)}}$$

As a simple example:
The no load noise voltage V_{NOISE} at the terminals of a 50Ω resistor for a bandwidth of 100kHz will be calculated:

Noise level with matching = -174dBm + 10log(100000) = -174dBm + 50dB = -124dBm

That results in an available noise power of

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

That will produce a voltage across the 50Ω resistor of:

$$\mathbf{V_{NOISE} = \sqrt{P \bullet R} = \sqrt{50\Omega \bullet 4 \bullet 10^{-16}\,W} = 141nV}$$

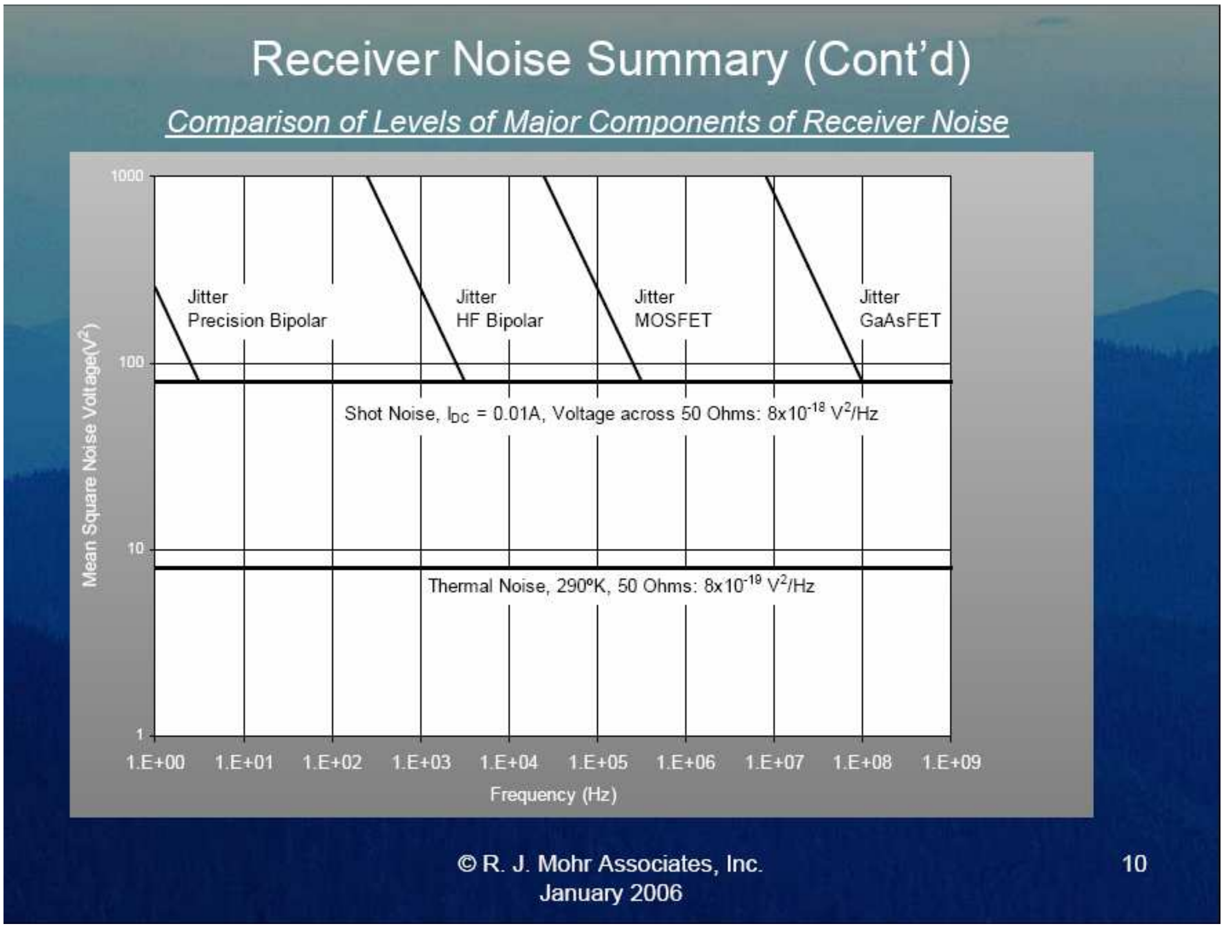
So the open circuit voltage is twice this value, thus 282nV.

17.1.2. Other Sources of Noise

Each active component (valve, bipolar transistor, barrier layer FET, MOSFET, HEMT, etc.) produces two additional types of noise:

a) ***Shot-Noise*** is produced in valve diodes and PN transitions by distortions of the current flow when crossing potential differences . It produces wide band, white noise.

b) ***Flicker-Noise*** or „1 / f – Noise" (= sparkling noise) results from defects in the crystal structure caused by impurities. They lead to short pulse type fluctuations in the current flow that produce a spectrum whose power density decreases with rising frequency. There exists a corner frequency and it is interesting to see how this differs between active components. The following figure is a good representation. The reference material in the homepage of Mohr Associates should be obtained from the Internet. It contains a precise but compact introduction to the subject.



Before semiconductors, gas discharge tubes were used as sources of noise, they produced a very wide band plasma noise.

The following section shows how the different kinds of noise in a circuit can be considered and how they can be summarized by only one parameter.

17.1.3. Noise Temperature and Noise Factor for a Twoport System

When components of a communicatons system are specified they usually have the same system resistance (normally 50Ω, but for radio, television and video systems it is 75Ω) and signal quality information required for correct transmission. This is often expressed by the SINAD value (= Signal to Noise and Distortion). The maximum signal level is limited by over modulation, distortion, intermodulation etc. The “noise floor” (in addition, other disturbances e.g. cross modulation from other channels) gives the minimum signal level in the system. Therefore each component must be specified by its S-Parameters as well as its noise behaviour.

When referring to noise, the following terms are usually evaluated (unfortunately not all authors use the same terminology):

a) The **„equivalent noise temperature T_e“** of a component expresses the self-noise of the component by an additional temperature rise for the noise source resistance. The component itself is thought of as noise free.

b) The **„NOISE FIGURE (NF)“** indicates, in dB, how much worse the signal-to-noise ratio becomes after the signal has passed through the component.

c) The **„NOISE FACTOR (F)“** is like the Noise Figure as above, BUT NOT IN dB, only as a simple power relationship.

17.2. Simulation of the Spectral Noise Density

This sounds complicated, but isn’t. In the introduction we stated that every resistor delivers at his terminals an Available Noise Power of:

kTB = (kT) x B = (Noise Power Densitiy) x (Bandwidth)

The noise power densitiy is the power which can be measured in every Hz of the bandwidth. You get the total available power if you multiply this density by the valid bandwidth (…or by integration over the bandwidth if the noise power density varies with the bandwidth).

But very often one prefers to handle the voltage instead of the power -- because it is easier to measure voltages at different frequencies than power. So use the well known equation to jump between these two values:

$$\mathbf{Power} = \frac{\mathbf{Voltage}^2}{\mathbf{Resistance}}$$

by using the known system resistance „R“.

When You now calculate the square root of the Noise Power Density, you’ll get an expression for the „Spectral Noise Voltage Density”, usually given in

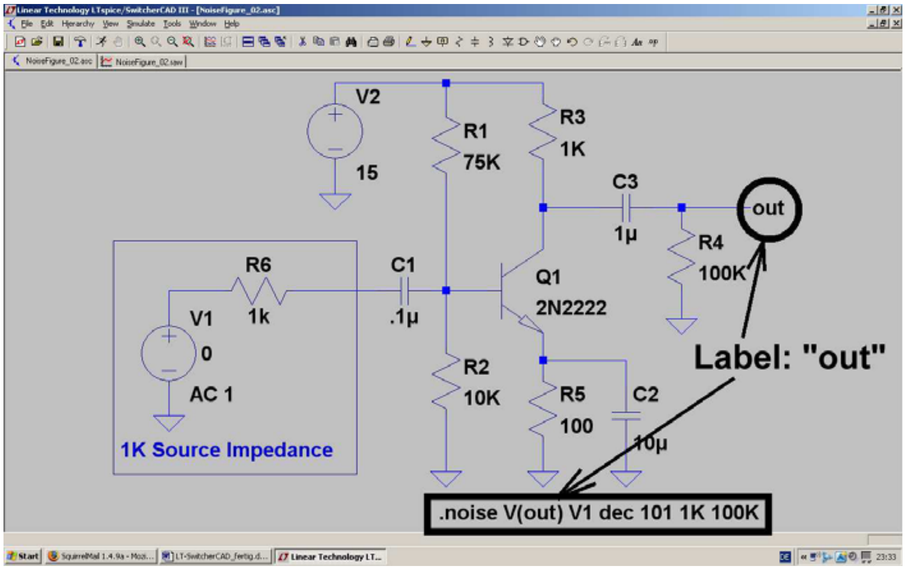
$$\frac{\mathbf{Nanovolts}}{\sqrt{\mathbf{Hz}}}$$

If you multiply this result with the factor

$$\sqrt{\mathbf{Bandwidth}}$$

you’ll get the total noise voltage.

Let use consider the included example „**NoiseFigure.asc**“ of the software -- you find it in the „Edudational“ folder and it is a one stage transistor audio amplifier, which we will modify a little bit for our purposes.



At the output we add the **label „out“** for the noise simulation.

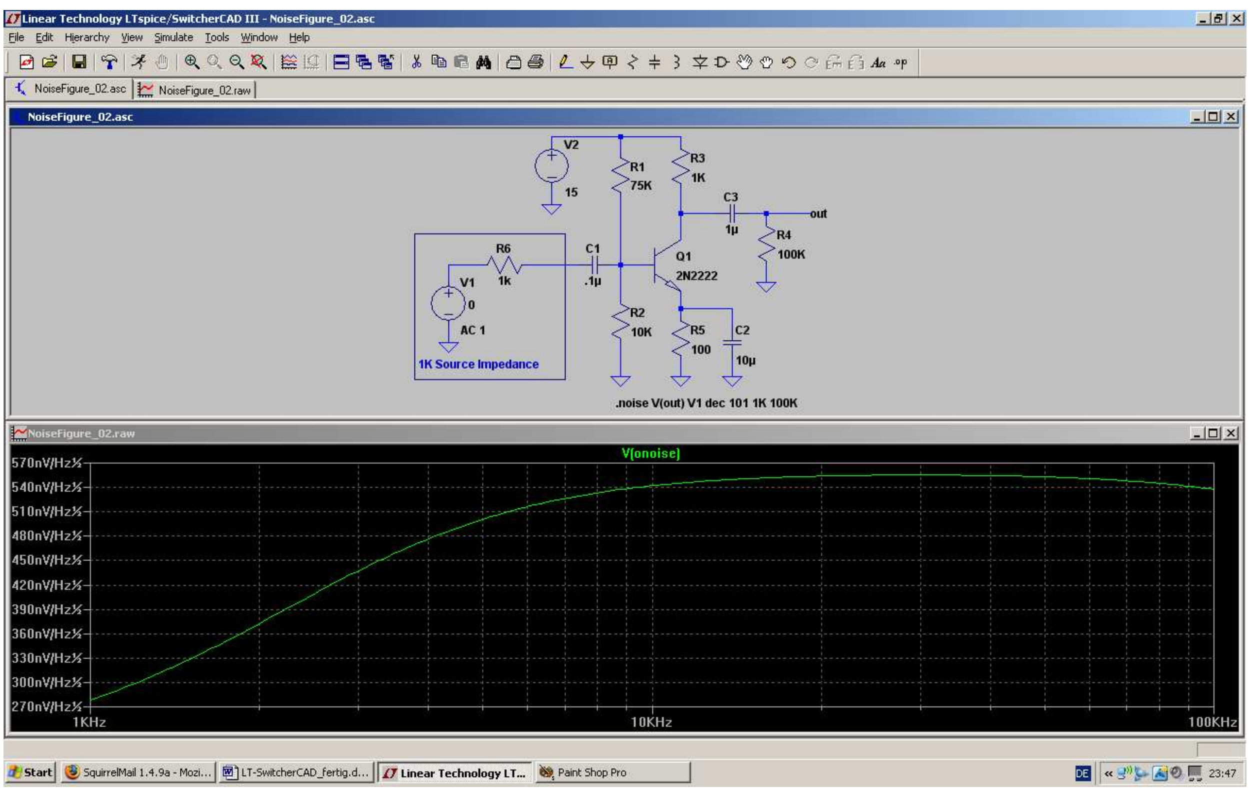
Use the path **„Simulation / Edit Simulation Command / Noise**“ to edit the necessary simulation command:

.noise V(out) V1 dec 101 1k 100k

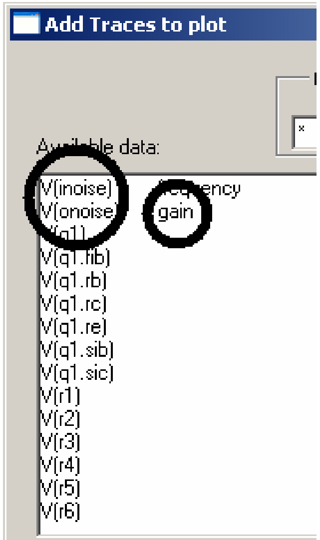
This means:

Simulate the spectral noise voltage density at the label „out“ and refer it to voltage source V1 with internal resistor R6 = 1k.
Use a decadic sweep with 101 points per decade from 1kHz to 100kHz.

And this is the result for the output noise voltage density V(onoise) when clicking with the „cursor probe“ on the output pin:



Please examine the scale and the units of the vertical axis....

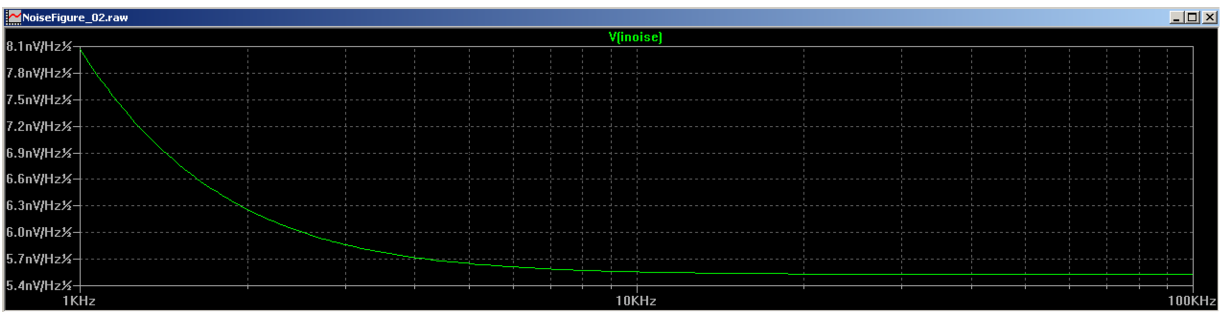


No problem, now to show the **noise voltage density V(inoise), referred to the input!**

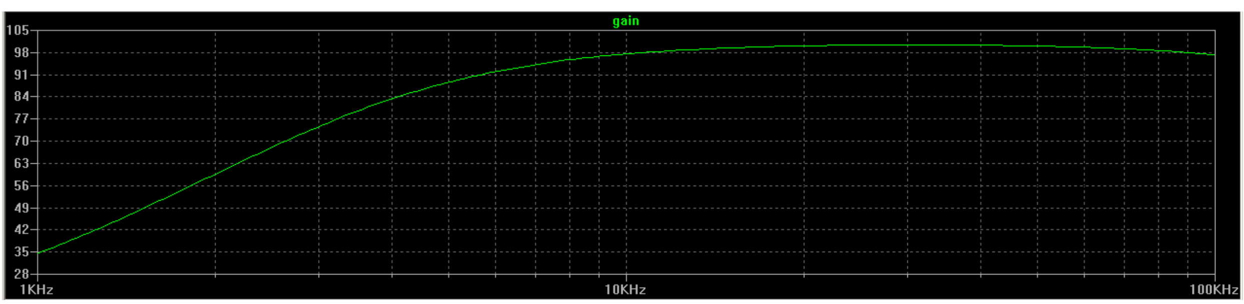
This is a voltage source which can be thought in series to the noise free source V1. So right click on the diagram, choose „**Add Trace**" and then „**V(inoise)"** in the list (see left figure).

If you examine the list then you'll find that not only V(inoise) can be shown but also V(onoise) and the **gain.**

This is the **input noise voltage density V(inoise) …..**



…and this is the **gain versus frequency**:



17.3. Simulation of the Noise Figure NF (in dB)

In communications systems many blocks are connected in series. So the reduction in signal quality in every „twoport" caused by noise must be known and is expressed by the

Noise Figure NF in dB

So the „**NOISE FIGURE (NF)**" ist the difference in db between the signal to noise ratio at the input and the output of the twoport.
For a noise free twoport the signal to noise ratio is identic at input and output pin and NF = 0dB

(Please note:
The input signal source V1 delivers not only the desired communication signal to the twoport but also noise due to the internal resistance of the source. So if only this noise is amplified by the same factor as the communicaton signal, the signal to noise ratio at the output would stay the same as at the input. But because the block adds noise itself....)

If you change the NF-value in dB to a linear ratio, then you get the „noise factor F".

Now let us tackle the NF-simulation with LTSpice, following these steps:

a) To get „dB" at the result, we need the relationship

$$NF = 10 \times \log_{10}(\ldots..)$$

b) In the brackets we must insert the ratio of the total input noise power (= total output noise power, refered back to the input = P_{inoise}) to the noise power part which comes only from the source resistance (= P_{Source}).
If both powers are equal, then we have a noise free twoport and the noise figure NF is

$$NF = 10 \times \log(1) = 0\ dB$$

But in reality you will always have noise factor values larger than 1 = NF-values larger than 0 db....

The noise power produced by a resistor increases linearly with the absolute temperature, and T = 290K was always the standard value for every noise calculation. But now the global temperature is rising, climate is changing and so electronic people prefer calculations with T = 300K.
Now let us continue with the calculation of the power ratio:

$$\frac{P_{inoise}}{P_{Source}} = \frac{\left(\frac{V_{inoise}^{\ 2}}{4 \bullet R_{Source}}\right)}{k \bullet T \bullet B} = \frac{V_{inoise}^{\ 2}}{4 \bullet k \bullet T \bullet B \bullet R_{Source}}$$

„k" is the Boltzmann constant, „T" is the absolute temperature in Kelvin, „B" ist he bandwidth of the system, R_{Source} is the internal resistance of the input voltage source.

If we consider spectral power density then the bandwidth is automatically 1Hz. Then we can ignore the bandwidth "B" for the calculation and get the final formula which must be converted to a SPICE command:

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

And assuming a actual climate change contribution we use an absolute temperature of 300K:

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

Problem:

In a simulation with LTSpice it is only possible to calculate this formula (and to show the result) by the usage of a

PLOT command!

So we must write the following line for the calculation of the noise figure NF(1K) with a text editor, when the source resistor's value is 1k:

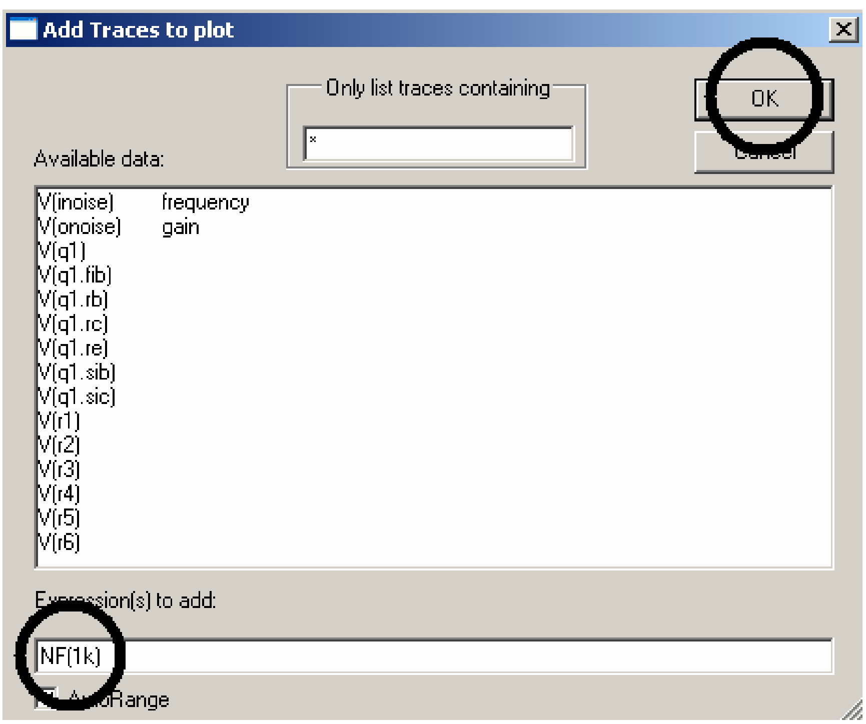
.func NF(1k) 10 * log10(V(inoise) * V(inoise) / (4*k*300*1k))

Save this line in a new file named

plot.defs

and save it in the LTSpice program folder.

Caution: After saving this file you must close LTSpice and start it again. Only in this manner the new „plot.defs" are included and used by the software!



The rest is simple: simulate the circuit, click right on the result diagram and choose

„Add Plot Pane"

to get an empty diagram.

Then click right on this new diagram an call

„Add Trace"

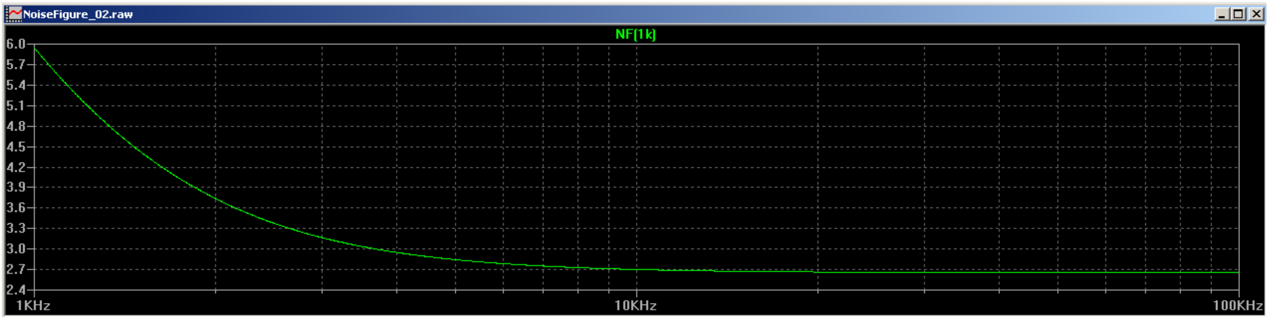
Enter

NF(1k)

In the empty command line and press OK

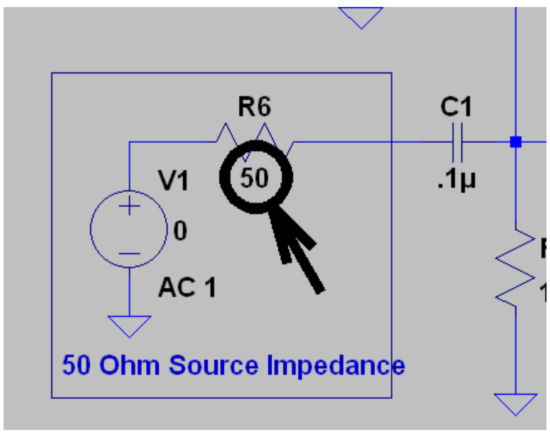
The result is not totally perfect, because

the „dB" calibration on the vertical axis is missing and this cannot be altered ...



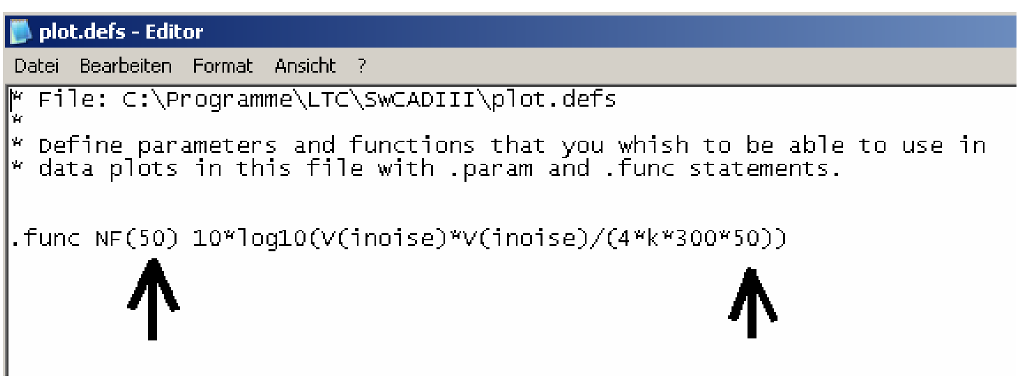
Task:
Communications systems generally use a system resistance of 50Ω. So please repeat the NF simulation for this value.

Step 1:



Change the internal resistance of the voltage source V1 in the schematic to 50Ω.

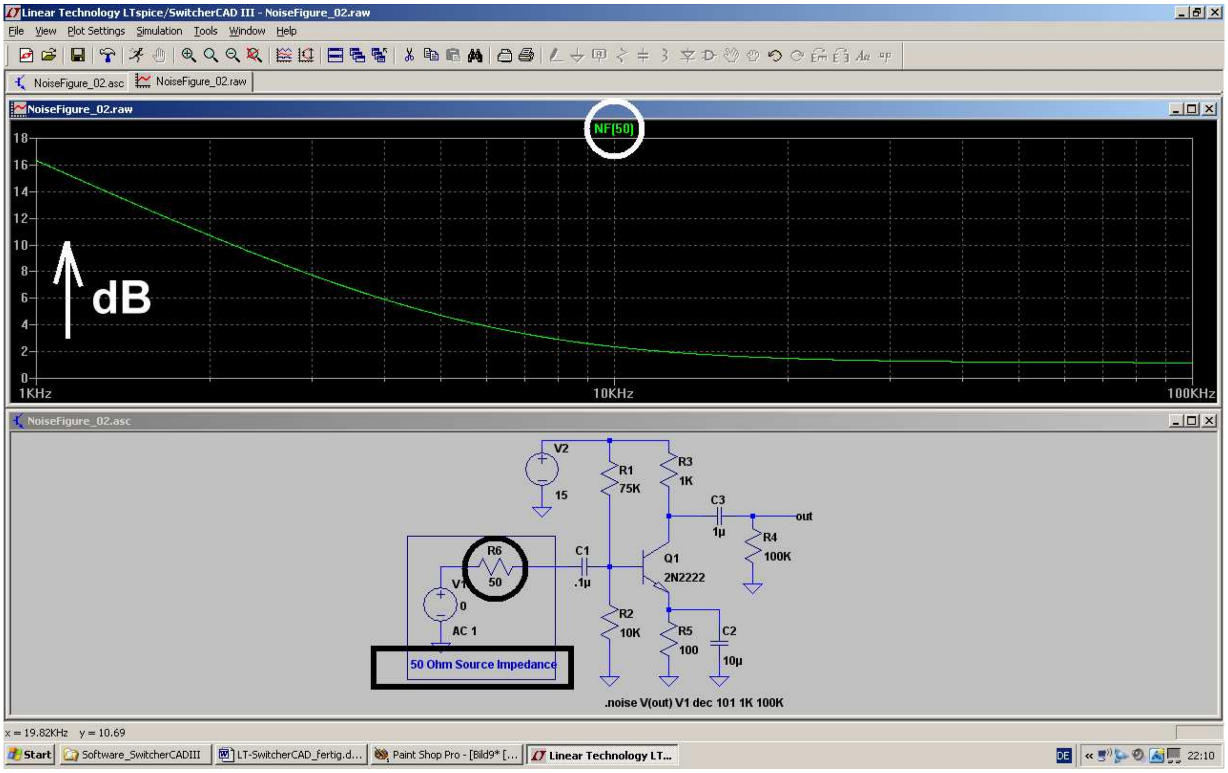
Step 2:



Open the file „**plot.defs**" with the text editor and modify the command line to „**NF(50)**" as shown in the figure. Do not forget to save this modified file correctly…

Caution:
Please close LTspice now and start it again. Only then the modification is accepted and used by the software. Otherwise you get a cryptic error message…

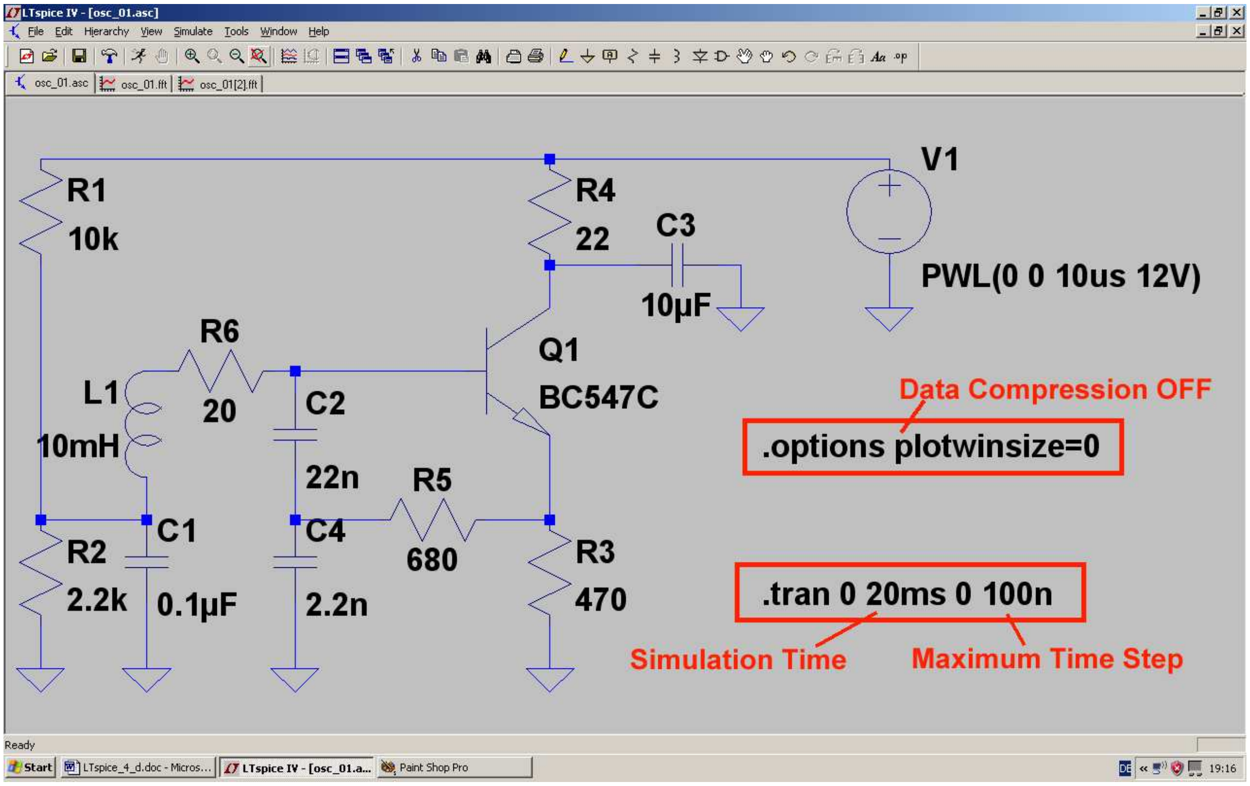
Step 3:
Simulate again, open a new „Plot Pane“ and then “Add Trace”**. Enter “NF(50)” in the line „Expression(s) to add“ and press OK. This is the result:**



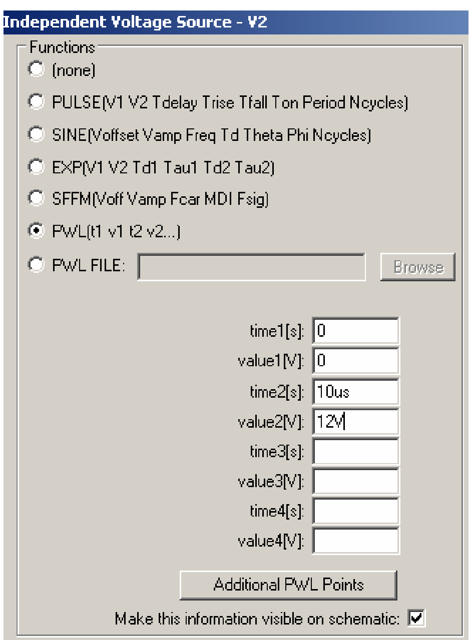
The vertical axis is now calibrated in „dB“, but this is not indicated….

18. Simulation of a Sine Oscillator

Let us have a look at a Colpitts sine oscillator using a collector-grounded NPN – transistor.
This is the schematic:



Note:



To start the (simulated) oscillation the circuit needs a little „push". This is done by programming a special property of the supply voltage:

„Rise linearly from 0V to +12V in a rise time of 10 microseconds".

We right click on the symbol of the voltage source, choose „**Advanced**" and use the PWL-function with the value pairs:

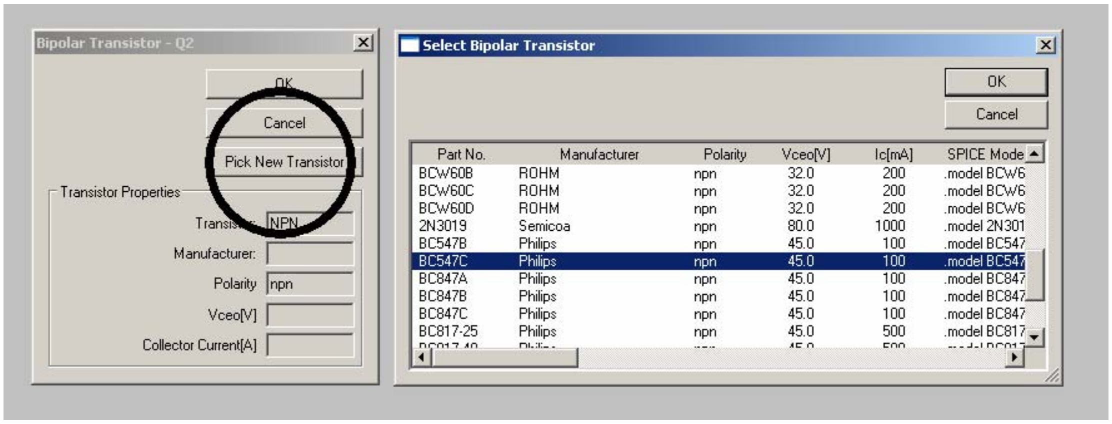
0V at t = 0

+12V at t = 1us

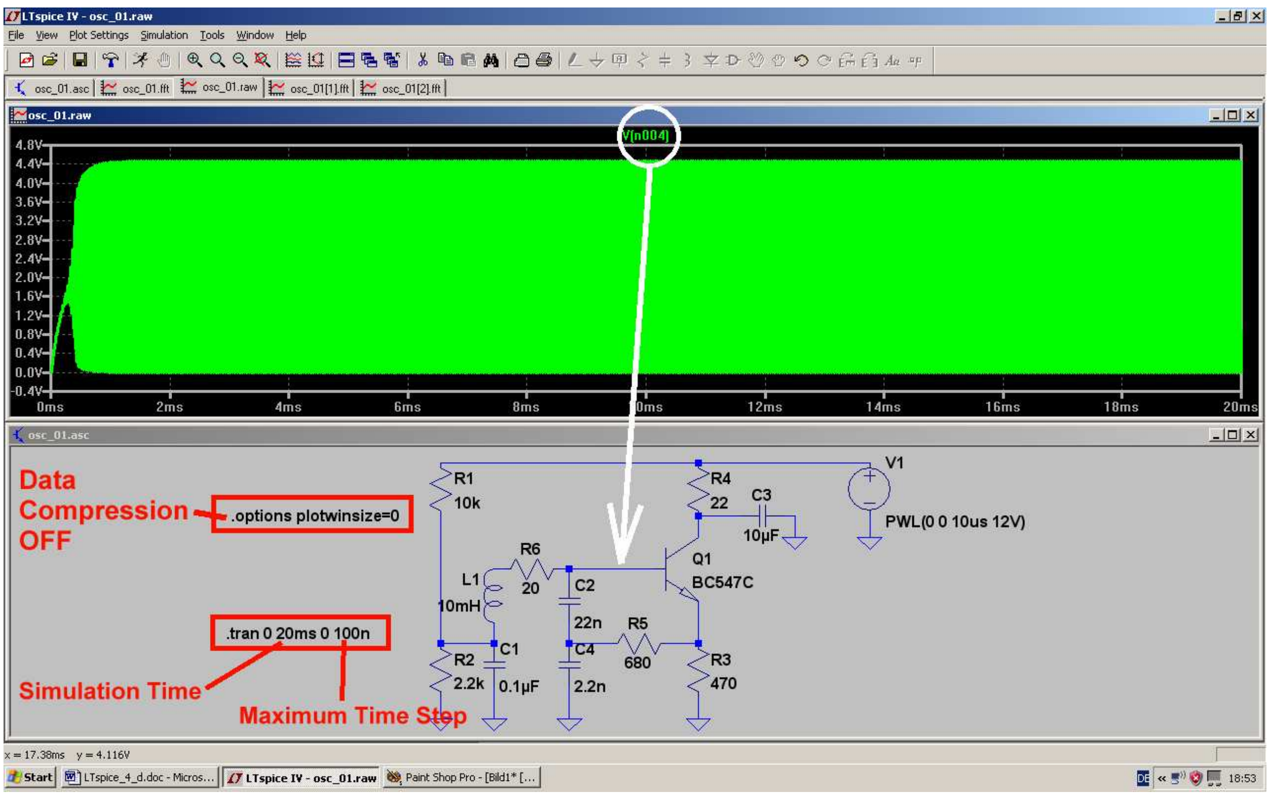
The „Simulation Settings" are set to „Time domain (Transient)" and the simulation time is **0….20ms. This gives a frequency resolution of 1/20ms = 50Hz in the spectrum.** The maximum time step is set to 100ns, so SPICE doesn't overlook the starting procedure described above. . This now gives a number of 20ms / 100ns = 200 000 sample for the FFT computation.

And the data compression must be switched OFF by the directive **.options plotwinsize=0**

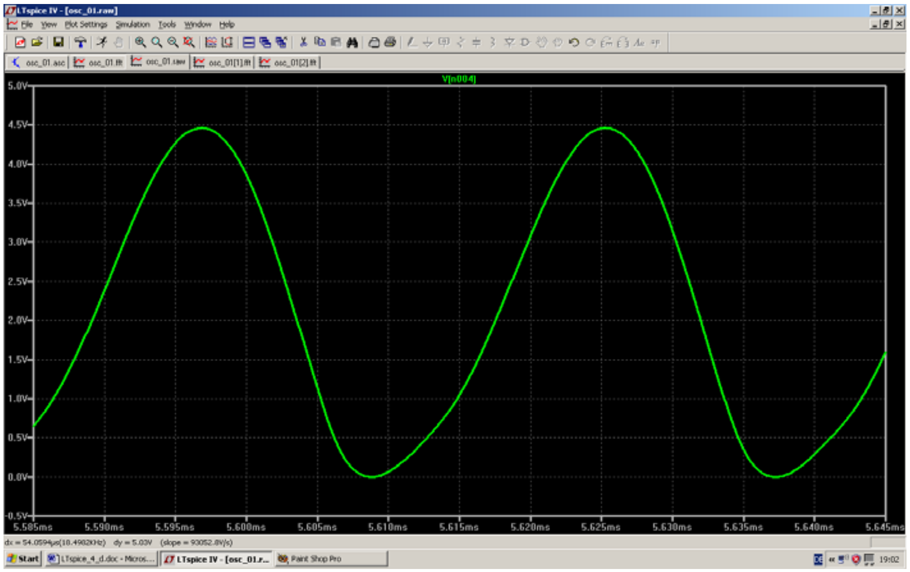
Now we need a transistor with the symbol name „**npn**“ from the part library. Search and place it on the schematic. As soon as you right click on the NPN transistor’s symbol, press the „**Pick New Transistor**“ button. Then find the **BC547B** which we want to use for our simulation.



Simulate the voltage at the base of the transistor to get this screen:

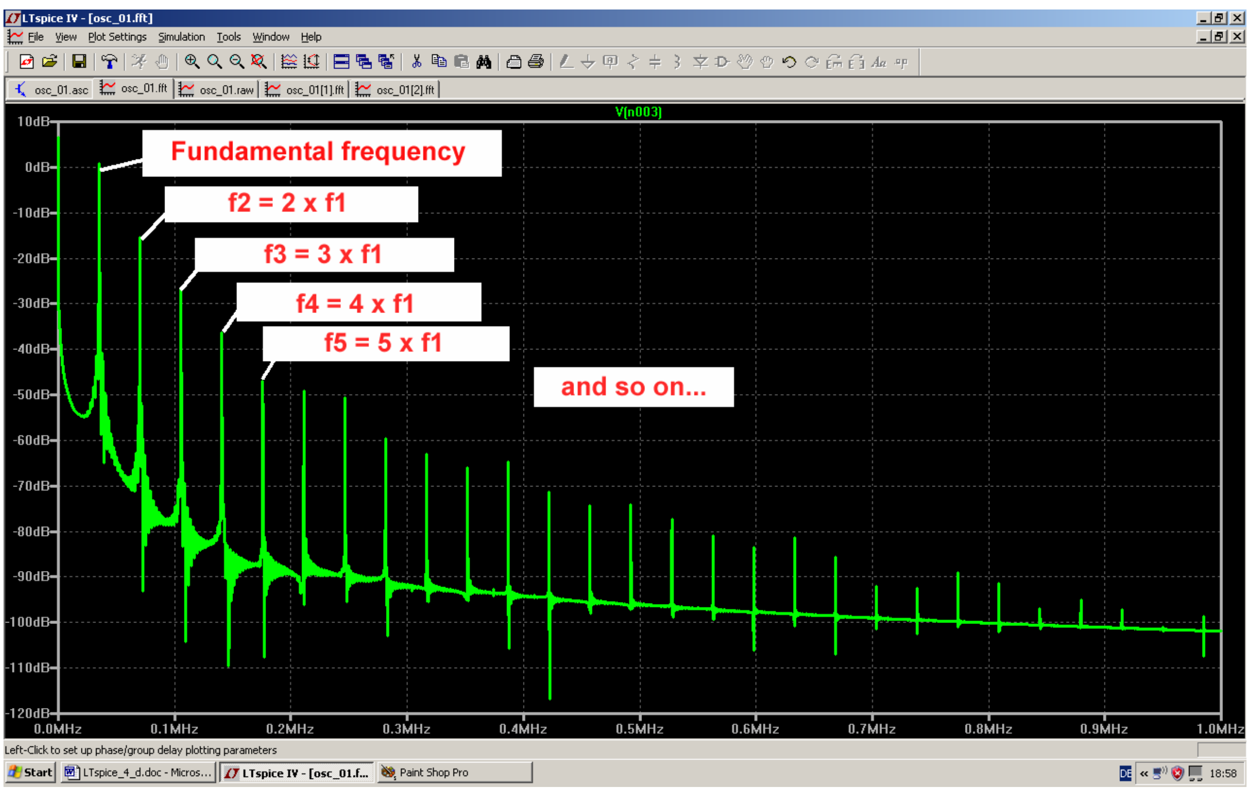


Use the zooming function to show the details of this voltage and to determine the time period for one cycle of the signal. Calculate the oscillating frequency. Examine the curve for distortion.



The duration of the period is 28ms and this gives a frequency of 35,7kHz.
But the curve is a little "clipped" on one side and this means that the feedback is too "strong". This can be seen when viewing the **frequency spectrum** of this signal.

So right click on the waveform viewer window and choose „View", then „FFT" with 131 072 data points in time.
At lastly show the curve with a linear frequency range from **0 to 300 kHz with a tick of 50 kHz**:



Now we can see the decreasing amplitudes of the harmonics with increasing frequency. Also the "fundamental oscillating frequency" can now be determined more accurately as 35.7kHz.

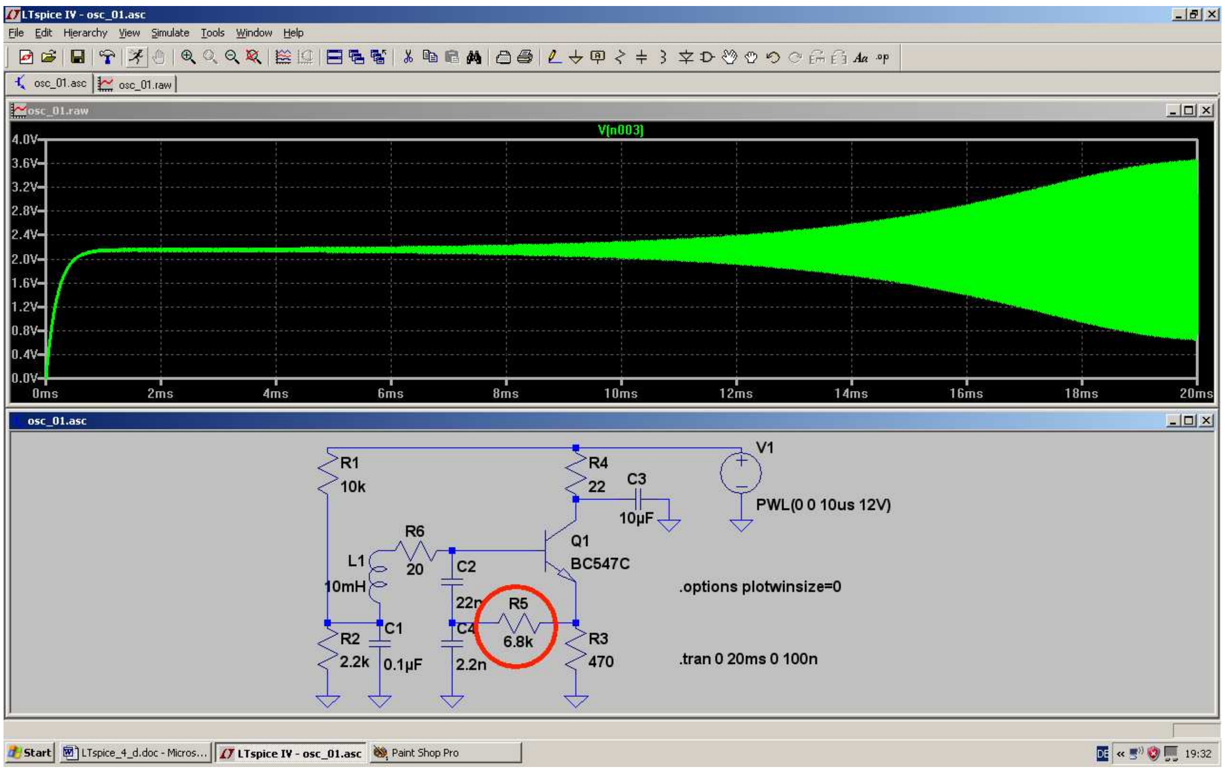
But the 2^{nd} harmonis is only attenuated by less than 20dB and this is caused by the unsymmetry of the signal (Remember: in a really symmetric signal you don't find any even harmonics….).
So any distortion of the symmetry can at once be recognized by testing for even harmonics in the spectrum.

Note:
Resistor R4 / 680Ω gives the positive feedback which causes the oscillation. A part of the emitter AC current is flowing into the resonant circuit through this resistor and „compensates the losses“.
So if you increase the resistor's value, the positive feedback will get weaker and weaker. This gives less distortion but the oscillation will suddenly stop.

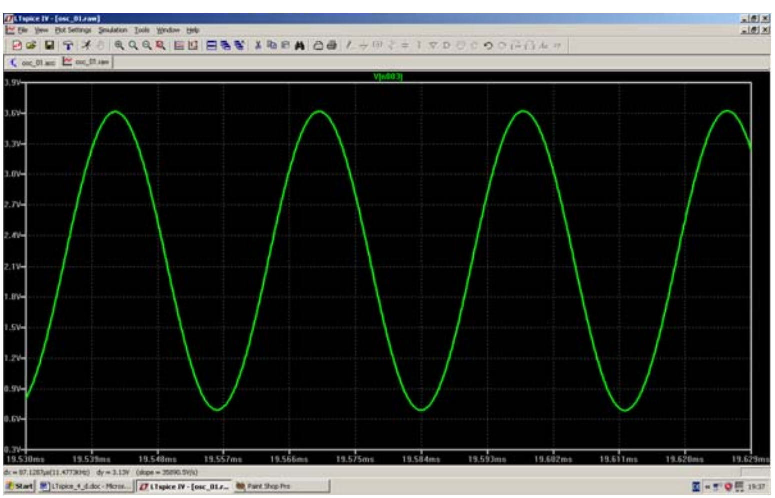
Task:
Try to find out this stop point by increasing the value of R4 and repeating the simulation.
Watch the improvement of the waveform and check the better attenuation of the harmonics by the FFT-simulation of the frequency spectrum..

Also be aware of the „oscillation start delay“ when reducing the feedback. When approaching the critical point you get signficantly longer delays before the oscillation starts.

(Note: with a resistor value of 6.8k you are not far from this point…please check it!)

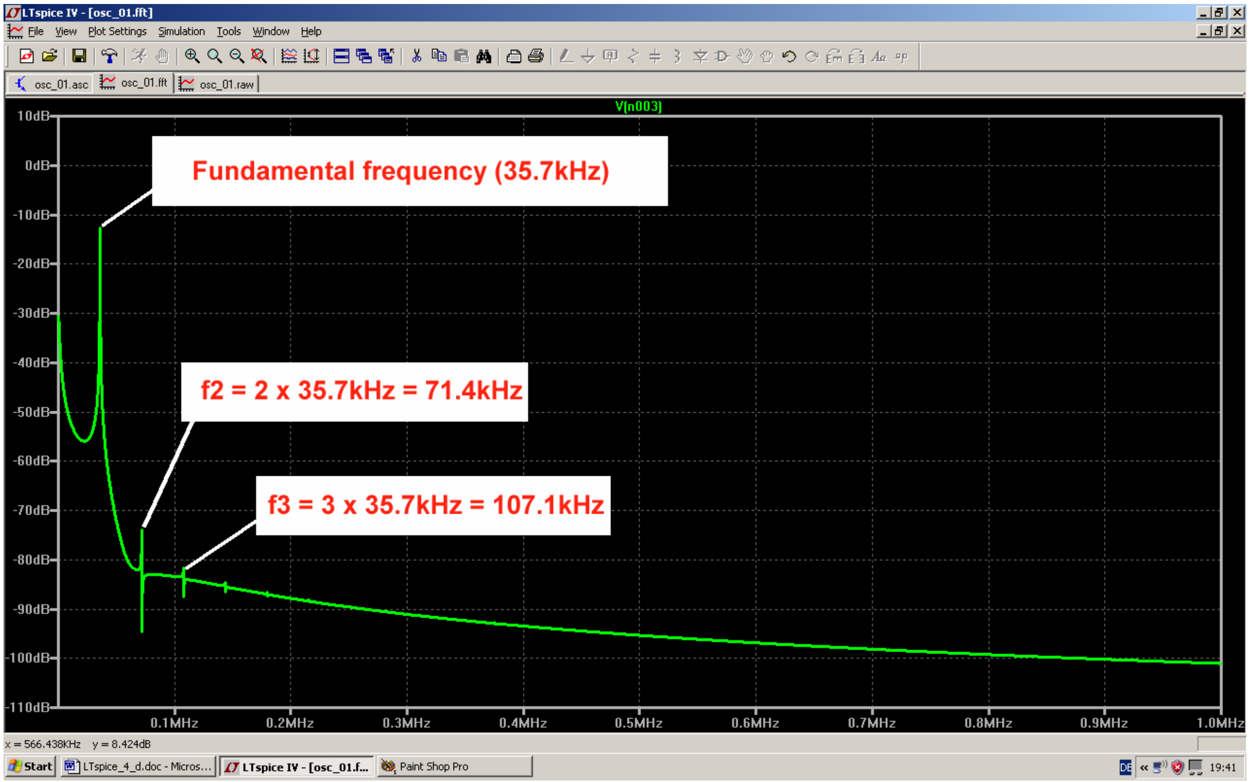


This is the result for 6.8k. The oscillation start delay is increasing….



…but now you get
a nearly perfect curve!

And the FFT spectrum proves this.



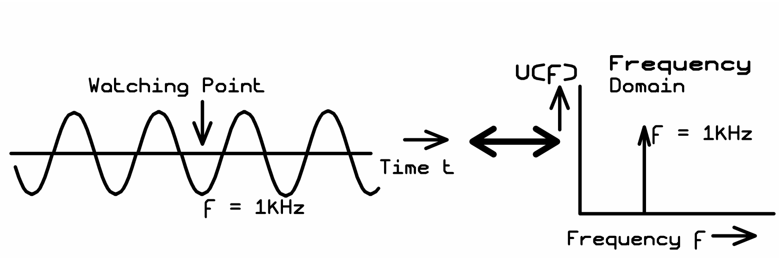
19. Signals and Harmonics

19.1. Fundamentals

If you ask anyone what signals are found on power lines all over the world, you'll always get the same answer: "a sine wave with a RMS-value between 110V and 230V and a frequency of 50 or 60Hz, depending of the specific country". But if you continue and ask:

"Is that all?"

Even specialists may give a wrong answer. OK, there are sometimes harmonics of the line frequency, but much more important are the effects caused by switching on or off. So let us look at the correct definition of a signal consisting of only one frequency:

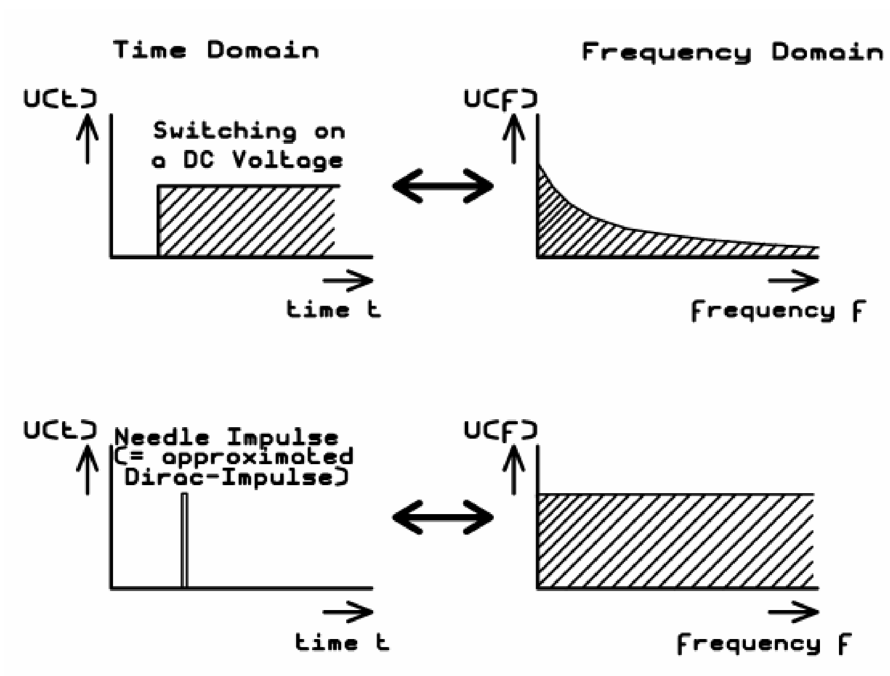


Only a pure continuous sine wave where neither the start point nor the stop point can be observed consists of only one spectral line!

This means that in every day usage you should always clearly distinguish between "**distortions of a signal**" and "**switching events including changes".**

Switching

Every switching ON or switching OFF of a signal or increasing or decreasing the amplitude produces a "signal collection" which travels away (with light velocity) from the point of generation. You know that as a "click" from a transistor radio when switching on the light in a room and you should know that in this moment you get the complete frequency range loaded with energy! Very low and low frequencies will travel away using wires, high frequencies prefer wireless travel. But normally the amplitudes decrease rapidly with frequency when "changing a state very slowly" and so TV and communication are still relatively unaffected.
Two examples:



Switching On or OFF of a DC voltage gives a spectrum with decreasing spectral power density at increasing frequencies.

But:
Very short pulses (with nearly Dirac form) have a constant power density over the complete frequency range.

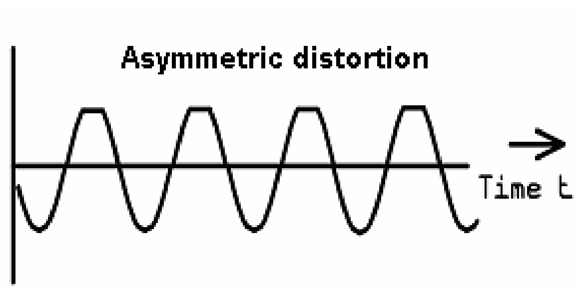
So you cannot avoid this "plop" because there are no gaps in the complete frequency spectrum!

Distortions of periodic signals

Please remember:

Every -- even the smallest! -- deviation from the perfect sine wave generates new additional frequencies!

These new frequencies are called “harmonics” and they are multiples of the fundamental frequency



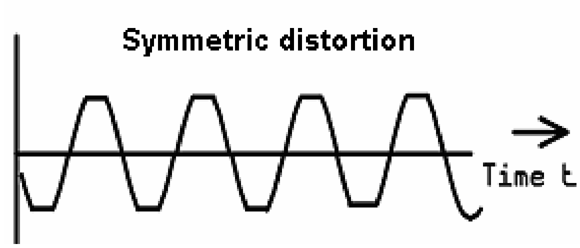
But there is another axiom:

If only one half of the curve is distorted then we speak of “**asymmetric distortion**” where you **find even and odd harmonics** in the spectrum.
This will also happen when the positive and the negative half is distorted in a different manner.

But when you find perfect symmetrical distortion, the spectrum consists only of the fundamental frequency and ODD HARMONICS!

This means that there are only the lines with

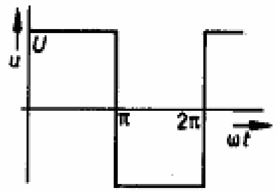
f1 / 3 x f1 / 5 x f1 / 7 x f1…



Note:
If a sine wave’s amplitude, phase or frequency is varied by another signal, you will also find new frequencies.
But this is the mystery of modulation…..

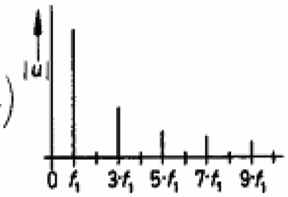
On the next page you find a little overview for different waveforms with the signals in the time domain, in the frequency domain and the formulae to compute the harmonics.

Square Wave

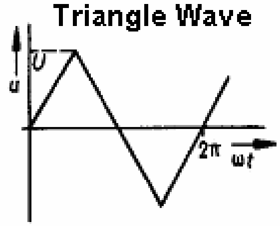


$$f(t) = \frac{4 \cdot U}{\pi}\Big(\sin \omega \cdot t + \dots$$
$$\dots + \frac{1}{3}\sin 3\,\omega t + \frac{1}{5}\sin 5\,\omega t + \dots\Big)$$

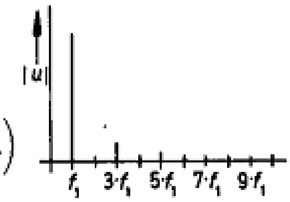
Frequency Domain



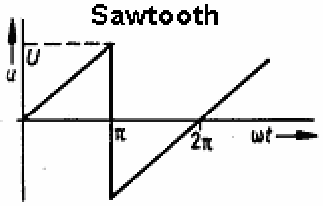
Triangle Wave



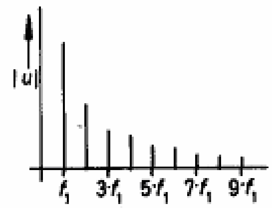
$$f(t) = \frac{8\,U}{\pi^2}\Big(\sin \omega t - \dots$$
$$\dots + \frac{1}{3^2}\sin 3\,\omega t + \frac{1}{5^2}\sin 5\,\omega t - \dots\Big)$$



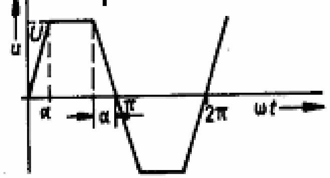
Sawtooth



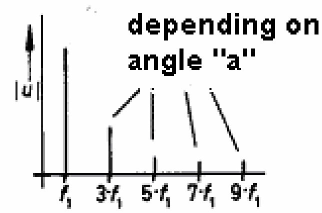
$$f(t) = \frac{2\,U}{\pi}\Big(\sin \omega t - \frac{1}{2}\sin 2\,\omega t + \dots$$
$$\dots + \frac{1}{3}\sin 3\,\omega t - \dots + \Big)$$



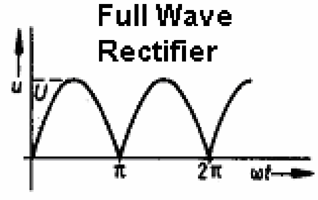
Trapezoidal Wave



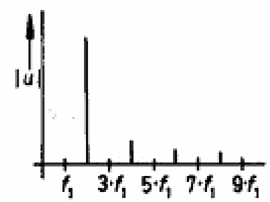
$$f(t) = \frac{4 \cdot U}{\pi \cdot \alpha}\Big(\sin \alpha \cdot \sin \underset{\text{tot}}{\omega t} + \dots$$
$$\dots + \frac{\sin 3\,\alpha}{3^2} \cdot \sin 3\,\omega t + \dots$$
$$\dots + \frac{\sin 5\,\alpha}{5^2}\sin 5\,\omega t + \dots\Big)$$



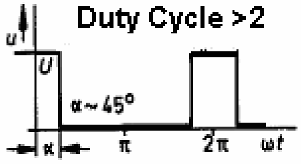
Full Wave Rectifier



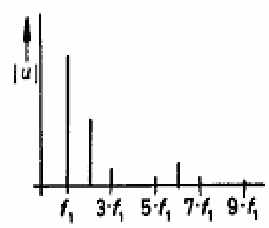
$$f(t) = \frac{4\,U}{\pi}\Big(\frac{1}{2} - \frac{1}{1 \cdot 3}\cos 2\,\omega t - \dots$$
$$\dots - \frac{1}{3 \cdot 5}\cos 4\,\omega t - \dots$$
$$\dots - \frac{1}{5 \cdot 7}\cos 6\,\omega t - \dots\Big)$$



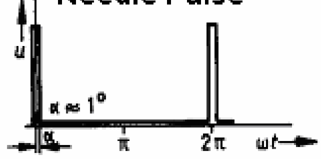
Square Wave Duty Cycle >2

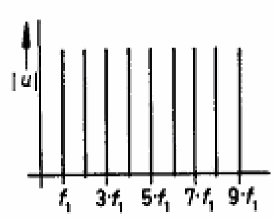


$$f(t) = \frac{2\,U}{\pi}\Big(\frac{\alpha}{2} + \frac{\sin \alpha}{1}\cos \omega t + \dots$$
$$\dots + \frac{\sin 2\,\alpha}{2}\cos 2\,\omega t + \dots$$
$$\dots + \frac{\sin 3\,\alpha}{3}\cos 3\,\omega t + \dots\Big)$$



Needle Pulse





19.2. Simulation of a Single Pulse Spectrum

Now let us test the statement in the last section and examine the following pulse signal:

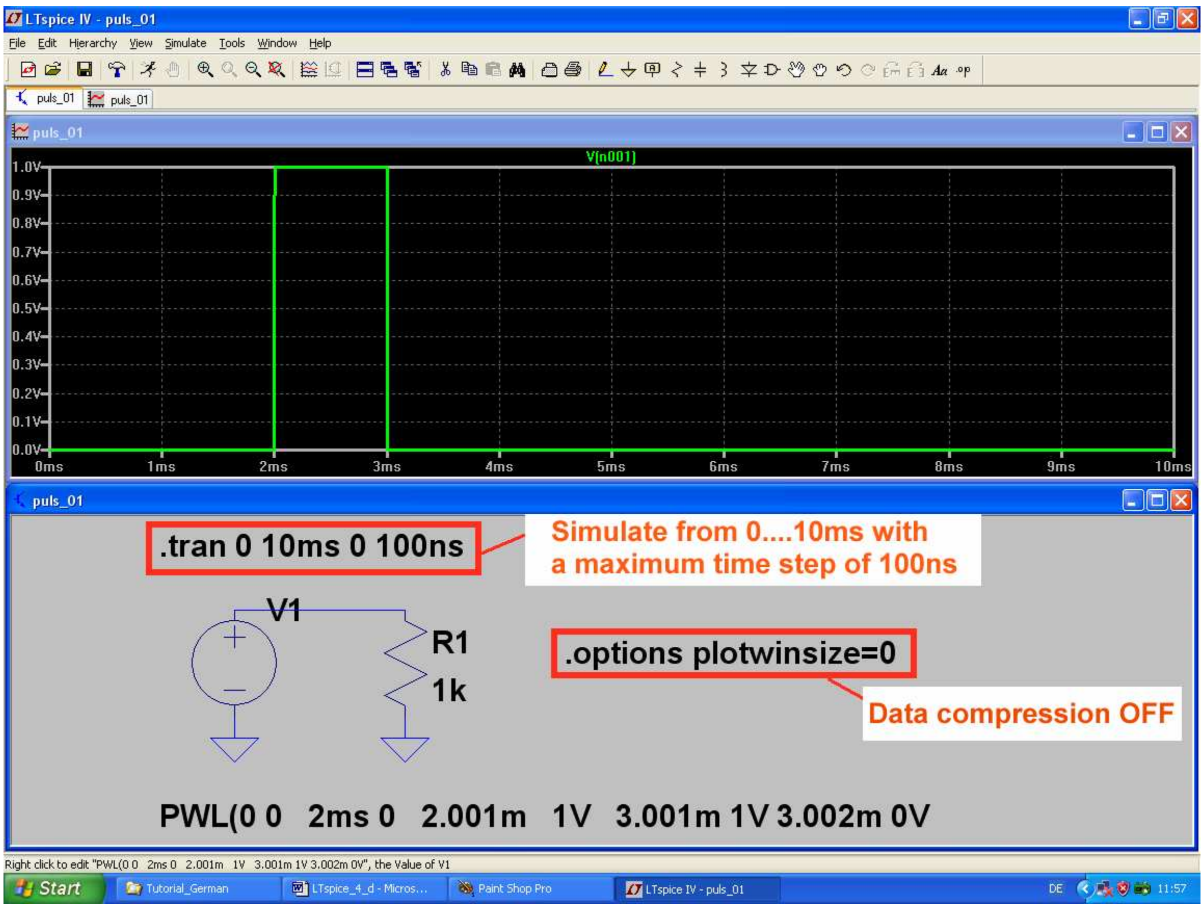
Umin = 0V ; Umax = 1V ; risetime = falltime = 1µs ; pulse width = 1ms.

The pulse signal (measured at R1) starts after a „**delay time“ of 2ms.** The simulation time is **0… 10ms** and the „maximum time step“ has a value of **100ns.** So we get 10ms/100ns = 100 000 real samples of the simulated curve. ***And the data compression of the result file must be switched off by the directive “.options plotwinsize=0”.***

The signal is generated by a **PWL voltage source using the following data:**

0 Volt	**at t = 0**
0 Volt	**at t = 2ms**
1 Volt	**at t = 2.001ms**
1 Volt	**at t = 3,001 ms**
0 Volt	**at t = 3,002 ms**

After the simulation of the voltage at R1 open the menu „Window“ and choose the option **tile horizontally**“. This gives this screen.



Now we want to perform a Fast Fourier Transformation and so some explanation is necessary:
Choosing a simulation time of 10ms is crucial for the FFT simulation result because you get from it the lowest frequency of the spectrum AND the frequency resolution (…including the line width of a signal…) by the following relationship:

frequency resolution = 1 / simulation time = 1 / 10ms = 100Hz

Choosing a “maximum time step” of 100ns during this simulation time of 10ms results in a minimum amount of **10ms / 100ns = 100 000 „samples“** which are then saved.

This is the absolute upper limit for the „number of data point samples in time“ which must be entered in the menu when calling and before starting the FFT. So let's stay on the safe side and use the preset value of 65536 Samples.

If you use more than the mentioned 100 000 samples: this can cause errors and / or an additional noise floor at high frequencies).

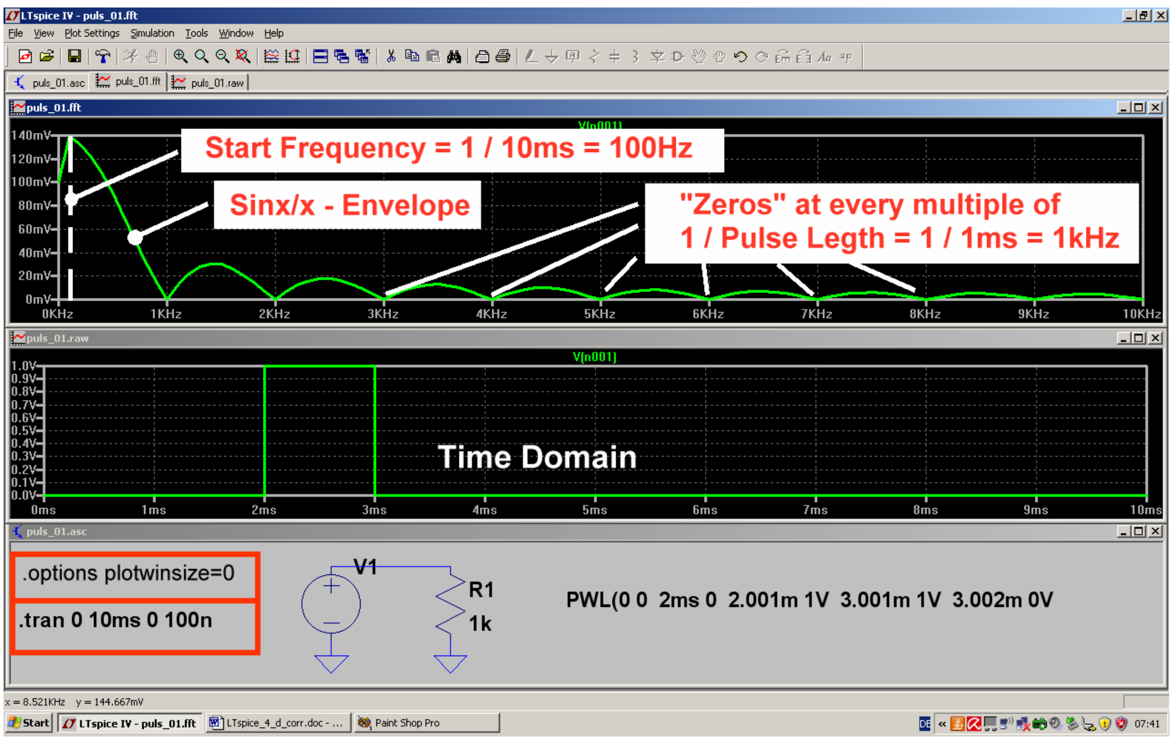
But:
Increasing the number of produced or used number of samples in a FFT gives a higher stop frequency and a higher amplitude dynamic range. In other words: a better resolution (and a longer computation time and a larger data file….)

So right click on the waveform viewer, go to „View“ and „FFT“ and set the sample number to 65536. Start the FFT simulation with OK and afterwards modify the result as follows:

a) Move the cursor to the scaling of the vertical axis and left click: now you can change to „linear presentation“

b) The horizontal axis set to „linear“, using a start frequency of 0Hz, a stop frequency of 10kHz and a tick of 1kHz.

c) Finally choose „tile horizontally“ under „Window“ in the main menu for this result:



The simulation result confirms the theory which says:

a) In the spectrum of a non-periodic signal you cannot find any discrete frequency lines. The complete frequency range is „loaded“ with energy. So what you see in the diagram is the spectral power density.

b) The amplitude of this density follows to a **„Sin x / x“ envelope function**.

c) “Zeros” in the envelope are always found at every multiple of 1 / pulse length
(Shortening the pulse length increases the “frequency of the fist zero” and all following)

d) The diagram starts with a frequency of 100Hz = 1 / 10ms = 1 / simulation time.

The curve is plotted with thick lines. You find this option by selecting the “button with the hammer” and „Waveforms tab“ as **„Plot data with thick lines**“…

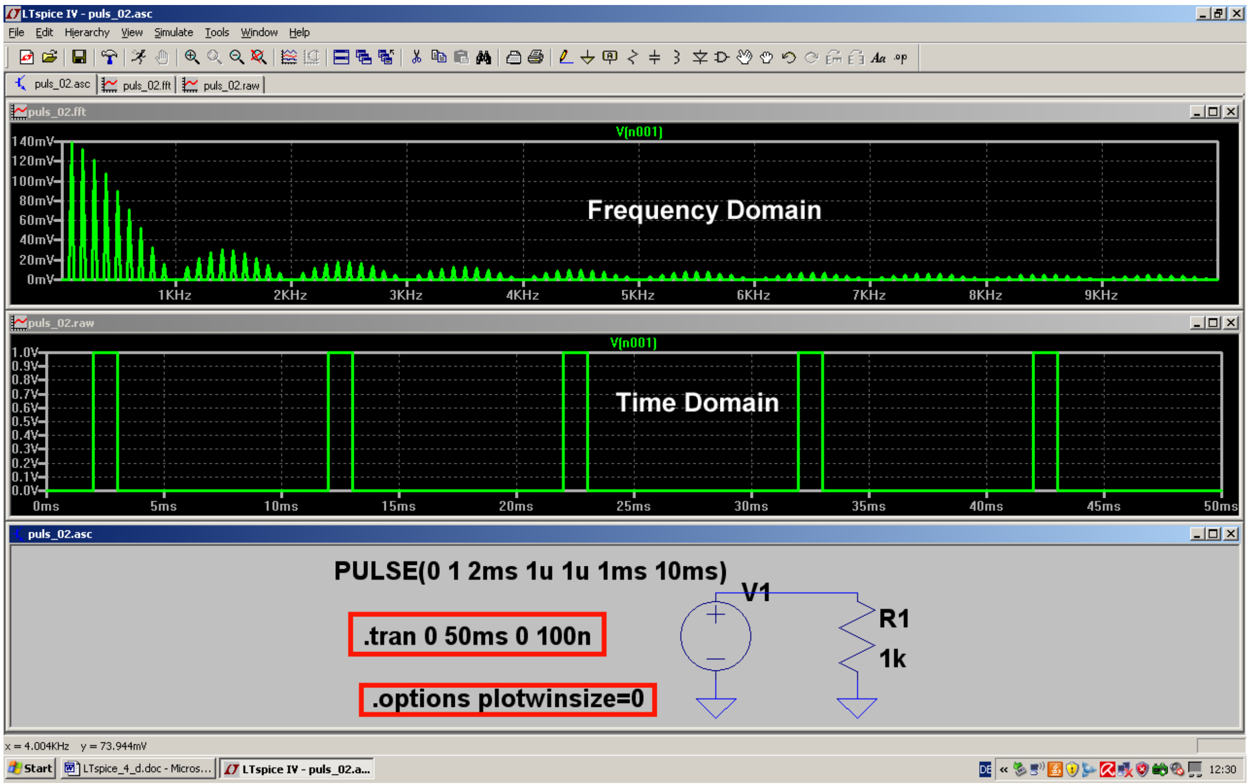
19.3. Simulation of a Periodic Pulse Spectrum

We use the same signal as in the preceding section, but repeat it at 100 times per second. Therefore we need **a pulse voltage source**. In the circuit diagram below you can see how to program the pulse source to achieve this. The simulation time is increased to **50ms** (which gives a frequency resolution of 20Hz) and the maximum time step is set to **100ns**. So we get a minimum of **500 000 actual samples** and can use a number of **262 144 samples for the FFT.**

Now simulate the voltage at R1 and plot the spectrum. Display the FFT result using the following settings:

Vertical axis: **linear**

Horizontal axis: **linear ; start = 0kHz ; tick = 1kHz ; stop = 10kHz**



You can see:

a) It's a **periodic signal,** so we get **a pure „line spectrum**" with the fundamental frequency of 100Hz and the harmonics at every multiple of 100Hz. Due to the simulation time of 50ms the diagram starts at 20Hz. This is also the width of every simulated line.

b) A **pulse length of 1ms gives a Sinx/x envelope function**

c) As before **the "zeros" in the envelope** can be found at

$$\mathbf{F_{NULL} = 1 \ / \ pulse\ length = 1 \ / \ 1ms = 1kHz}$$

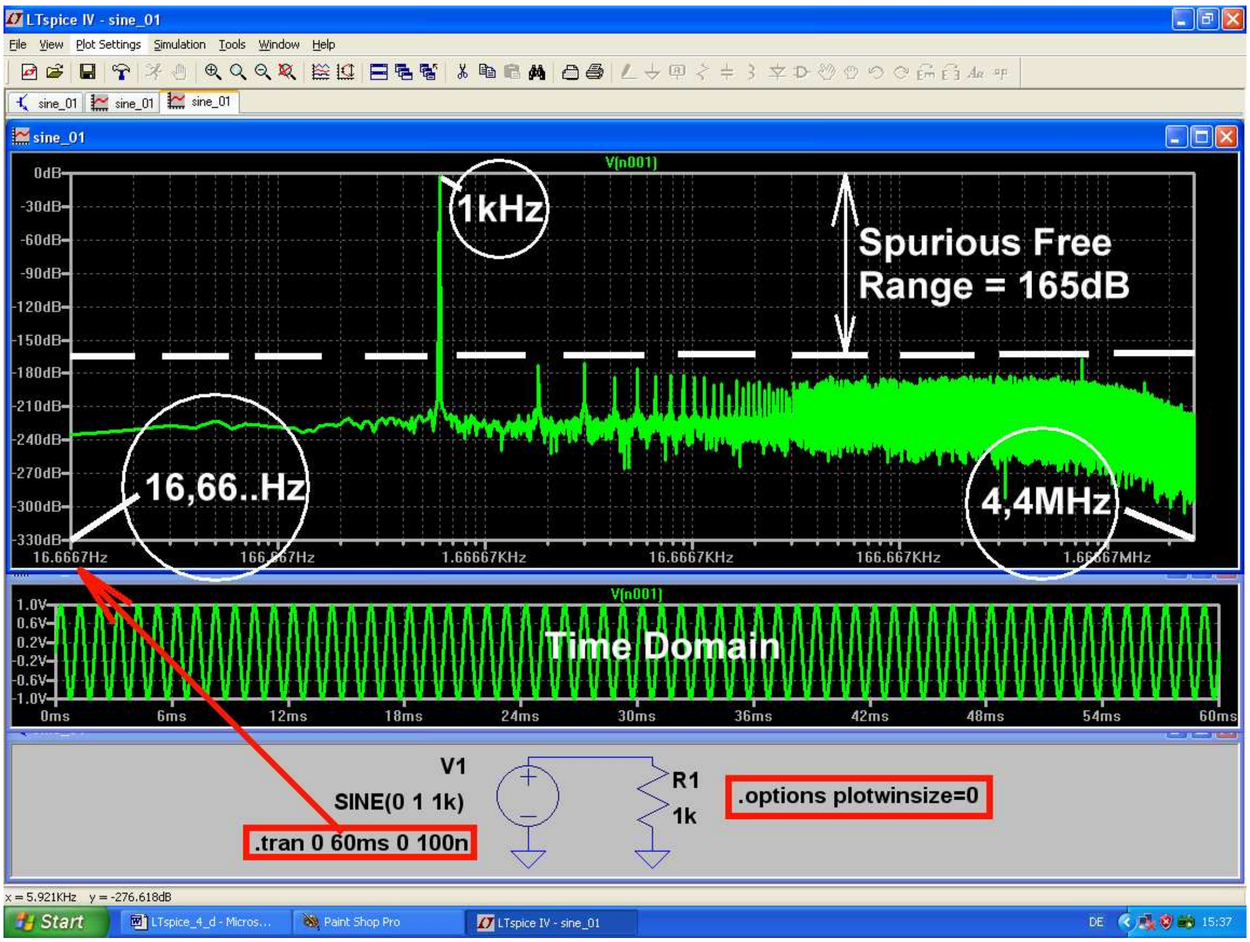
and it's multiples.

19.4. An Ideal Sine Wave

Let us proove the definition which we stated in chapter 19.1 (= fundamentals):

Only a pure continuous sine wave where neither the start point nor the stop point can be observed consists of only one spectral line!

We use a sine wave with a peak value of 1V and a frequency of 1kHz. Simulation time is 60ms using a maximum time step of 100ns. Thus we get 600 000 samples and so it should be possible to find every undesired signal when using 524 288 samples for the FFT. The data compression must also be switched off and the simulation is an integer multiple of the signals's time period (as recommended in chapter 5, FFT)



It is true! From 17Hz up to 4.4MHz every undesired signal is attenuated by more but 160dB and only the 1kHz sine wave line can be seen. Nearly ideal and perfect…

19.5. The „Asymmetric Clipped“ Sine Wave

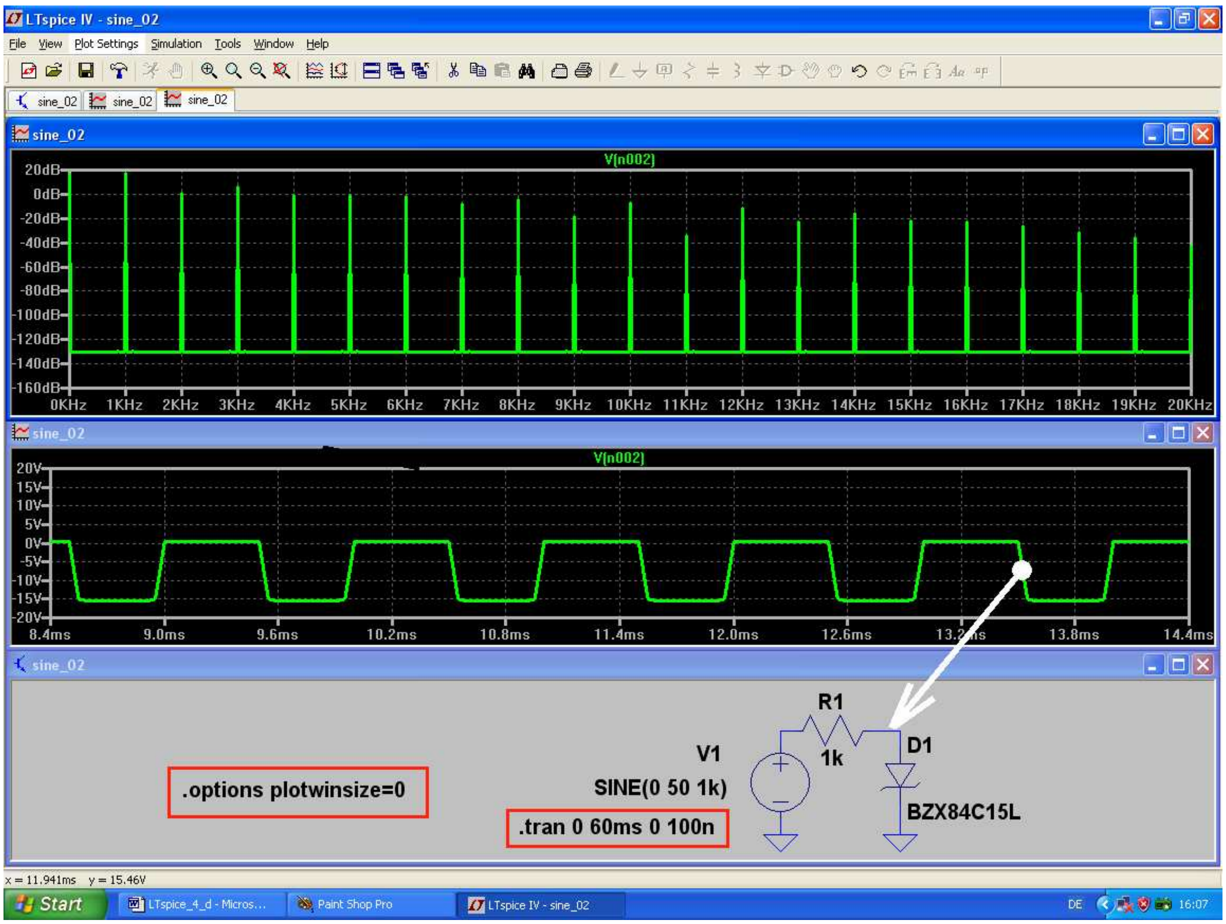
This is done by a zener diode in series with a resistor and showing the simulation of the diode voltage. So the positive half of the sine wave is limited to 0.7V by the conducting silicon diode in the circuit.
When polarity is changing we measure -15V due to the Zener voltage of a BZX84C15L.

But first you have to pick up the diode symbol from the menu and place it. Then right click on the symbol and open the menu „Pick new diode“.

The peak value of the input sine wave must be increased to 50V. Only then the limiting effect can clearly be seen.

Simulation time is 0…60ms with a maximum time Step of 100ns. This gives 60ms/100ns = 600 000 samples for the FFT. Do not forget to switch off the data compression.

On a linear scale show the frequency range from 0 to 20 kHz to see the mixture of even and odd harmonics:



19.6. The Symmetrically Clipped Sine Wave

This is a simple exercise because we connect two BZX84C15L zener diodes in series (opposite polarization).

A sine wave is applied again (peak value = 50V, f = 1 kHz). Simulation time is 60ms and the maximum time step is 100ns. This gives 600 000 samples. Data compression is switched off.

We use the same scaling as in section 19.5 and now you see that only odd harmonics can be found.

20. The Secret of the Impulse Response

In modern system theory, remote control and communications the Laplace- and Fourier- and Fast Fourier Transformations are very important tools.

But before starting with an application you have to know the transfer function of the system, which is the ratio of output voltage to the input voltage for all frequencies in complex form (= magnitude and phase).

f you know this "transfer function" then it is possible to predict the "system's response " for any input waveform using two different methods.

==

Method 1 / frequency domain:
Multiply the spectrum of the input signal with the transfer function spectrum and you get the spectrum of the output signal.
Then the inverse Fourier transform will give you the response of the output signal in the time domain.

==

In a complicated system , sometimes the exact calculation or measurement of the transfer function as „magnitude and phase for different frequencies" is difficult or impossible to determine. In these cases use the following:

If the input to an unknown system is a „Dirac - Impulse", then the output of the system has all the information about the transfer function (in the form of the impulse response)!

As we have already seen the power density spectrum of a Dirac-Impulse is constant over the whole frequency range (up to infinity).

This is equal to an AC sweep with a start frequency at 0 Hz and a stop frequency at infinity!

OK, this is a response in the time domain. But time domain and frequency domain simply are the different sides of the same coin! All we then need is a conversion of this response in the time domain to a response in the frequency domain! So we come to

==

Method 2 / time domain:
The output voltage Ua(t) for a behaviour input voltage comes by application of the convolution integral to the input voltage Ue(t) and the impulse response g(t):

$$Ua(t) = Ue(t) * g(t)$$

===

Remarks:
A **Dirac – impulse** is a needle impulse with infinite amplitude and a pulse length decreasing to zero. **But the pulse area is constant and defined as 1Volt x 1Second.**
In practice this is impossible, but it can be approximated by a pulse with huge amplitude and very short pulse length. This pulse length should 100…..1000 times shorter than the system's time constant. If the area of the „single pulse" is the same as that of the Dirac impulse you'll get the same results.

Also in practice you cannot use a Dirac pulse length of 1 Microsecond and an area of 1Volt x 1Second -- this would give a pulse amplitude of 1 Megavolt!

So in an LTI-System (LTI = linear and time invariant) it is customary to apply much smaller pulse amplitudes which can still be handled by the system (e. g. a transistor amplifier). If necessary the answer to a correct pulse area with 1Volt x 1 Second can be calculated by multiplying the measured output signal with the inverse reduction factor.

20.1. First Example: Dirac Pulse applied to a 160Hz RC Lowpass Filter

Task:
Feed a needle pulse with an area of 1Volt x 1Second to the input of a RC LPF **(R = 100kΩ, C = 10nF)**.

a) Calculate the time constant of the LPF and choose a pulse length which is 1000 times shorter.

b) Calculate the pulse amplitude from the pulse length and the pulse area. Use a PWL-source to generate the pulse signal with a **risetime = falltime = 1000 times shorter than the pulse length, a start delay of 1ms and a simulation time from 0 to 10ms with a maximum time step of 100ns. Switch off data compression.**

c) Draw the schematic with the LTspice editor, simulate and present input and output voltage in different plot panes (use „tile horizontally").

d) Apply an FFT to the input voltage (with **65 536 data samples in time**). Before transforming calculate whether you have enough true (= simulated) samples for a correct FFT.

e) Apply an **FFT** to the output signal.

Use a linear scaling from 0….20kHz with a tick of 1kHz for the frequency axis when presenting the spectrum.

Solution for a): Calculation of the necessary pulse length

We have a corner frequency of

$$f_{CORNER} = \frac{1}{2\pi \bullet RC} = \frac{1}{2\pi \bullet 100k \bullet 10nF} = 159Hz$$

This gives a time constant of:

$$RC = \frac{1}{2\pi \bullet f_{CORNER}} = \tau = 100k\Omega \bullet 10nF = 1ms$$

So we use a pulse length which is 1000 times shorter = **1µs.**

Solution for b) and c): Simulation using the PWL source

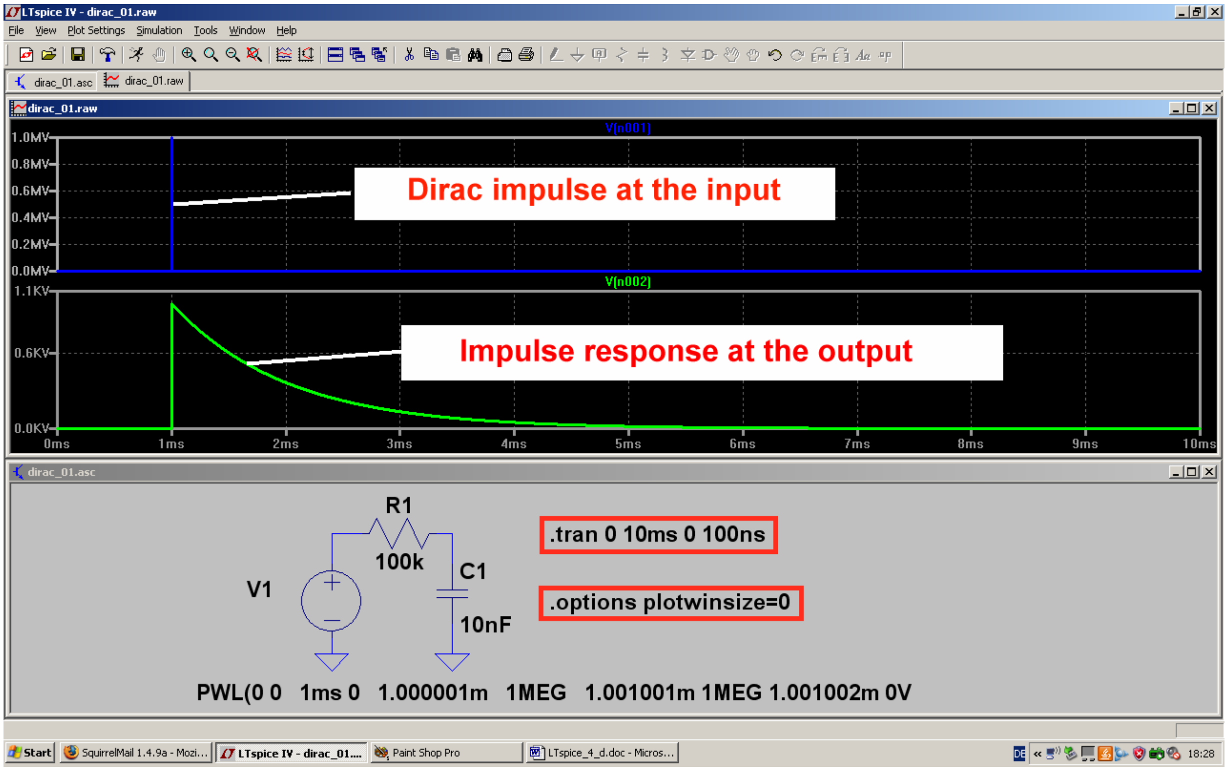
The pulse length is 1µs, **so the rise time and the fall time must each be 1ns (= 1µs / 1000).**

A pulse length of 1µs and a pulse area of 1 Volt x 1 second gives a pulse amplitude of
1 Megavolt = (1V x 1s) / 1µs.

The PWL source is now programmed as follows:

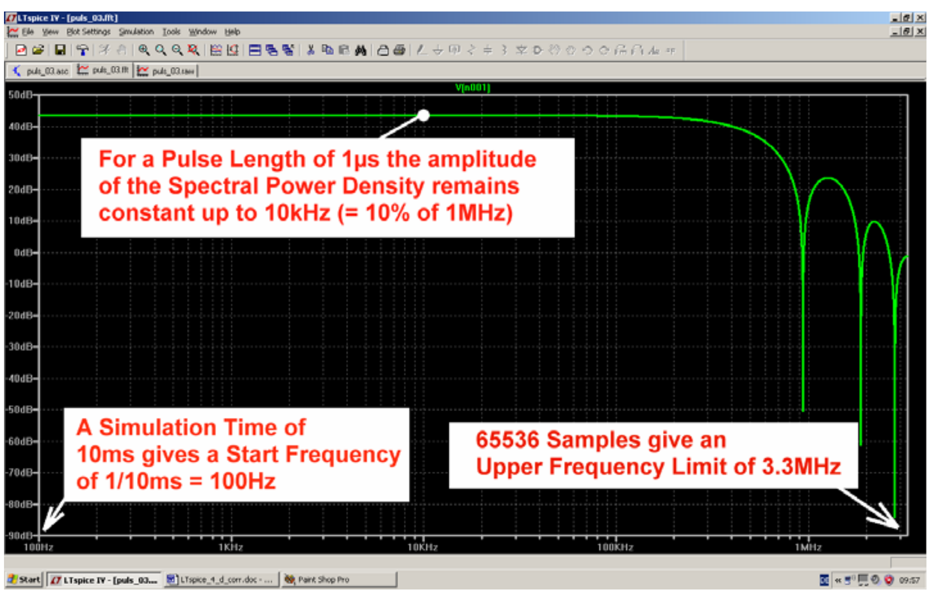
At T = 0:	**Amplitude = 0**
At T = 1ms	**Amplitude = 0**
At T = 1.000001ms	**Amplitude = 1MEG**
At T = 1.001001ms	**Amplitude = 1MEG**
At T = 1.001002ms	**Amplitude = 0**

See the simulation result screen on the next page including the properties of the PWL source.



Solution for d): FFT of the input signal

Simulating for 10ms with a maximum time step of 100ns gives a minimum of 10ms / 100ns = 100 000 true samples. So right click on the trace of the input voltage in the diagram, choose “View” and “FFT” and use **65536 sampled data points.** Close with OK and examine:



a) start frequency
=
frequency resolution

=

minimum frequency step

=

minimum spectral line width

which are determined by the simulation time of 10ms = 1 / 10ms = 100Hz.

b) The upper frequency limit in the diagram (= stop frequency) is determined by the number of samples used for the FFT **AND** by the maximum time step (which gives the minimum sample frequency in the time domain).

But:

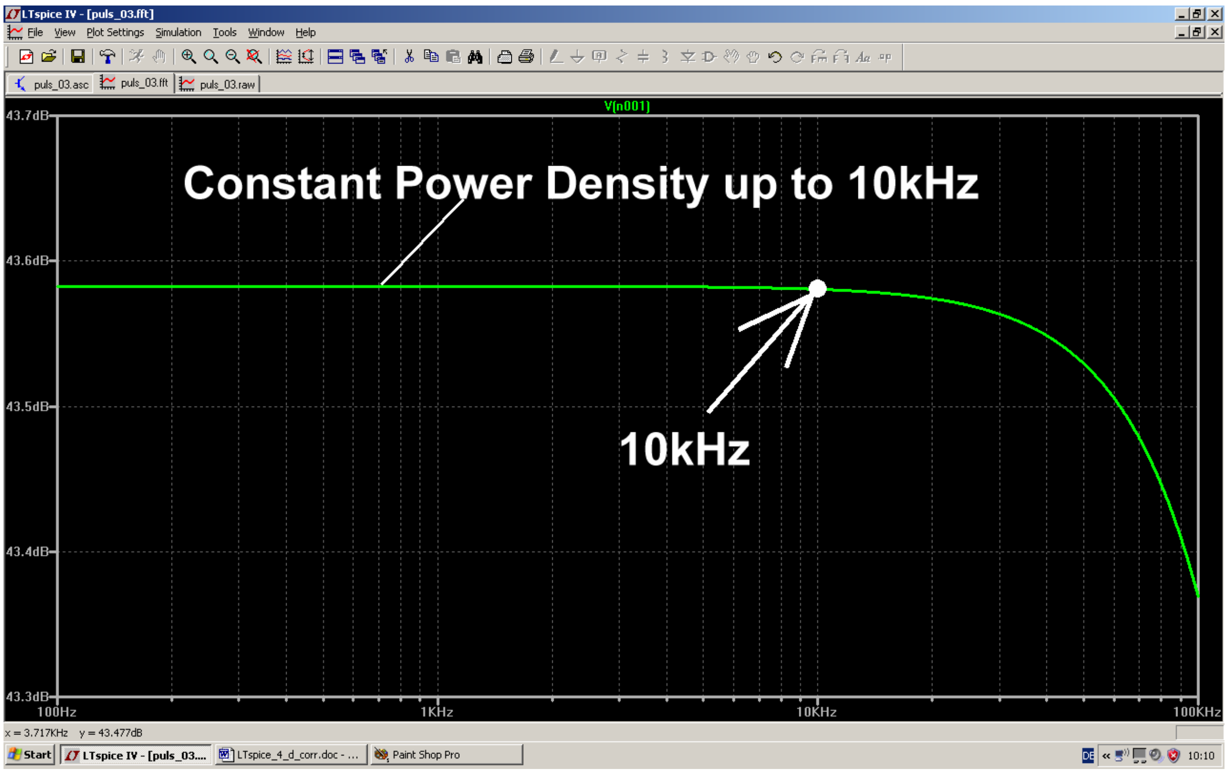
For a correct FFT you always need a "multiple of 2" samples (e. g. 128 or 256 or 512 or 1024 or 2048….). Do not try to change this. Always check by calculation whether you have enough "true sampled values" before calling the FFT. This number is simply given by the ratio "**simulation time / minimum time step".**

c) A pulse length of 1µs gives a first "zero" in the spectrum's envelope at

1 / 1µs = 1MHz

but the Sinx/x function of the envelope only shows a constant amplitude of the spectral power density up to 10% of the „first ZERO frequency". For this example this means 10% of 1MHz = 10kHz.

This is equal to an AC sweep with constant input voltage amplitude up to a stop frequency of 10kHz. Check this by zooming the diagram for a frequency range from 100Hz to 100kHz and an expanded amplitude range:



Additional Tasks:

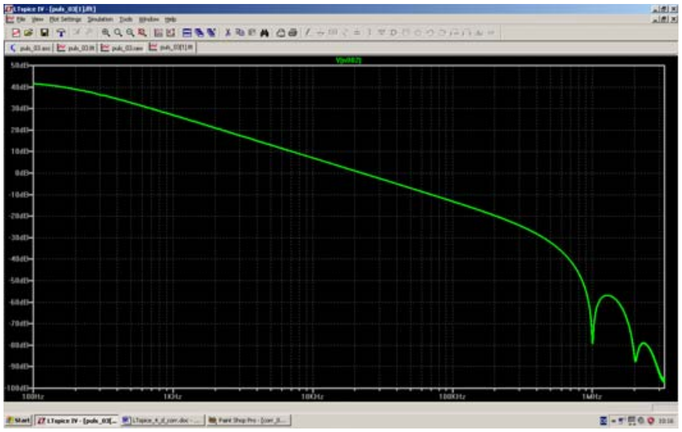
Repeat the simulation and the FFT with different simulation times (e. g. 20 ms / 50ms / 100ms) and compare with a).

Repeat the FFT with different numbers of sampled data points and compare (e. g. 16384 / 32768 / 131072 / 262144 samples).

Solution for e): FFT of the impulse response = output signal
Remember:

Time domain and frequency domain are simply the two different sides of the same coin! Both completely describe the properties of our LPF.

So when applying an FFT to the output signal, we use „convolution" which would then show the transfer function of our LPF as a result.

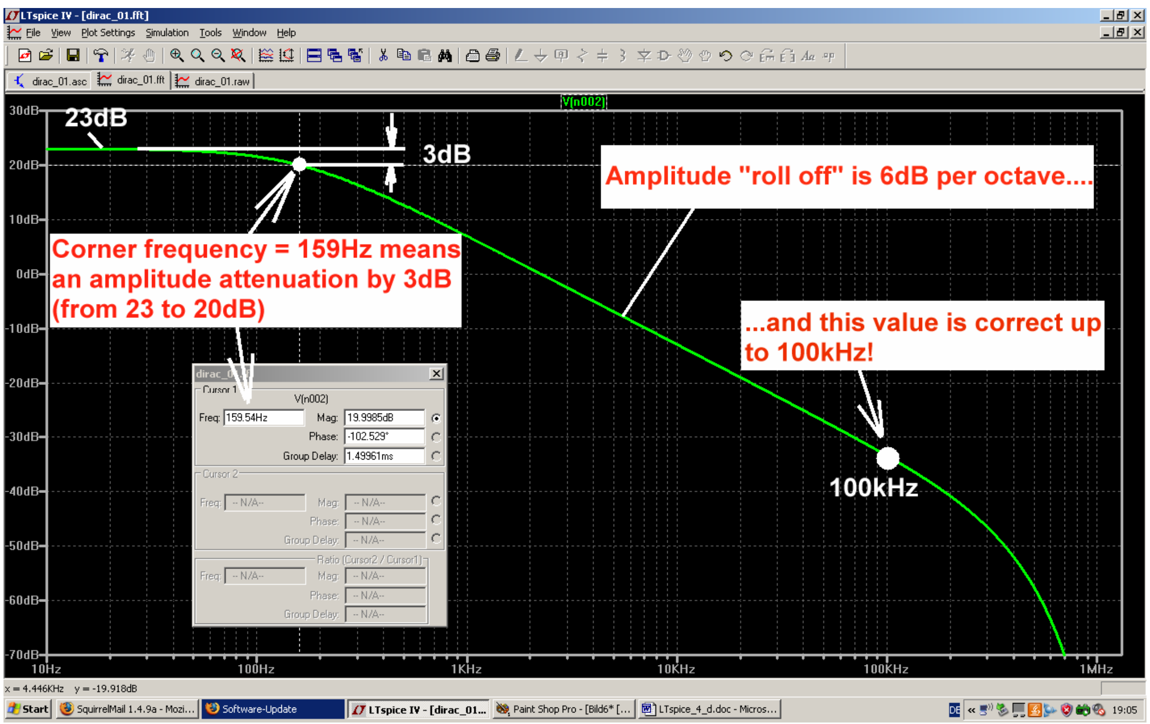


At higher frequencies the influence of the "single Dirac impulse" can be seen. This means that we see the Sinx / x – envelope with zeros belonging to the NON-ZERO pulse length. So do not use the results in this region.

But the results at the low frequencies should also be improved. The corner frequency of the LPF is 159Hz and so a frequency resolution of 10Hz should be applied.

Now repeat the simulation with a stop time of 1 / 10Hz = 100ms. The minimum time step can be reduced to 200ns without loosing to much information or decreasing the „number of samples" to less than 500 000 (check: 100ms / 200ns = 500 000).

When using 262 144 samples you see everything what you want:



Start frequency AND frequency resolution are now exactly 10Hz.

At 159 Hz we find the corner frequency of the passband. Here the amplitude is exactly reduced by 3dB.

In the stop band the amplitude "rolls off" (= decreases) with 6dB per octave.

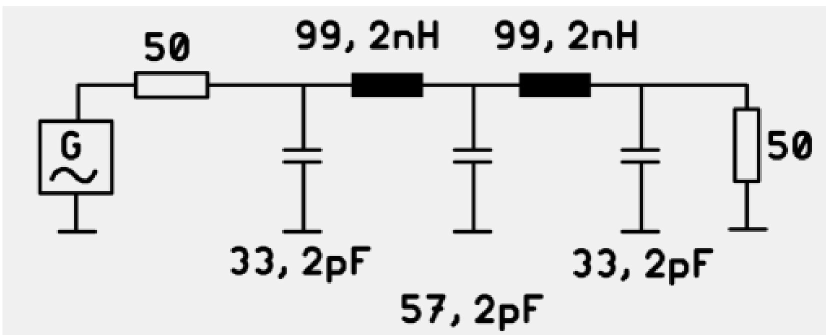
This roll off value of 6dB per octave is constant and correct up to nearly 100kHz.
If you want to improve this: repeat the simulation with a shorter pulse length. This shifts the first "null" in the envelope function to higher frequencies.

20.2. Second Example: Dirac Test of a 110MHz Lowpass Filter (see Chapter 14.2)

20.2.1. Simulating S21 (= Forward Transmission)

These were the specifications of chapter 14.2

„Ripple“ corner Frequency	**fc = 110 MHz**
Minimum inductance version	
Filter order	**n = 5**
System resistance	**Z = 50Ω**
„Passband ripple“	**0,1 dB.**



…and these were the part values calculated by a LC-filter design program.

Now the task:

a) Start an AC sweep of the circuit and **plot S21 from 100kHz to 1000MHz with 401 points per decade**. Zoom the result to demonstrate the “Tchebychev ripple“ of 0.1dB in the passband.

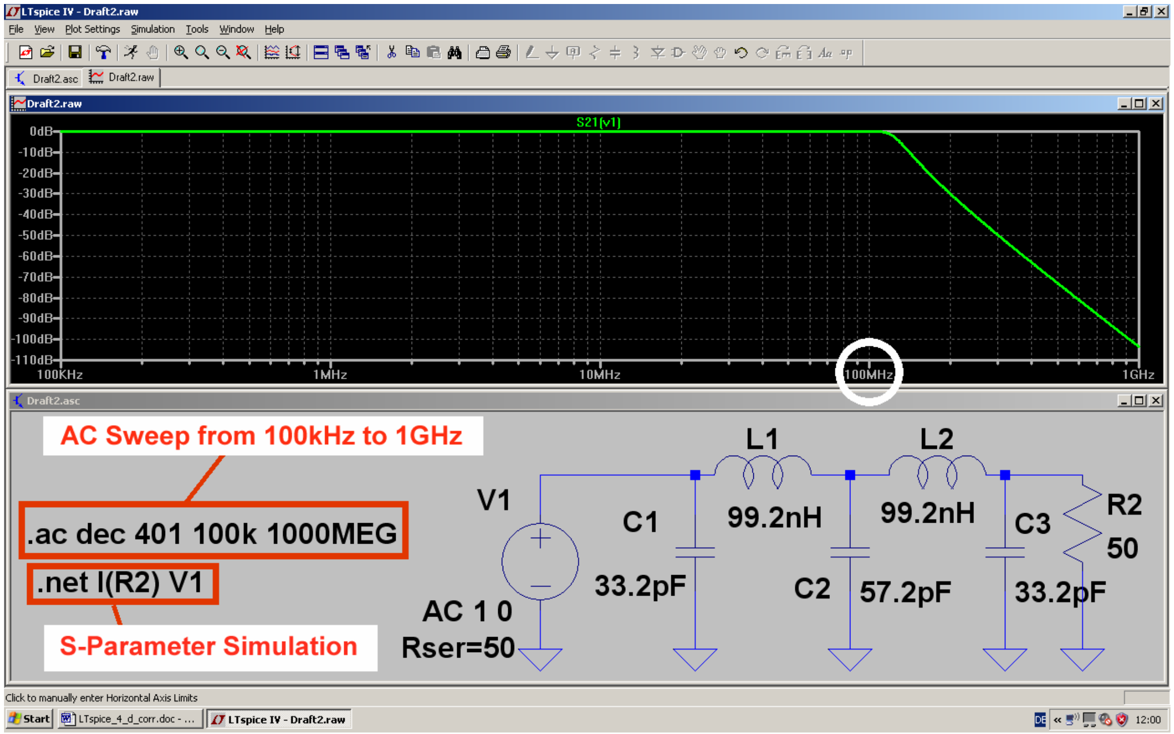
b) Replace the swept input voltage by a „ Dirac impulse“ and use a pulse length which is 100 times shorter than the system’s time constant.
Rise time and fall time should be 100 times shorter than the pulse length.
The pulse amplitude is 1 Megavolt. A start frequency = frequency resolution of 100 kHz is required.
Calculate the necessary sweep time and the maximum time step of the transient simulation for 100 000 true samples (and use 65536 samples when starting the FFT).
The simulation starts after a delay of **1ns**.

c) Choose the Maximum Time Step so short that the programm cannot oversee the Dirac impulse (e.g. 2 or 2 or 3 three samples during the Dirac impule length and simulate in the Time Domain. Switch the data compression off.

d) Calculate the number of usable samples for the FFT and apply the FFT to the impulse response of the filter. Show the Chebychev ripple with it’s amplitude of 0.1dB in the passband and compare the result to the S parameter sweep.

e) Repeat the FFT with only **65 536 samples** and compare the result to d).

Solution for a): Simulated S21 from 100kHz to 1GHz (phase plot switched off):



….and this is the passband ripple of 0.1dB:

This picture is the „specimen" for all the following work.

Solution for b) and c):
The corner frequency of the LPF is 110MHz. This gives a time constant of

$$\tau = \frac{1}{2\pi \bullet f_{CORNER}} = \frac{1}{2\pi \bullet 110MHz} = 1.44ns$$

So the length of the Dirac impulse must be 100 times shorter. We choose ca. **15 picoseconds.**

Rise time and fall times are reduced to (1 / 100 of the pulse length) = **0.15ps = 0.00015ns.**

The amplitude of the pulse is limited to 1Kilovolt. Otherwise we will get problems with the computer's numeric precision capability for this extremely short pulse length. This is customary for an LTI-System like this LPF.

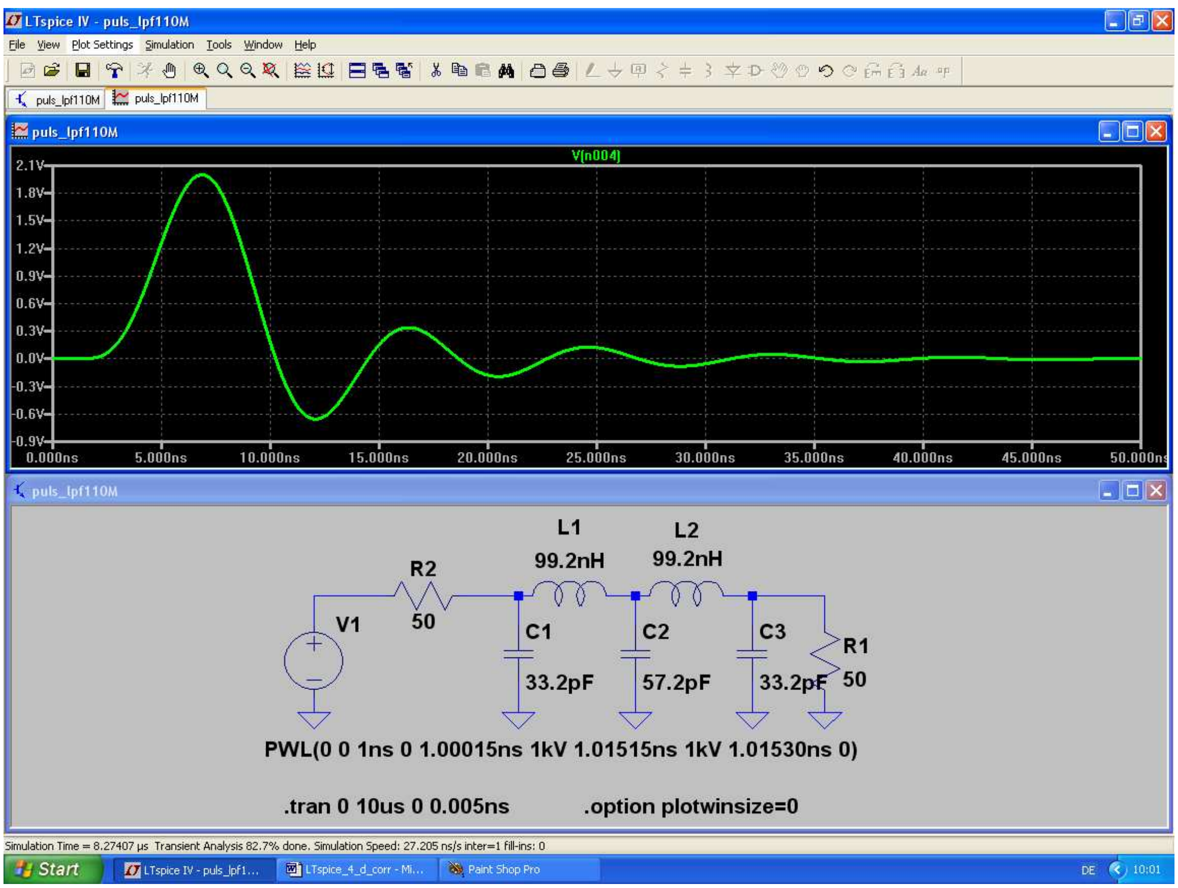
Together with the start delay of 1ns and the amplitude maximum of 1 MegaVolt we get the following properties for the PWL source:

At T = 0:	**U = 0**
At T = 1ns:	**U = 0**
At T = 1,00015ns:	**U = 1 kiloVolt (= 1kV)**
At T = 1,01515ns:	**U = 1 kiloVolt (= 1kV)**
At T = 1,01530ns:	**U = 0**

For a start frequency = frequency resolution = 100kHz in the simulated spectrum a transient simulation time of

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

is necessary. And a Maximum Time Step of **15 picoseconds / 3 = 5 picoseconds** is a good choice that the Dirac impulse will not be overseen by SPICE. Data compression is switched to OFF.



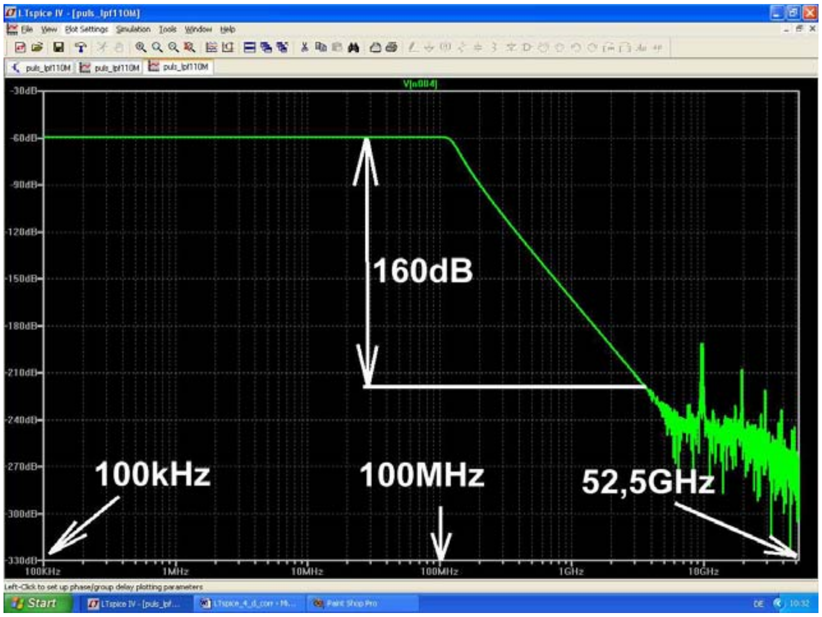
These are the first 50ns of the output voltage = **impulse response.**

Solution for d): FFT

A simulation time of 10µs and a Maximum Time Step of 5 picoseconds give

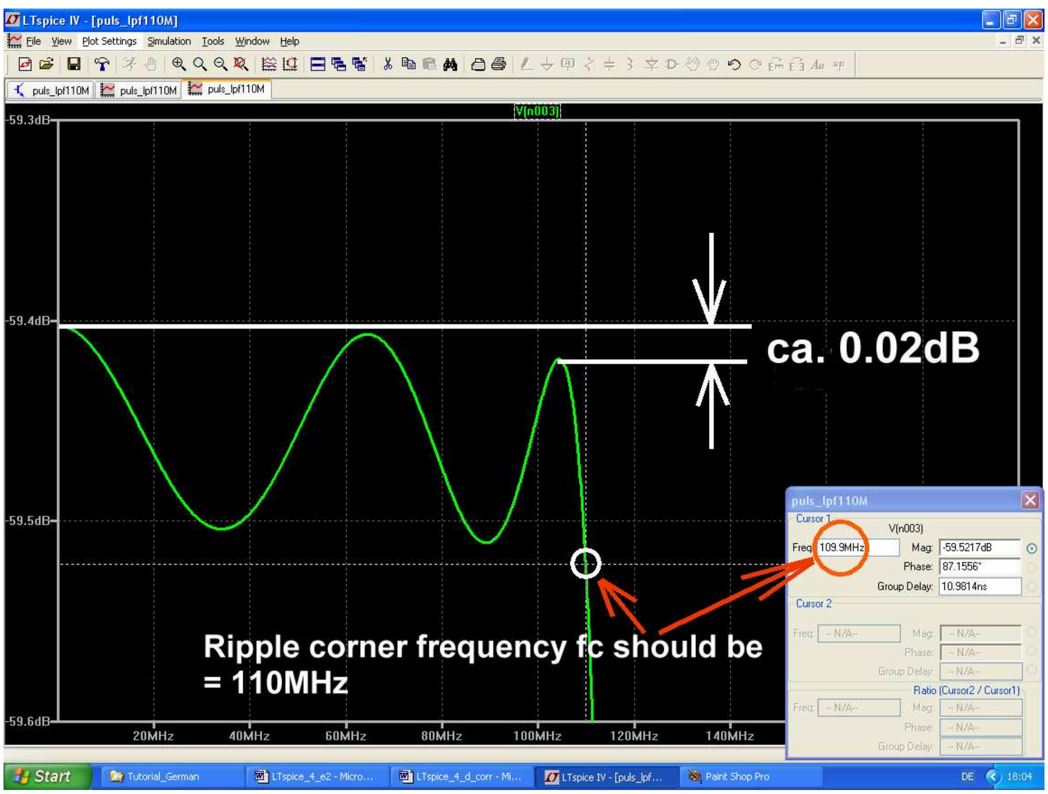
10µs / 0.005ps = 2 000 000 samples

This means that we can use 1 048 576 values for the FFT.



This is the simulation result for a linear frequency axis:

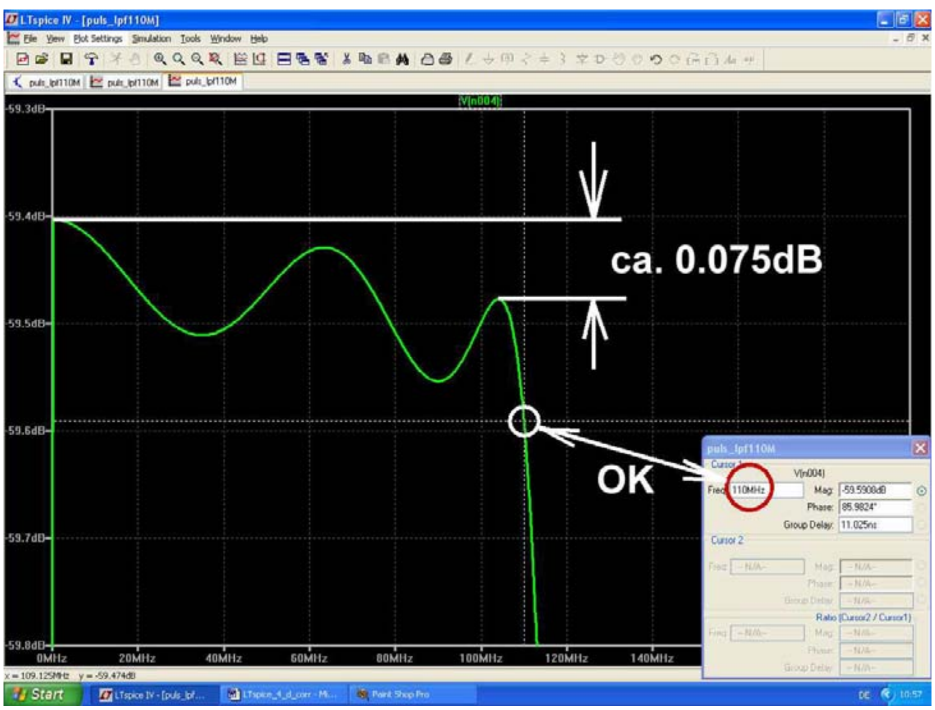
And when zooming we can see the Chebychev ripple and compare to the S parameter sweep:



Not so bad….

Solution for e): simulation with only 65536 samples

Reducing the number of samples for the FFT will also reduce the simulation accuracy AND the the maximum frequency presented in the diagram from 52.5GHz to 3.2GHz



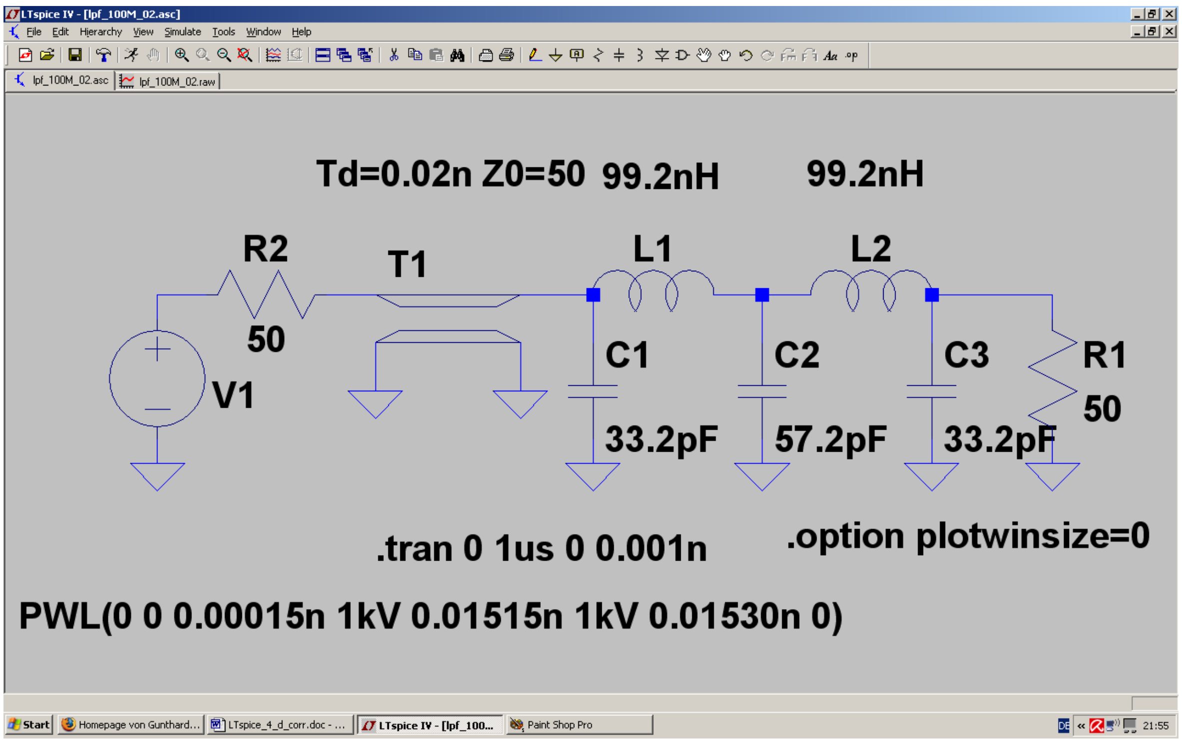
The ripple corner frequency value with 110MHz is correctly simulated, but there is an error of 0.075B at the Chebychev ripple calculation.

20.2.2. Simulation of S11 (= input reflection) or S22 (= output reflection)

The S parameters of a Twoport consist of two transmission coefficients (S21 and S12) and two reflection coefficients (S11 and S22). In practice for the determination of the reflection coefficients directional couplers are used to separate the incident and the reflected wave.

But in a simulation this is very easy when using a Dirac impulse: we insert a short piece of RF cable between the voltage source and the port. The travelling time for the Dirac impulse on this piece of cable should be greater than the pulse length. Now an echo caused by mismatch on the port's input will arrive after 2 x (travelling time on the cable) at the cable's entry. This echo can be used for the FFT and to determine S11 or S22 in dB.

This is the used schematic to determine S11:



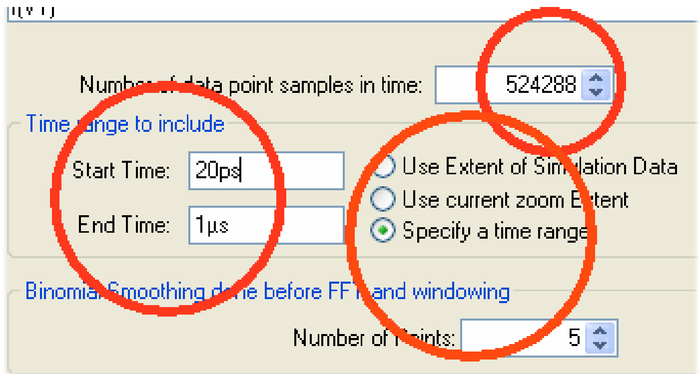
Voltage source V1 is connected to the input of a 50Ω cable with a signal travelling time of **20 picoseconds (= part "tline" in LTspice)**. The output of this cable feeds the input of the LPF with the Dirac impulse (**amplitude = 1 kiloVolt, pulse length = 15 picoseconds, rise and fall time = 0.15 picoseconds, start at t = 0)**. The following value pairs are used to generate this impulse:

At t = Null:	**U = Null**
At t = 0,00015ns:	**U = 1 kiloVolt**
At t = 0,01515ns:	**U = 1 kiloVolt**
At t = 0,01530ns:	**U = Null**

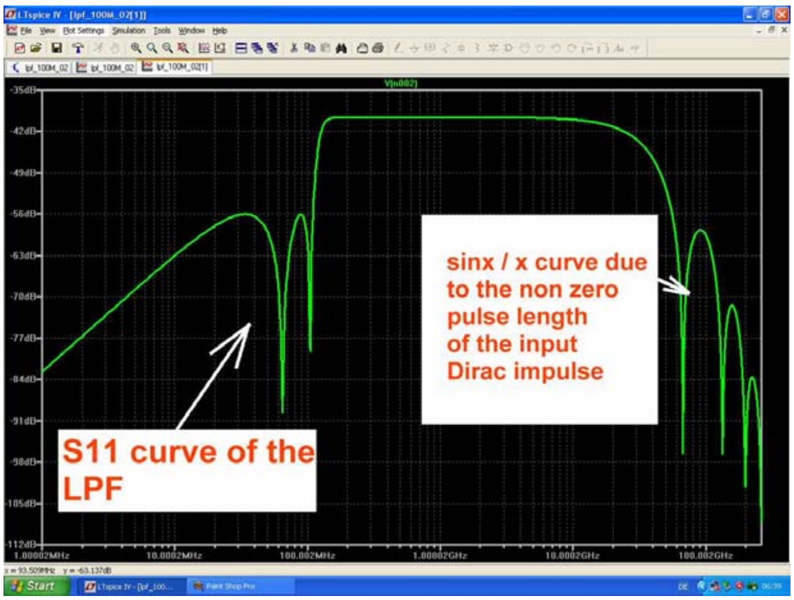
So the echo can be seen and measured at t = 40 picoseconds at cable's entry. **But for the calculation of the FFT this time slot from 0….40 picoseconds must be suppressed.**

Data compression is switched OFF by **.option plotwinsize=0**.

To reduce the PC calculation time only a **simulation time of 1 µs** is used. This gives a frequency resolution of **1 / 1µs = 1MHz** which will do for a S11 calculation. But the Maximum Time Step was reduced to 1 picosecond. So for a pulse length of 15 picoseconds for the input Dirac pulse (…and an echo pulse length in the same order) enough details of the curve are sampled.
This gives **1µs / 0.001ns = 1 000 000** calculated samples of which 524288 are used for the FFT.

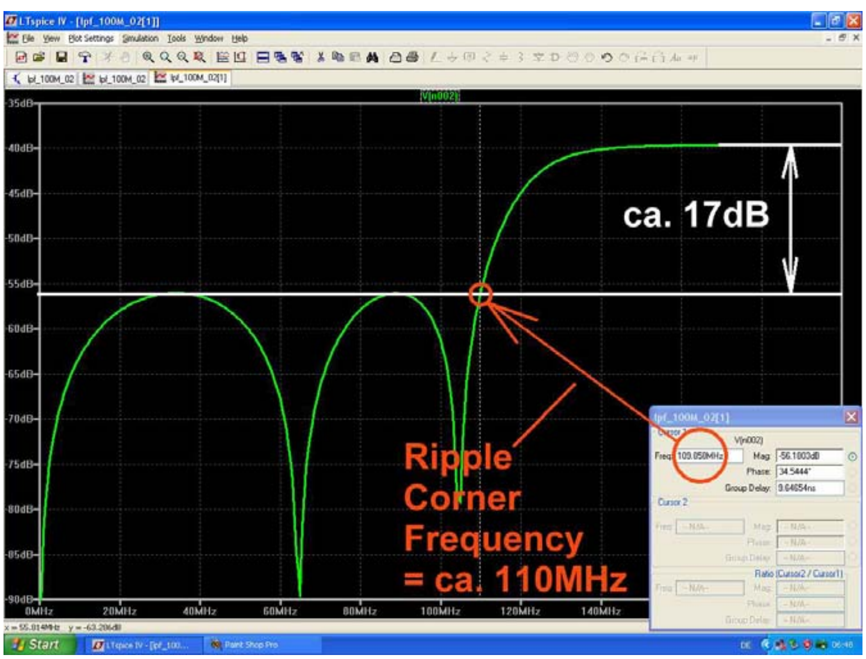


Here come the necessary entries for the FFT. **Do not forget to ignore the time range from 0….20 ps** with the input pulse and use 524288 samples.



This is the result: the S11 curve of the LPF appears at the left hand side of the diagram. At the right hand side the sinx / x curve for the "spare Dirac pulse" due to its non-zero pulse length is present at high frequencies.

But: the maximum frequency value at the horizontal axis is 250 GHz enabled by the short Maximum Time Step of 1 picosecond.

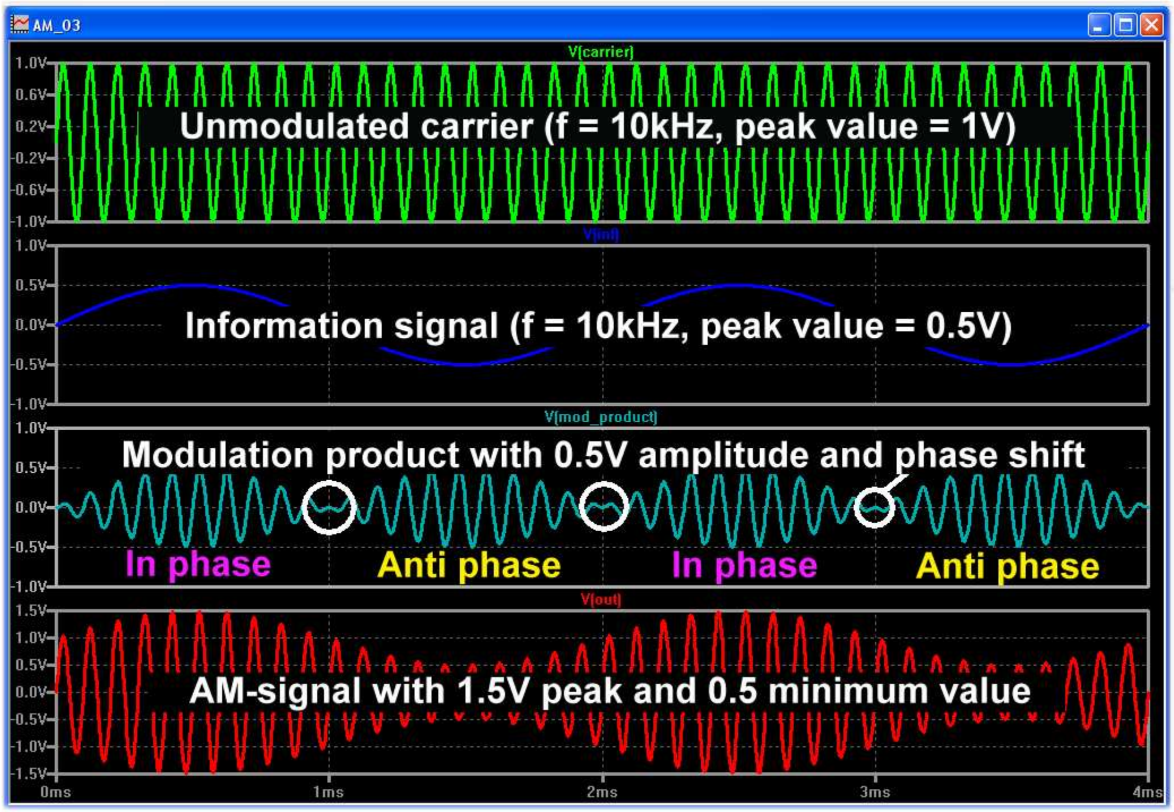


When zooming the result everything is OK and equates to the S parameter sweep in chapter 14.2.

21. Modulation

21.1. Principle of Amplitude Modulation (AM)

It is probably the oldest type of modulation that enabled the transmission of language and music approximately 100 years ago following on from Wireless Telegraphy using a Morse key. The basic idea is simple: Audio signals have wavelengths that are far too large to be radiated wirelessly using antennas with acceptable efficiency. Therefore a high frequency carrier signal was used so that a smaller antenna could be used and the information loaded onto the carrier for problem free transmission. The amplitude of the carrier is changed in sympathy with the information. Mathematically this is a multiplication and addition of two different signals. The secret of this is shown in the following figure:



The top trace shows the constant 10kHz carrier with a peak value of 1V. Below that is the information that is a 500Hz sine wave with a peak value of 0.5V. If these two signals are multiplied the result is a signal at the carrier frequency where:

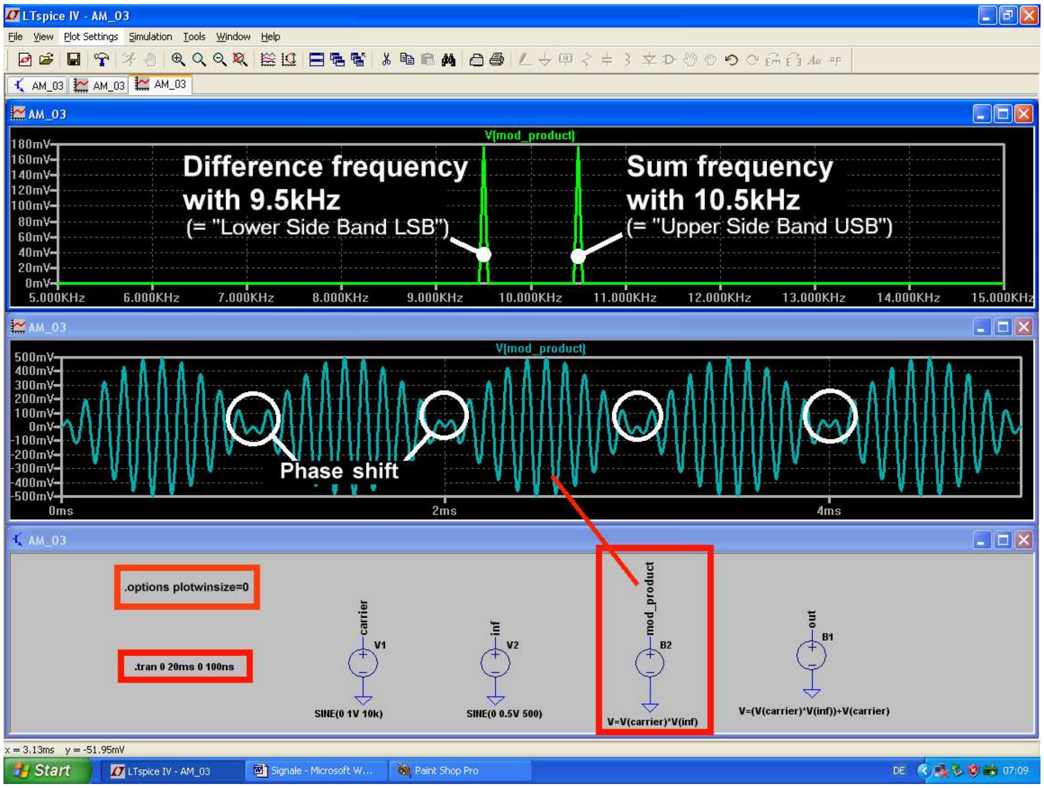
- The amplitude follows the information signal but:

- When the information signal voltage goes negative the phase reverses.

This multiplication is called the Modulation Product shown in the third trace. If this is added to the unmodulated carrier signal the total voltage becomes larger if the modulation product is in phase with the carrier. Likewise it becomes smaller if the modulation product and carrier are out of phase. The bottom trace is the AM signal where the signal amplitude varies in sympathy with the information. Now the question is: how many different frequencies are involved because this is no longer a constant sine wave? With the carrier signal it is simple because it is a constant sine wave with only one spectrum line. The modulation product already has a mathematical formula for the multiplication of two sine signals:

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

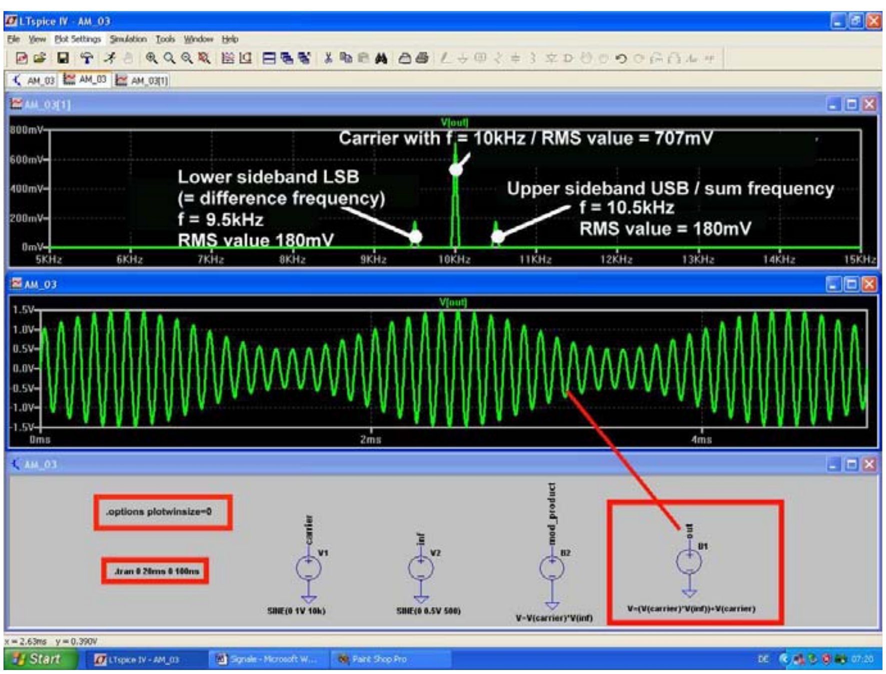
The products are the sum and difference frequencies of the two cosine signals. The carrier and information frequencies are not present.



To simulate this we use the Behavioural Voltage (bv), this has the necessary tools to simulate the multiplication of the two signals. This is confirmed in the above figure:

- The sum frequency is 10kHz + 0.5kHz = 10.5kHz
- The difference frequency is 10kHz - 0.5kHz = 9.5kHz

They are called upper and lower sidebands having equal amplitudes. The peak value of both signals amounts to 50% of the information signal thus 250mVs. In the spectrum display this is 180mV because of the RMS representation.

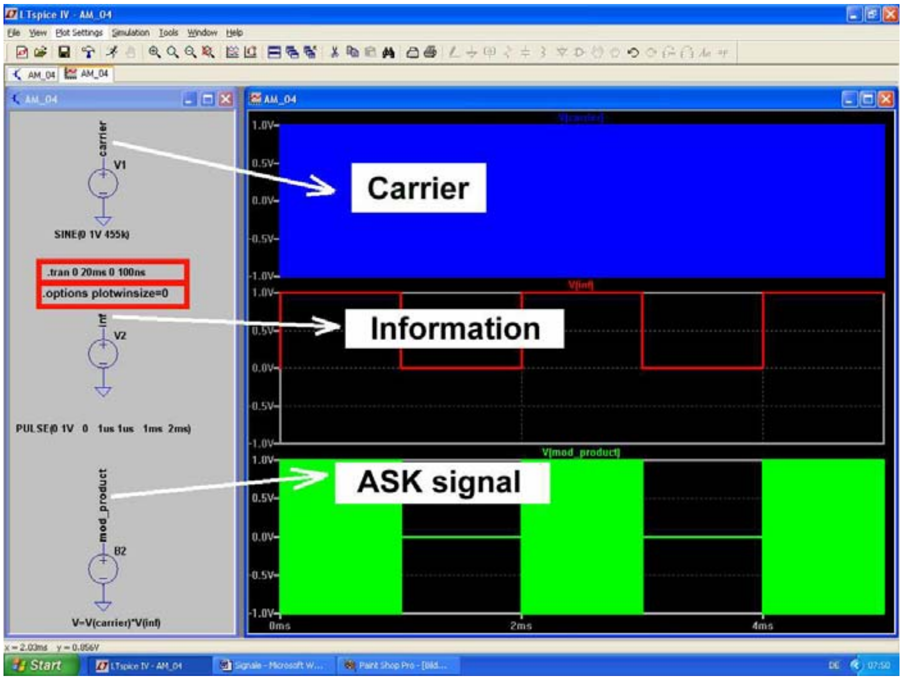


For the complete AM signal the carrier signal is added to the modulation product. This can be simulated using the bv source as shown in this figure.

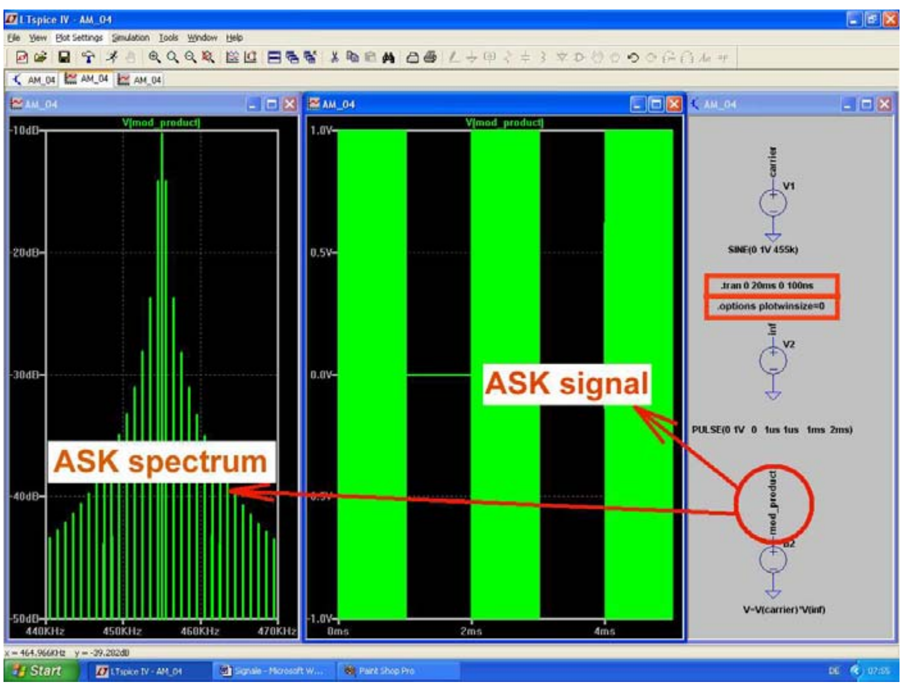
Note:
Usually several frequencies or a complete frequency band are involved in the information signal. These are called the lower and upper sidebands, the Lower Side Band is the LSB and the Upper Side Band is the USB of the AM signal.

21.2 Amplitude Shift Keying (ASK)

The good old Morse key gave just “Open” and “Closed”. This simple method still has its place today in as a modern transmission technique, the logical levels are simply assigned to the two conditions “Zero” and “One“. This works amazingly well and is reasonably interference-proof (if additional limiting is used). In addition the circuits required are fairly simple. The bv source is used again for the simulation multiplying the carrier with a 1kHz symmetrical square wave information signal. The smallest voltage level is zero V, resulting from “Key open”. The maximum voltage is 1V resulting from “Key closed”.



A carrier signal of 455kHz was selected for this example. This figure shows the simulation circuit with the output signals of the three voltage sources used with a peak value of 1V or 0.707V RMS.



The spectrum of the ASK signal is shown in Fig 22, it shows what was to be expected.
The symmetrical square wave information signal contains the 1KHz fundamental and all odd number harmonics. The result is LSBs and USBs with all these spectral lines of the information signal that spread to the left and right of the carrier.

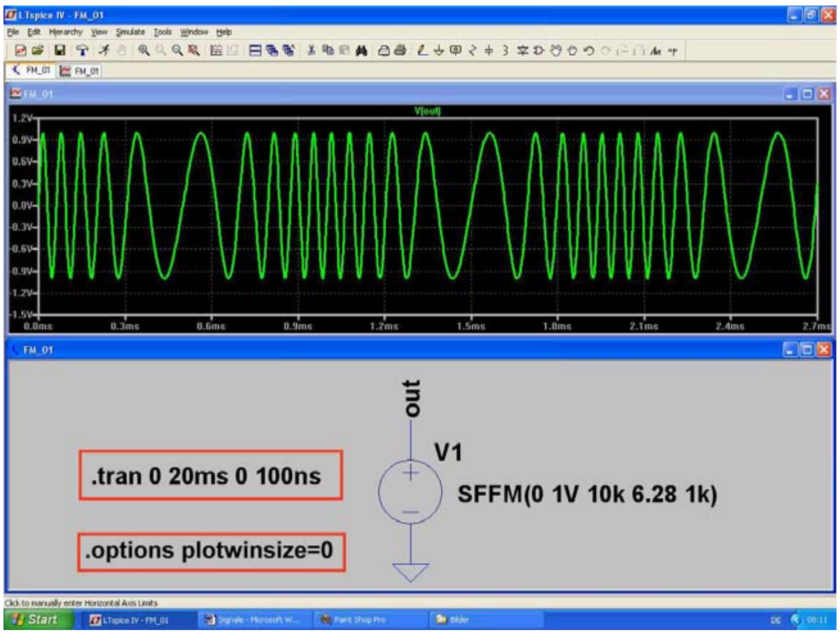
21.3 Frequency modulation (FM)

This is a refinement because this time the amplitude of the carrier is not used. Yes, it is completely unimportant and in an FM receiver a limiter removes all fluctuations and glitches. The carrier frequency is changed in the sympathy with the information signal. That costs more and clearly requires substantially higher technical expertise. Success is convincing, it can be heard by listening to good music received on a VHF receiver. It is good that LTspice has three components available:

• One can be used to switch the voltage supply to produce FM (SFFM). A sinusoidal carrier is produced and sinusoidal information signal is automatically superimposed.

• There is complete FM-AM generator with a “modulate” function that can be found in the “Special Functions” section. It is suitable for information signals with different waveforms. • Finally the same FM-AM generator has a “modulate2” function that produces quadrature outputs (two separate outputs with a 90 degree phase shift between the carrier signals).

21.3.1. Generating an FM signal using the SFFM voltage supply



Take a close look at this figure. The frequency of the signal increases during the negative half wave and decreases during the positive half wave of the information signal. Unfortunately the modulating information signal cannot be shown on a separate diagram in this mode. Data compression is switch off as usual. It is simulated over 20ms (100ns resolution) but for clarity only a short section is shown. The rest can be seen from the SFFM source programming. The line:

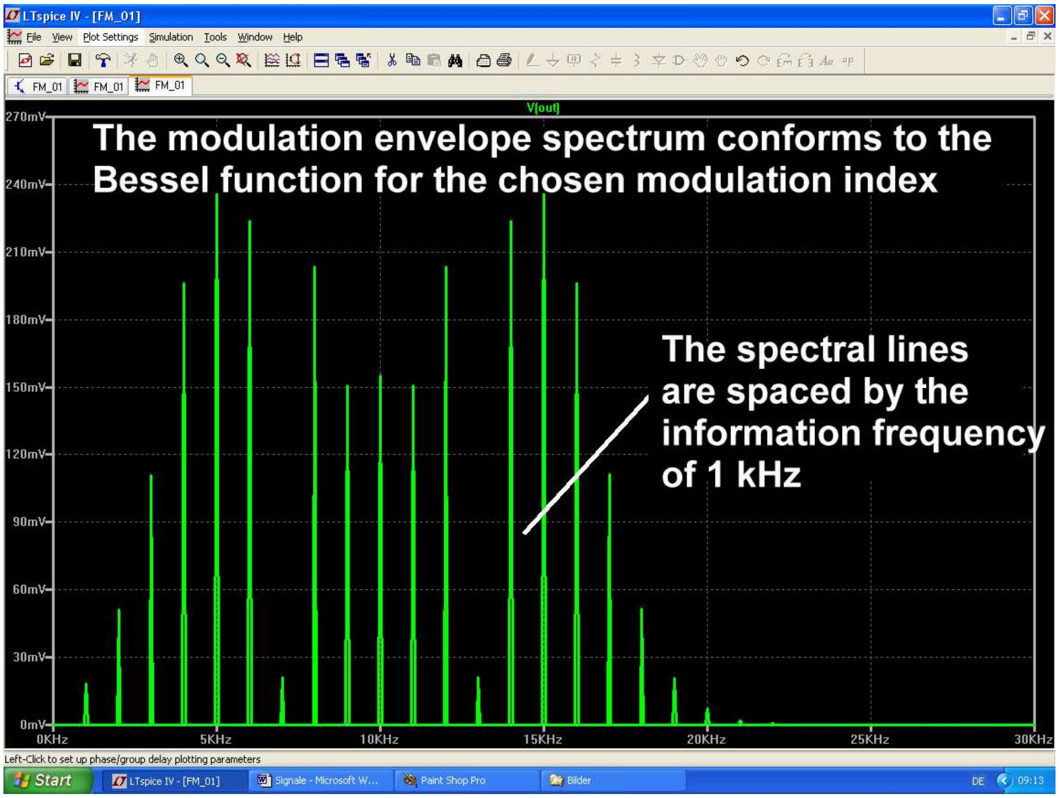
SFFM (0 1V 10k 6.28 1K)

means:

• 0 = no DC portion
• 1V = peak value of the carrier
• 10k = 10kHz unmodulated carrier frequency
• 6.28 = modulation index
• 1k = 1kHz sinusoidal information signal

The modulation index needs an explanation. If the carrier frequency increases it can be interpreted as advancing the phase in relation to the unmodulated signal. Similarly a lower frequency means a reduction of the phase in relation to the unmodulated signal. The modulation index expresses this change of the phase. It represents the maximum phase shift in relation to the steady state with the phase deviation measured in radians. For the example the modulation index of 6.28 represents the phase shift of a circle with a radius of 1. A complete circle is 360 degrees or 6.28 radians. It is so simple!

Now let us regard the simulated FFT spectrum of this signal using 131,072 samples.



The entire spectrum is occupied with lines spaced at a distance of the information signal frequency. The amplitude distribution envelope can be described using a Bessel functions. It is difficult to understand the detail of this but simply the frequency range occupied by the FM signal increases with the modulation index but the envelope and the number of spectral lines obey the Bessel functions. In an extreme case increasing the modulation index, the unmodulated carrier would reach zero.

The **frequency deviation** indicates the maximum deviation of the instantaneous frequency in relation to the unmodulated signal and this naturally related to the modulation index (this is sometimes known a the phase change). As an example the VHF broadcast stations from 88 to 108MHz use a frequency deviation of ±75kHz. That seems small but it cannot be more otherwise the maximum permissible channel spacing (300kHz) would be exceeded and the transmitter on the adjacent channel would be disturbed.

As a guide:

FM channel spacing = 2 x (frequency deviation + information signal frequency)

For interest the phase change can be calculated from the frequency deviation and the information signal frequency and the modulation index in radians can be calculated using:

Modulation index = (peak frequency deviation) / (information frequency)

The differences between frequency modulation and phase modulation are:

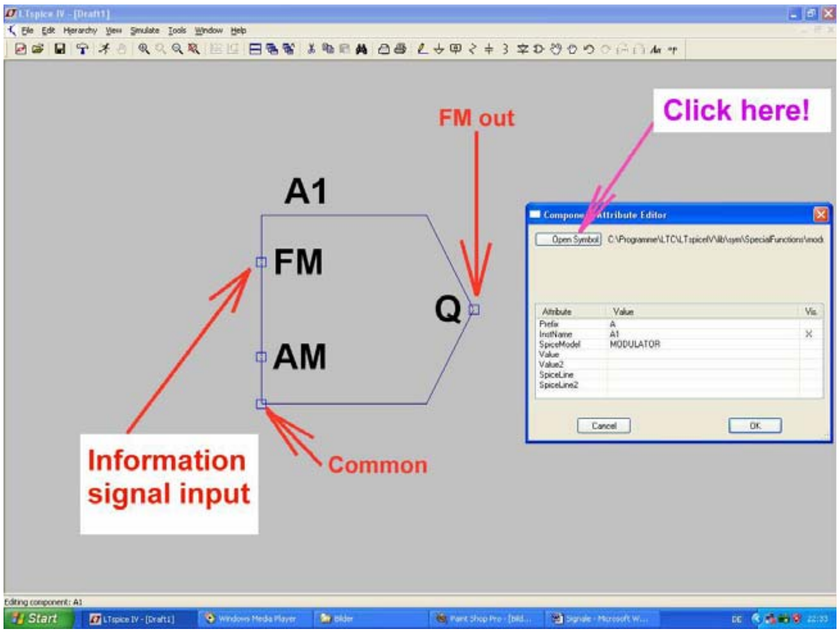
• For **frequency modulation** the **peak frequency deviation is kept constant**. From the formula the modulation index and thus the maximum phase fluctuation must decrease with rising information frequency. Unfortunately this worsens the signal signal-to-noise ratio for high tones.

• For **phase modulation** the **modulation index and phase fluctuation are kept constant**. According to the formula the frequency deviation increases with increasing information signal frequency. This increases the frequency range occupied.

21.3.2. Frequency Shift Keying (FSK)

Frequency Shift Keying is a digital mode that assigns the two conditions, "logic 0" and "logic 1" to two different frequencies. The origin of this mode comes from the days of teleprinter technology. The only things that remain from that technology are the two names, **"Mark"** for the higher frequency and "**Space"** for the lower frequency. These are required to programme the **FM-AM generator "modulate" function of the LTspice simulator**.

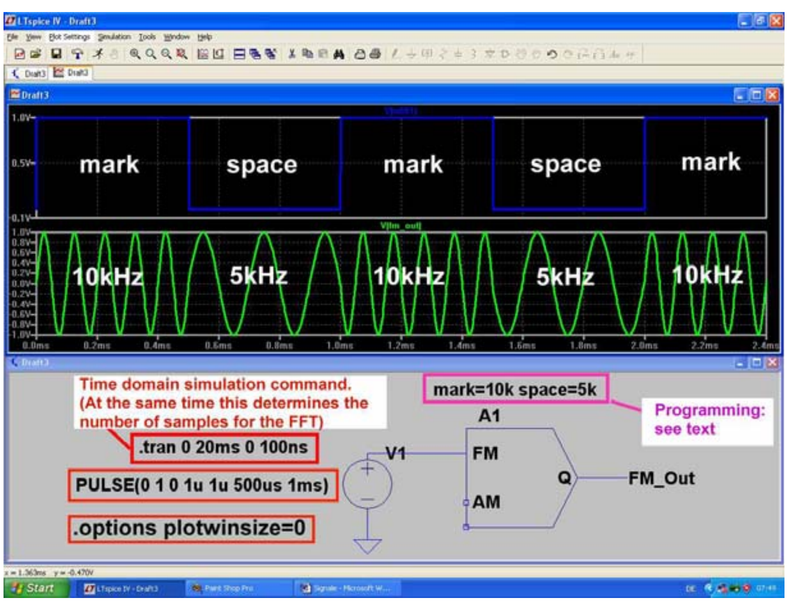
Unfortunately some work is needed. When the VCO (Voltage Controlled Oscillator) is selected the basic adjustment is missing. In order to correct this, click the right mouse button on the circuit symbol and open the "Component Attributes Editor". Click on the "Open Symbol" button to change to the actual symbol editor (recognised by the small red circles at the ends of the individual lines that emerge in the circuit symbol and represent connection points).



The path "**Edit/attributes/Edit At**tributes" leads to the "**Symbol Attribute Editor**" as shown in the figure. In the line "Value" enter "**mark=10k**" and "**space=5k**" as the control characteristic of the VCO to give:

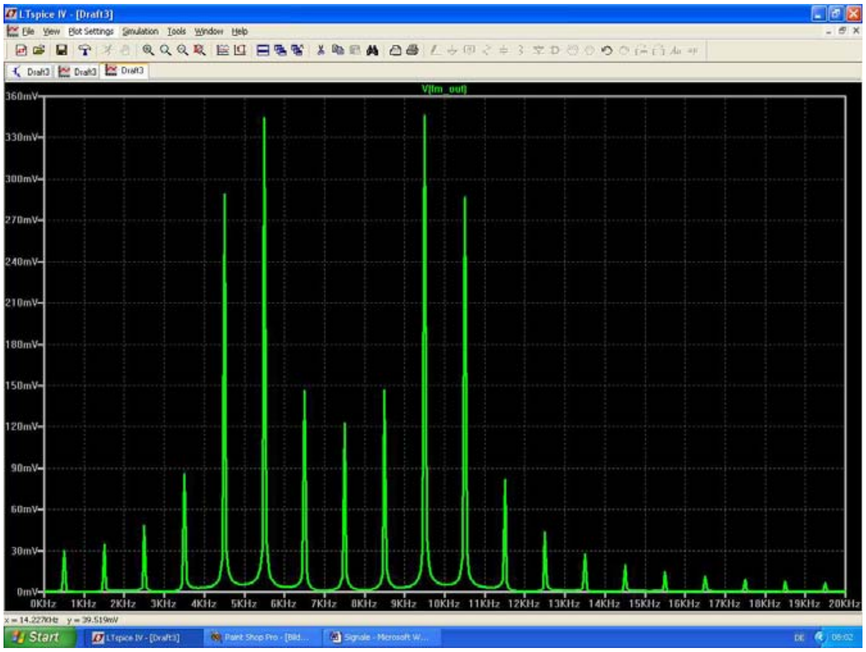
- **For 0V at the FM input the output frequency will be the space value of 5kHz.**
- **For 1V at the FM input the output frequency will be the mark value of 10kHz.**

For a beginner this entry is somewhat complicated: First click on the line "Value" and then type "mark=10k space=5k" in the window above the table. Click OK to make the entry - however it is not shown on the screen. Now select the path: "**Edit/Attributes/Attributes Window**" again and click on "Value" in the table then click OK and the cursor changes to "Mark Space Programming" so that it can be can placed on the circuit symbol. Now this changed symbol must be stored in the component library with a path like:
"LTspiceIV / lib / sym/ Special Functions / modulate".



Now this VCO will be available but only if all windows are closed and the program restarted. Otherwise the program does not know anything about the changed symbol.
The figure shows the symbol being used with a 1kHz pulse with **Vmin = 0V and Vmax = 1V connected to the FM input** and the simulated output (FM_Out).

To ensure that the additionally voltage information is shown, right mouse click on the diagram and use the "Add Plot Pane/Add trace". For the following FFT simulation enter the attributes "20ms simulation time" and "maximum time step 100ns" in order to receive 200,000 genuine samples. Also no data compression should be used.



The linear frequency spectrum simulated with 131,072 samples is shown here.

It exhibits two maxima, one for “Mark” and one for “Space”. This is a wellknown fact that can be found in technical literature.

It occupies a considerable frequency range, probably 20kHz.

For interested readers who have access to a computer there are some additional exercises:

• Change the minimum and maximum pulse amplitude at the input. Examine if negative voltage levels are permissible.

• Use a 1kHz sine wave information signal in place of the pulse. Select the amplitude similar to “Mark” and “Space”. Also use an offset for the sine wave. Compare the frequency range now occupied to the pulse simulation example.

• Use triangle information signal with the same frequency of 1kHz.