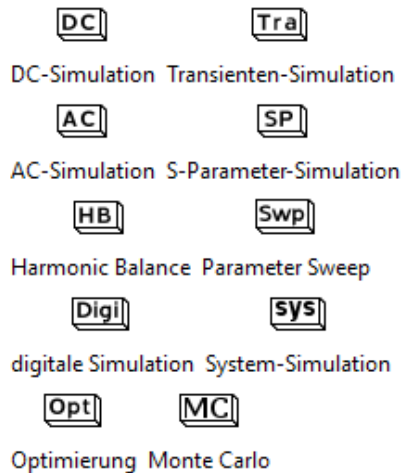


Schaltungs-Simulation mit qucsStudio

von Gunthard Kraus, DG8GB

1. qucsstudio = Quite Universal Circuit Simulator

Dieses Programm kann in der **Time Domain** und in der **Frequency Domain** simulieren, **arbeitet wahlweise mit deutscher oder englischer Bediener-Oberfläche und ist kostenlos.**



Dass der Mikrowellen-Spezialist sein geliebtes Smith-Diagramm verwenden kann, das ist ebenso selbstverständlich wie die Rauschsimulation oder die Stabilitätskontrolle anhand der S-Parameter oder der Parameter-Sweep. **Was aber einen HF-Fachmann sehr zur Einarbeitung reizt, ist speziell die enthaltene „Harmonic Balance Simulation“.** Sie steht in keiner anderen Software kostenlos zur Verfügung und so macht das nebenstehende Bild mit den Simulationsmöglichkeiten großen Appetit...

Wenn man sich mit Schaltungs - Simulations - Programmen etwas auskennt, dann fällt einem gleich

die hier herrschende Philosophie auf:

Es geht bei diesem Programm nicht darum, dem Schaltungsentwickler im Labor ein Werkzeug an die Hand zu geben, mit dem sich gegenüber einem Konkurrenzprodukt wieder einige Sekunden an Arbeitszeit einsparen lassen. Dieses Programm will dagegen möglichst viele Werkzeuge bereitstellen, mit denen einer Schaltung gründlich auf den Zahn gefühlt werden kann. Das bedeutet, dass sehr viele Funktionen für die Simulation, aber auch unterschiedlichste Darstellungsmöglichkeiten bei den Ergebnissen bereit stehen und die Bedienung recht einfach ist, denn die Arbeit soll Freude machen und der Neugier wie auch der Fantasie Raum lassen.

Nun zur Frage: Wie schaffe ich am leichtesten den Einstieg?

Ganz einfach: in meiner Homepage (www.gunthard-kraus.de) gibt es ein zweibändiges kostenloses Einsteiger-Tutorial.

Band 1 umfaßt nun 220 Seiten und behandelt Simulationen in der Time Domain und in der Frequency Domain. Es beginnt mit einem einfachen RC-Tiefpaß für den Einsteiger und behandelt in Form von einzelnen Projekten mit steigenden Ansprüchen möglichst viele Themen der Elektronik und Nachrichtentechnik (vom Verstärker über Filter, von Drehstrom bis zu einer WLAN-Patchantenne, von der Modulation über Mischung bis zum Rauschen).

Band 2 ist etwas für Fortgeschrittene, denn da wird auf 50 Seiten gezeigt, was **Harmonic Balance** bedeutet und was es leisten kann. Da werden z. B. IP3 Punkte bei Mischern oder Verstärkern ermittelt, Mischerspektren oder S-Parameter bei unterschiedlicher Aussteuerung untersucht....usw.....usw...

2. Installation von QucsStudio

2.1. Installation daheim oder auf dem Notebook

Da reicht die Eingabe „qucsstudio“ in einer Suchmaschine, um auf der Seite „**QucsStudio Homepage – DD6UM**“ zu landen. Die aktuellste Version wird heruntergeladen, aber anschließend gibt es bisweilen ein Problem: **diese exe-Datei ist mit „7z“ gepackt und man muss sich da erst ein passendes Programm zum Auszippen im Internet suchen..**



Der Rest ist reine Installationsroutine und läuft ohne Probleme. Holt man sich anschließend den nebenstehenden Icon auf den Bildschirm und klickt darauf, dann hat man beim Aufruf eines Projektes folgenden Bildschirm vor sich:

Wire, Label, Ground, Port, Equation, DC Values, Simulation, Marker...

Schaltplan-Editor (Zeichenfläche)

Bauteile, Projekte, Quellen, Diagramme, Simulationen, Zeichenwerkzeuge....

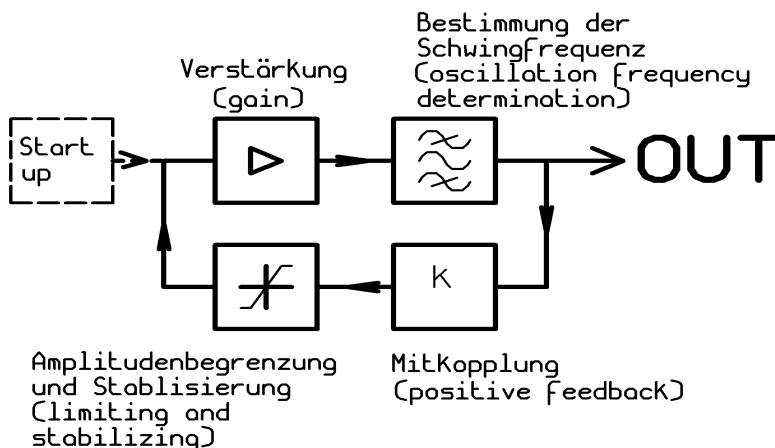
frequency: 35799.9
dBm(time2freq(U_emitter.Vt)): 20.52

Man sollte jedoch nicht versäumen, sich **zusätzlich in der Homepage umzusehen**: da gibt es da Bildschirmfotos und Videos, Beispiele, häufige Fragen usw. Und das hilft sehr oft weiter. Das Programm erzeugt keine Einträge in Windows (...also in der Registry) und kann deshalb ganz einfach durch Markieren seines Programm-Ordners, gefolgt von „Delete“, wieder von der Festplatte gelöscht werden.

3. Simulation eines Colpitts-Oszillators in der Time Domain

3.1. Oszillator-Prinzip

Am häufigsten wird bei Frequenzen unter 1000 MHz diese Anordnung gewählt:



Das Prinzip ist leicht zu verstehen: Die „Startup-Einheit“ führt dem folgenden Verstärker ein Signal mit der vorgesehenen Schwingfrequenz zu. Dieses Signal wird verstärkt und im nachgeschalteten Bandpass herausgefiltert (= „bandbegrenzt“).

Wichtig:

Dieses ausgesiebte Signal muß **gleichphasig** mit dem Eingangssignal sein und speist ein Mitkoppelnetzwerk, dem eine Amplitudenstabilisierung folgt.

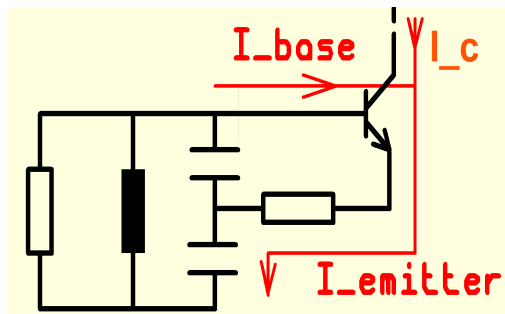
Nun kommt der Anteil „ $k \cdot U_{out}$ “ **gleichphasig zurück zum Verstärkereingang** und unterstützt das Startup-Signal.

Ist die „Schwingbedingung“ „ $k \cdot \text{Verstärkung} > 1$ “ erfüllt, dann erhält man ein stetig anwachsendes Ausgangssignal. Das Startup-Signal ist nun überflüssig, **weil „die Schaltung nun selbst ihren Eingang mit der nötigen Energie versorgt“**. Diesem stetigen Anwachsen gebietet nun irgendwann die „**Amplitudenbegrenzung und Stabilisierung**“ Einhalt und **sorgt dadurch für eine konstante Amplitude der Ausgangsspannung**.

Die Sache mit der „Starteinrichtung“ ist recht einfach: dazu genügt das im Transistor des Verstärkers erzeugte Eigenrauschen, das am Verstärkerausgang gefiltert und zum Eingang zurückgeschickt wird. Darin sind auch Anteile mit der Schwingfrequenz enthalten, die dann verstärkt und erneut „im Kreis herum“ zum Eingang gelangen. Das führt zu einem exponentiellen Anwachsen der Ausgangsspannung bis zur Begrenzung.

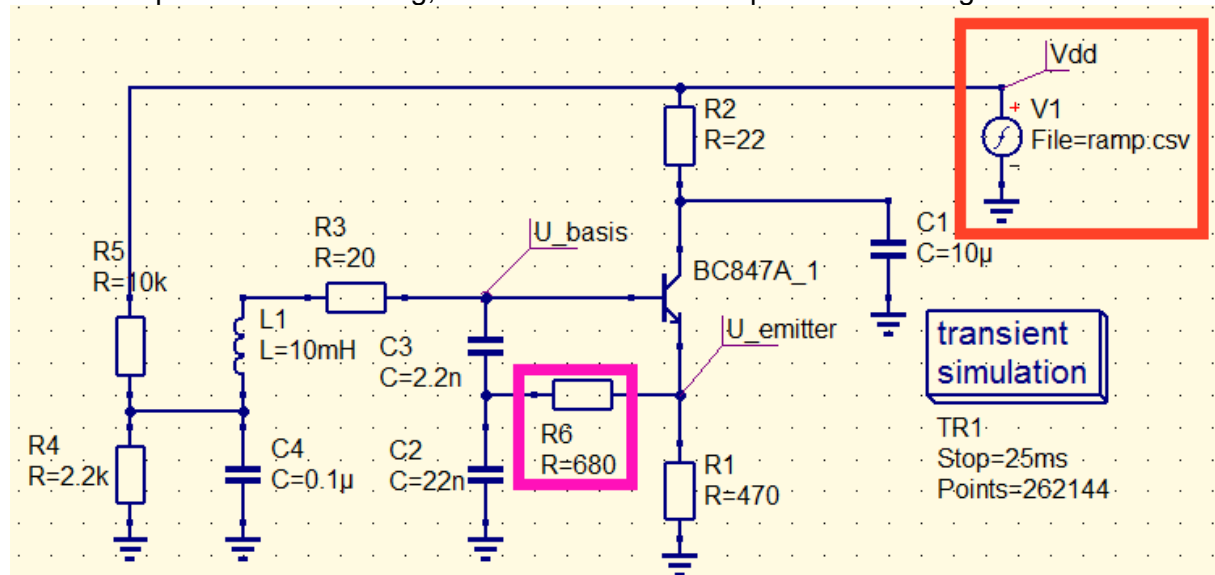
3.2. Simulierte Schaltung

Wir wollen eine Sinus-Oszillatorschaltung (Colpitts - Oszillator mit npn - Transistor BC847C in Kollektorschaltung) untersuchen. Hier die grundsätzliche Arbeitsweise:



Die Schwingkreisspannung liegt an der Basis des Transistors, der dadurch den Basisstrom " I_{base} " aufnimmt. Er stellt einen Stromverstärker dar und speist seinen (um die Stromverstärkung erhöhten) **gleichphasigen Emitterstrom in den Schwingkreis ein -- das deckt die Verluste**. Der Mitkoppelgrad " k " wird über die Größe des Widerstandes zwischen Schwingkreis und Emitter eingestellt.

Das ist die praktische Schaltung, erweitert um die Arbeitspunkt-Einstellung:



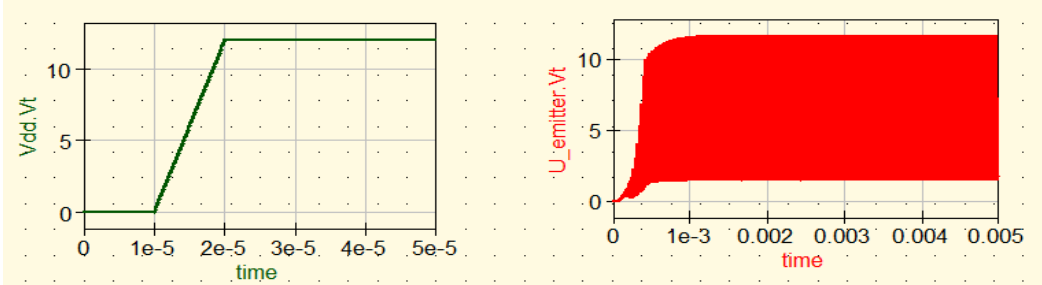
Da sieht man gleich (rot eingerahmt!) ein wichtiges Detail, denn die Simulation braucht als Startup-Signal einen "**Schubs**", **damit die Schaltung zum Schwingen kommt**. Das wird so gelöst:

Die Versorgungsspannung Vdd springt beim Start nicht sofort auf +12 Volt, sondern wird durch eine "**dateibasierte Spannungsquelle**" (= PWL source) nach einem **Rampenverlauf**

"hochgefahren". Er wird in der Datei "ramp.csv" abgelegt und sieht so aus:

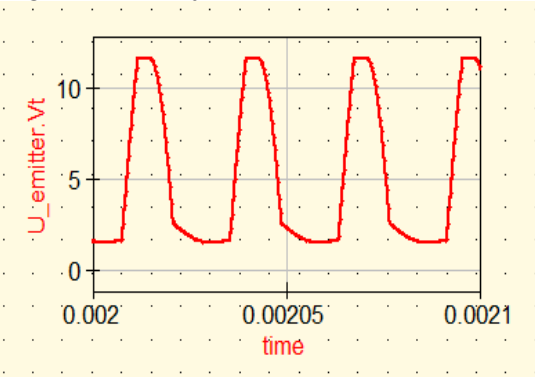
```
0      0
10e-6  0
20e-6  12V
40e-6  12V
```

Man sieht, dass die Versorgungsspannung Vdd bis $t = 10 \mu\text{s}$ auf Null bleibt, um dann in weiteren $10 \mu\text{s}$ linear bis auf +12V anzusteigen. Auf diesem Wert verharrt sie dann. Jetzt wird simuliert und anschließend der Verlauf der Emitterspannung in den ersten 5 Millisekunden dargestellt:



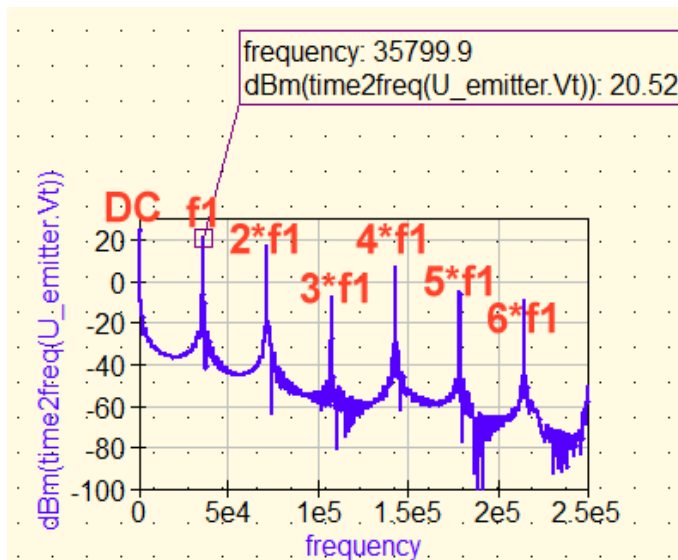
Hier haben wir diese Spannung und sehen sehr schön das

exponentielle "Anschwingen", gefolgt von der Begrenzung. Allerdings ist das Ganze irgendwie unsymmetrisch und das schauen wir uns näher an.



Da haben wir den Grund:

Die **Mitkopplung ist viel zu stark und deshalb wird der Transistor weit in die Sättigung und Begrenzung getrieben**. Die Folge ist eine stark einseitig verzerrte Kurvenform, die einen starken Oberwellenanteil befürchten lässt. Und die extreme Unsymmetrie lässt leider auch kräftige Amplituden bei den geradzahligen Harmonischen befürchten (...denn die fehlen vollständig bei perfekter Symmetrie..).



Also rufen wir die FFT auf mit **`dBm(time2freq(U_emitter.Vt))`**

und bekommen die Bestätigung für das vermutete Spektrum.

Übrigens:

Der im Schaltbild markierte Widerstand R6 / 680Ω stellt die Mitkopplung (=positive feed back) dar, denn über ihn fließt ein Teil des gleichphasigen Emitter-Wechselstromes in den Schwingkreis, deckt die Verluste und bewirkt so eine Dauerschwingung. Erhöht man seinen Wert, dann wird diese Mitkopplung schwächer. Dadurch wird zwar die

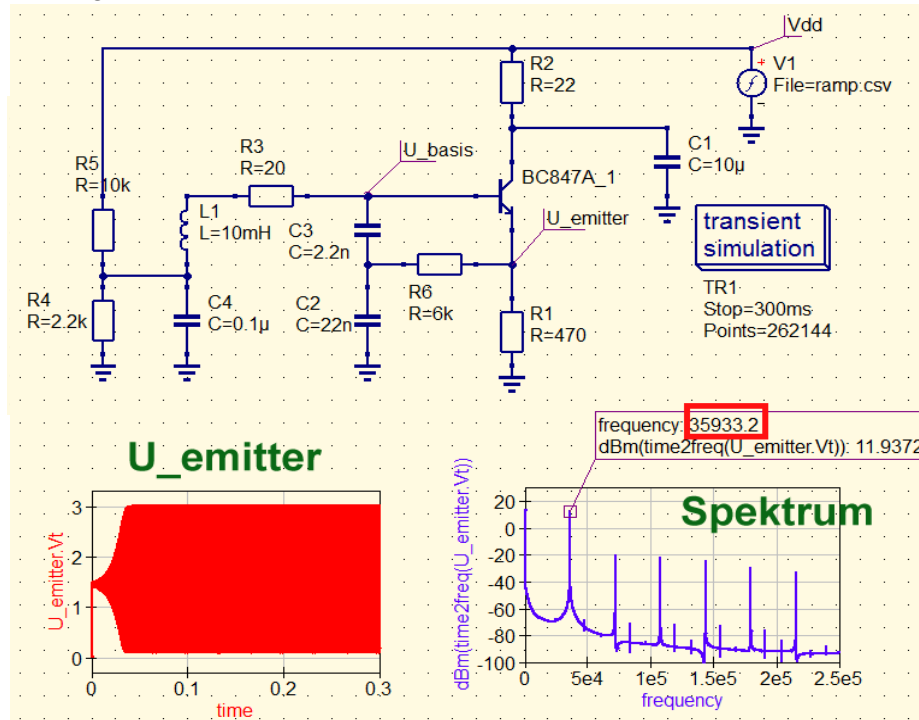
Kurvenform des Ausgangssignals verbessert und der Anteil der Oberwellen nimmt ab. Aber das Anschwingen dauert nun immer länger und irgendwann reicht die Mitkopplung nicht mehr zum Ausgleich der Verluste. Dann geht überhaupt nichts mehr

Man kann diese Suche nach dem optimalen Wert z. B. nacheinander mit unterschiedlichen Werten von R6 durchführen und jeweils den Verlauf der Ausgangsspannung analysieren.

Oder man programmiert einen **Parameter-Sweep**, bei dem man einen **Startwert und einen**

Stoppwert für R6 eingibt. Und dazu natürlich die Anzahl der Schritte, mit denen vom Minimalwert bis zum Maximalwert „gesweept“ wird. Geht schnell, aber alle Ergebniskurven werden in gleicher Farbe übereinander geschrieben...und das muss man hinterher genau auseinander pflücken....

Lösung:

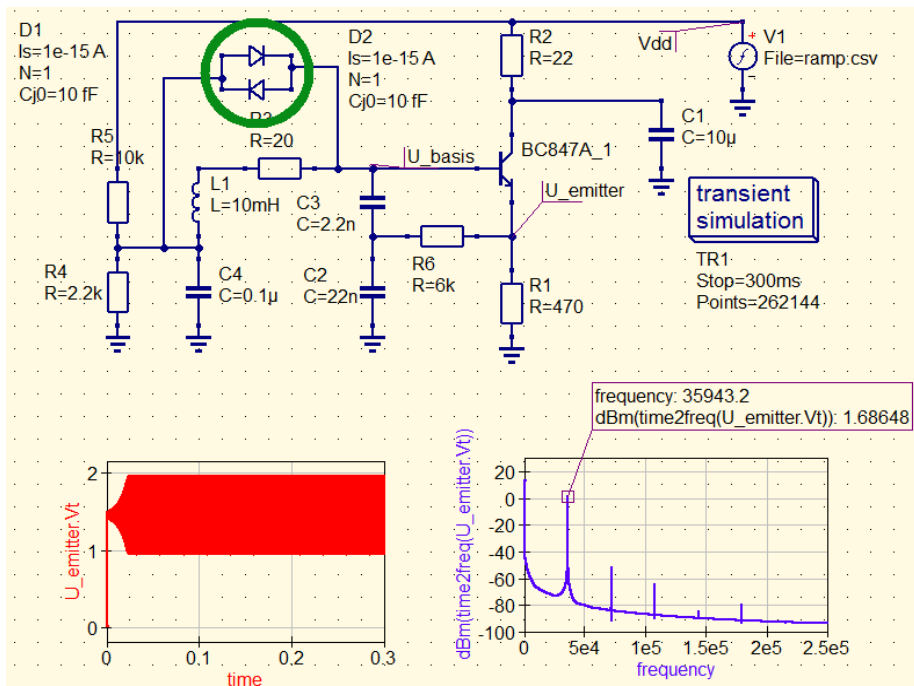


Man sieht, dass bei R6 = 6 kΩ bereits deutlich mehr als 25 Millisekunden bis zum Anstieg auf die maximale Ausgangsspannung nötig sind. Auch im Spektrum erkennt man nun deutlich die Amplitudenreduktion bei den Oberwellen.

Und die Verzerrungen der Sinusform sind nun deutlich zurückgegangen. Lediglich die Amplitude muss noch reduziert

werden, um die eingedrückte negative Spitze zu beseitigen (= korrekte Amplitudenstabilisierung nötig)

Auch die Schwingfrequenz können wir dem Spektrum entnehmen, indem wir die Linie für die Grundwelle mit einem „Frequenzmarker“ versehen.



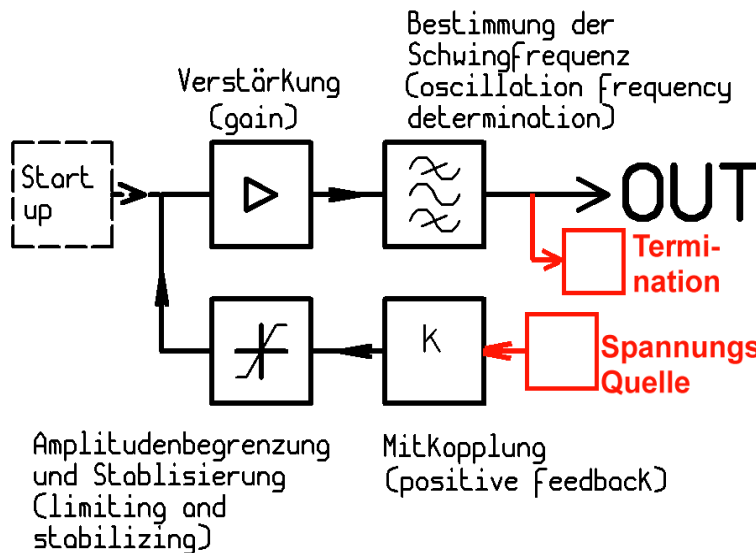
Perfekt wird die Sache aber erst, wenn wir z. B. eine zusätzliche Amplitudenstabilisierung und Begrenzung mit zwei antiparallel geschalteten Schottky-Dioden über dem Schwingkreis dazufügen:

Man beachte nun die Dämpfung der Oberwellen....

4. Simulation des Colpitts-Oszillators in der Frequency Domain

Die vorige Simulation in der Time Domain liefert uns alle Informationen über den zeitlichen Verlauf des erzeugten Oszillatorsignals (wie Amplitude, Frequenz usw.). Aber wir sehen auf Anhieb nicht, wie weit wir die „Mitkopplung“ (= positive feedback) vermindern dürfen, um die Kurvenform zu verbessern und die Übersteuerung zu verhindern.

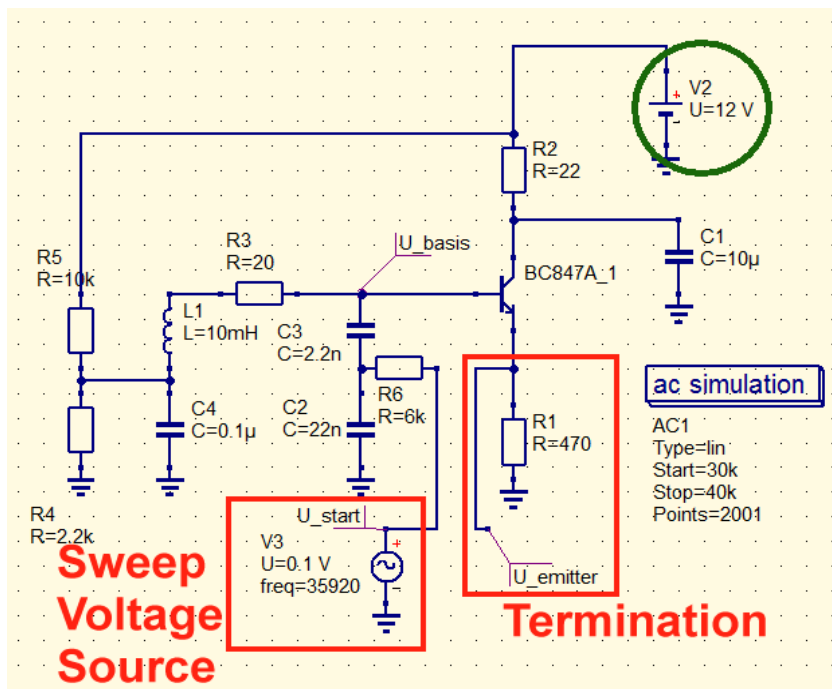
Eine Simulation in der **Frequency Domain** liefert uns diese und weitere Informationen, wie z. B. die „Phasensteilheit“ (...dadurch werden die Rauschseitenbänder wesentlich beeinflusst). Aber: wie geht das?



Dazu holen wir uns nochmals den Übersichtsschaltplan aus der Einleitung im ersten Kapitel und **trennen die Verbindung zur Mitkopplung auf**.

Eine **getrennte neue Spannungsquelle** spielt nun die Rolle des „positive feedbacks“ und wir messen, **was dann bei der Termination ankommt**. Die bildet nämlich den Eingangswiderstand der Mitkopplung nach. Und wenn wir nun auch noch den **Innenwiderstand der**

Spannungsquelle genau so wählen wie „den Rückblick vom Termination-Punkt in Richtung OUT“, dann messen wir exakt die „Ringverstärkung“ (**loop gain**) der Anordnung. **Diese „Loop Gain“ muss größer als 1 (bzw. größer als Null dB) sein und die Phase „Null Grad“ betragen, wenn die Anordnung eine exponentiell ansteigende Dauer-Schwingung erzeugen soll.**



So sieht unsere Schaltung für den AC-Sweep aus:

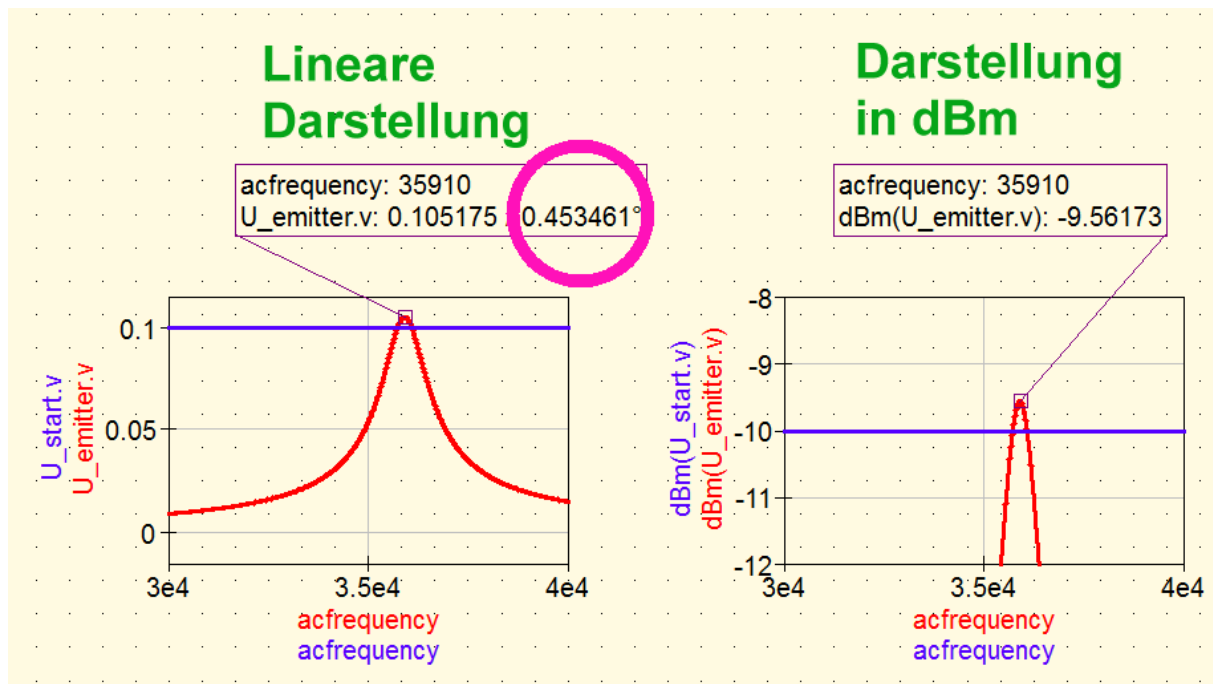
a) **Die Betriebsspannung muss nun konstant sein. Sie wird auf +12V eingestellt.**

b) Bei der „Termination“ haben wir gar keine Arbeit, denn der **Innenwiderstand des Transistors beträgt bei seinen 3 mA Ruhestrom nur ca. 8 Ω** und da brauchen wir den jetzt fehlenden R6 gegenüber dem R1 nicht berücksichtigen.

c) Ebenso spielen die eben erwähnten 8 Ω beim Ri

gegenüber R6 = 6 kΩ keine Rolle für die Spannungsquelle.

Gesweept wird von 30 kHz bis 40 kHz, die Generatorspannung wird dazu auf $U = 100 \text{ mV}$ eingestellt. Und das kommt dabei heraus:



Die **Loop Gain ist größer als 1 (bzw. Null dB)** und damit wissen wir, dass die Schaltung schwingen wird.

Aber bitte genau hinsehen:

a) Im linken Bild sehen wir (im violetten Kreis), dass auch die **Phasenbedingung** korrekt erfüllt ist: **wir beobachten an der Termination ein gleichphasiges Signal.**

b) Im rechten Bild sehen wir die „**Verstärkungs-Reserve**“ von **0,5 dB**. Das reicht zum Schwingen, treibt aber den Transistor nicht brutal in die Sättigung.

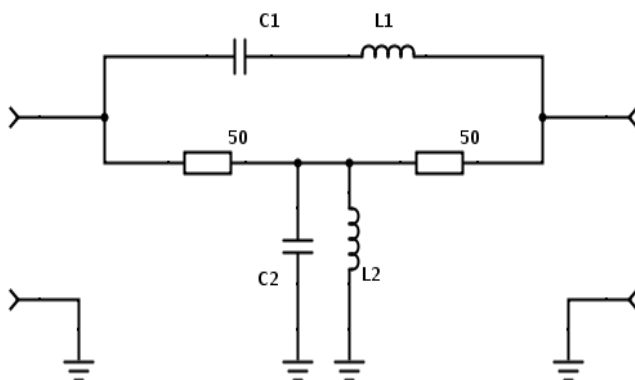
5. S-Parameter-Simulation einer Diplexer-Schaltung in der Frequency Domain

Double Balanced Mixer mit Dioden verlangen immer einen korrekten breitbandigen Abschluß mit dem **Systemwiderstand (meist. $Z = 50\Omega$) an allen Ports**, um optimal und mit geringsten Verzerrungen zu arbeiten. Nur so erreicht man die im Datenblatt angegebenen Spezifikationen. Die dazu gehörigen Entwicklungsprozeduren und Simulationen verwenden stets die „**S-Parameter**“.

Wir wollen aber als Beispiel am ZF-Ausgang (= „IF Port“) das USB-Signal als Zwischenfrequenz mit $f = 100\text{ MHz}$ weiterverarbeiten und brauchen dazu eine Schaltung, die

a) einen **Bandpass für das gewünschte IF-Signal** (hier: $f = 100\text{ MHz}$) bildet, aber

b) trotzdem einen korrekten breitbandigen (und möglichst kobstanten) Eingangswiderstand von **50Ω für den Mischerausgang bereitstellt**.



Diese Schaltung trägt den Namen „**Diplexer**“ und im Internet gibt es dazu passende Online – Calculatoren.

Wir geben darin unsere **Frequenz mit 100 MHz**,

den Systemwiderstand mit **$Z = 50\Omega$**

und die vom Calculator vorgeschlagene **Güte von $Q = 10$** für HF-Schaltungen ein.

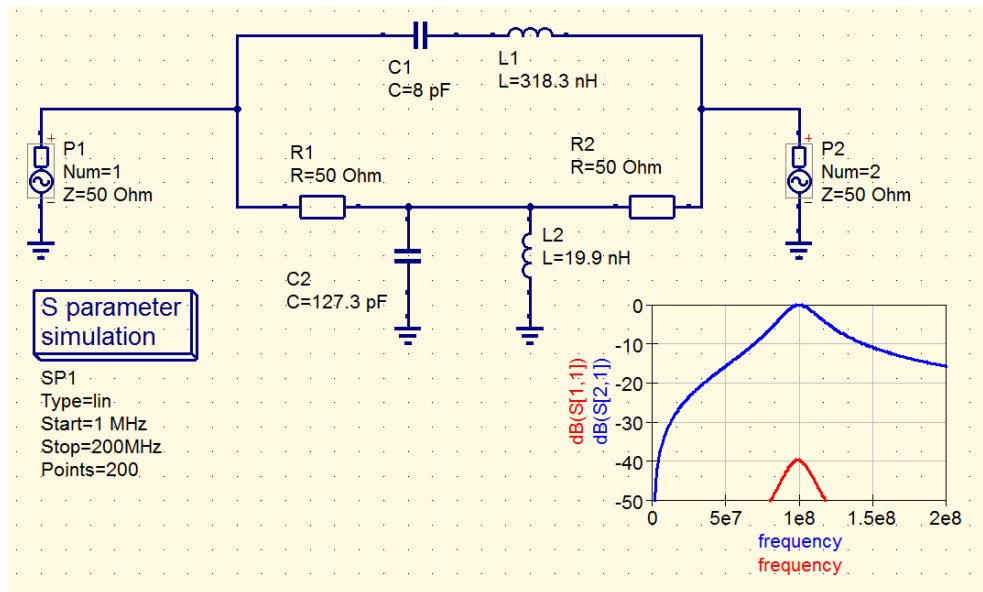
Dann bietet der Calculator automatisch die nötigen Bauteilewerte an.

Anschließend erstellen wir diese Schaltung in qucsstudio und führen eine **S-Parameter-Simulation** durch.

Dazu müssen wir lediglich die Schaltung

FREQUENCY	100	[MHz]
IMPEDANCE	50	[Ω]
Q	10	[use 10 for RF, 2...4 for VHF]
C ₁	3.2	[pF]
L ₁	795.8	[nH]
C ₂	318.3	[pF]
L ₂	8	[nH]
RESISTOR	50	[Ω]

zeichnen, sie mit zwei „**Microwave Ports**“ versehen, eine „S-Parameter-Simulation“ aufrufen und den gesweepen Frequenzbereich auf 1 MHz bis 200 MHz (mit 200 Simulationenpunkten) festlegen.



Wie man sieht, bleibt die Eingangsreflektion unter -40 dB. Und wenn wir ideale Bauteile verwenden, verursacht die Schaltung auch keine zusätzliche Dämpfung am IF-Ausgang, wenn wir das 100 MHz-Signal auskoppeln.

6. Harmonic Balance

6.1. Wozu?

Simulationen im Zeitbereich (= Time Domain) liefern nicht nur Informationen über den zeitlichen Verlauf aller beteiligten Spannungen und Ströme, sondern zeigen über eine nachfolgende FFT (= Fast Fourier Transformation) auch die durch Nichtlinearitäten entstehenden neuen Frequenzen – **aber immer nur für die vorgegebene und konstante Frequenz der verwendeten Eingangssignale.**

Ein „AC-Sweep“ ignoriert dagegen alle Nichtlinearitäten und liefert den „Frequenzgang der Übertragungsfunktion“ für das sinusförmige und „gesweepete“ Ansteuersignal in der Frequency Domain.

Die dadurch noch vorhandene Lücke (= die zusätzliche Simulation der Nichtlinearitäten bei sich ändernder Speisefrequenz) schließt „Harmonic Balance“ und das ist in qucsstudio integriert. Aber: dieses Programm simuliert IMMER in der Frequency Domain“!

6.2. Wie geht das?

Das ist eine raffinierte und deshalb auch patentierte Idee, denn wenn in einer Schaltung lineare Bauteile (z. B. Widerstände, Spulen, Kondensatoren...) und nichtlineare Bauteile (z. B. Dioden, Transistoren, FETs...) zusammen eingesetzt werden, geht das Programm folgendermaßen vor:

- Lineare Bauteile** faßt man samt ihren Verbindungsleitungen und Schaltungsknoten komplett in einem „**linearen Block**“ (= linear subcircuit) zusammen.
- Alle **nichtlinearen Bauteile** samt ihren Verbindungsleitungen und Schaltungsknoten kommen in eine zweite Kiste mit dem Namen „**nichtlinearer Block**“ (nonlinear subcircuit).
- Dann werden beide Blöcke miteinander über deutlich erkennbare (und exakt benannte) zusätzliche Leitungen (= connections) miteinander verbunden – damit ist die ursprüngliche Startschaltung wieder korrekt hergestellt. Nicht vergessen darf man natürlich, noch die gewünschten Eingangssignale anzulegen.

Sehr schön ist das in diesem Bild aus [1] zu erkennen:

Let us define the following symbols:

- M = number of (independent) voltage sources
- N = number of connections between linear and non-linear subcircuit
- K = number of calculated harmonics
- L = number of nodes in linear subcircuit

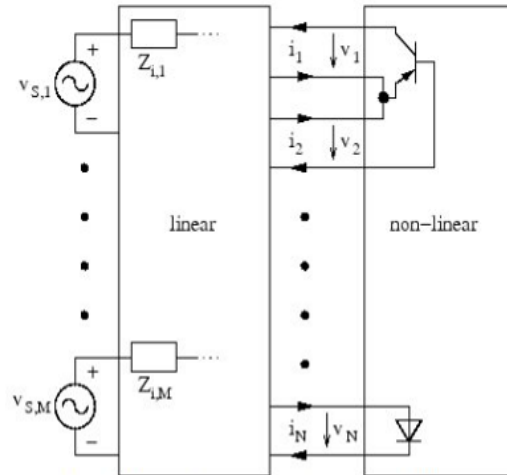


Figure 7.1: circuit partitioning in harmonic balance

Die Frage „Und jetzt?“ bekommt eine verblüffende Antwort:

Alles im **linearen Block** kann direkt (über die Bauteilmodelle) in der **Frequency Domain** berechnet werden.

Alle Berechnungen und Simulationen im **nichtlinearen Block** werden dagegen (nach der Übergabe der Einzelsignale aus dem linearen Block über die „connections“) im **Zeitbereich** durchgeführt. Dadurch erhält man die „durch Nichtlinearitäten verzerrten Stromverläufe“ – und nach einer Rücktransformation jedes Stromes können die Ergebnisse wieder in Form der Grundwelle und jeder entstandenen Oberwellen an den linearen Block **im Frequenzbereich zurückgegeben** werden.

Man ahnt, dass der erforderliche Rechenaufwand enorm ist. Es muss nämlich erst mal vom Anwender vor Beginn angegeben, welcher höchste Oberwellengrad berücksichtigt werden soll. Dann werden in Form von „mehrdimensionalen Matrizen“ die „Spektren für alle Verbindungen zwischen linearem und nichtlinearem Block“ berechnet. Das ginge ja noch, aber:

Bei jeder Verbindungsleitung (connection) muß folgende Bedingung erfüllt werden:

Was da reingeht, muß auch wieder herauskommen!

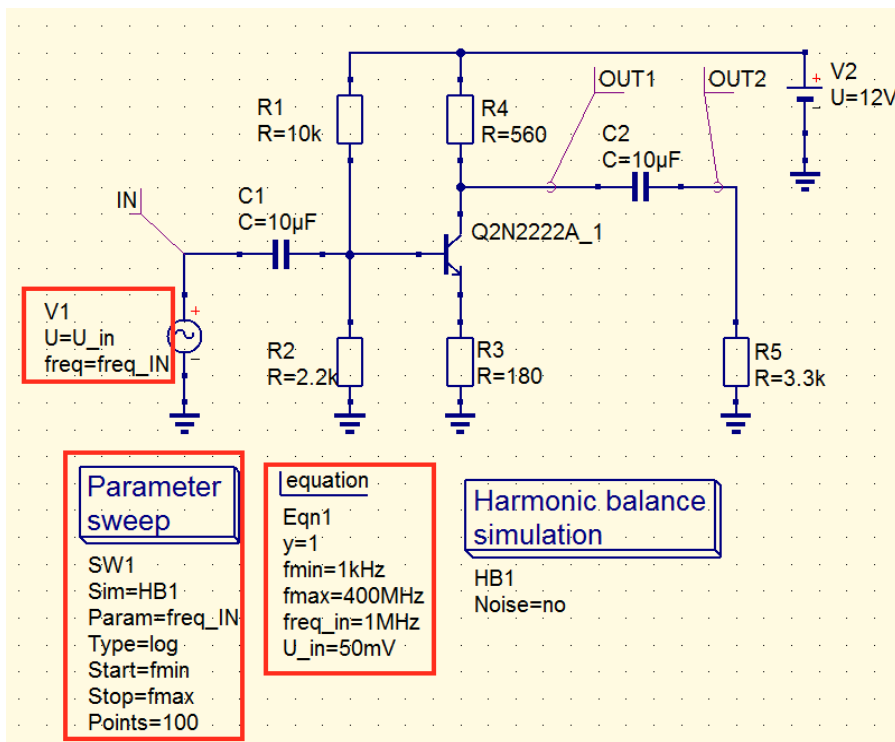
Das bedeutet im Klartext:

Was der lineare Teil (z. B. als Leistung) in Form eines Stromes über eine Verbindungsleitung zum nichtlinearen Teil schickt, muss genau so groß sein wie das vom nichtlinearen Teil nach der Fourier Transformation in dieser Leitung zurückgegebene Spektrum (= Summe der Leistungen von Grundwelle und zugelassenen Oberwellen).

Das funktioniert leider nicht beim ersten Anlauf, denn das Programm startet mit (ungefähren) Vorgabewerten aus dem linearen Teil, die dem nichtlinearen Teil angeboten werden. Dort wird im Zeitbereich gerechnet, es werden die nötigen Fourier Transformationen durchgeführt und hinterher die Leistungsbilanzen bei allen Verbindungsleitungen erstellt. Geht die Rechnung nicht auf (= die vom linearen Teil hineingeschickte Leistung stimmt noch nicht mit der vom nichtlinearen Teil als Spektrum dargestellten Leistung überein), dann wiederholt das Programm mit geänderten Werten die ganze Prozedur. Diese „Iteration“ wird so lange ausgeführt, bis das Gleichgewicht (= also die „Harmonic Balance“) erreicht und der Fehler minimal ist. Und das kann dauern, denn linearer und nichtlinearer

Teil beeinflussen sich gegenseitig – schließlich handelt es sich ja um EINE zusammengehörige Schaltung.....man ahnt: die Theorie und der mathematische Hintergrund sind recht kompliziert. Die sich ergebenden Möglichkeiten einer erfolgreichen Harmonic Balance Simulation sind jedoch faszinierend. Wenn man **zusätzlich mit einem „Parameter Sweep“ arbeitet**, hat man anschließend einen riesigen und kompletten Ergebnis-Datensatz **für die Ansteuerung mit Signalamplituden von „sehr klein“ bis „jetzt reicht es aber..“ zur Verfügung**. Damit kann man z. B. den Frequenzgang der Verstärkung bei kleiner und bei großer Eingangsspannung vergleichen, kann den 1 dB Kompressionspunkt ermitteln, kann den IP3-Punkt errechnen lassen, kann sich Oberwellen-Spektren über der Frequenz ansehen.....usw.usw.

6.3. Harmonic Balance-Simulation in Frage und Antwort bei einer Verstärkerstufe



Da schauen wir einfach in die aus der qucsstudio-Homepage heruntergeladenen Harmonic Balance-Beispiele, denn darin findet sich ein **„simple_amplifier.sch“ und der bildet auch das Thema im letzten Kapitel des Tutorials**. Das paßt gut und wir ersetzen lediglich die Amplitude der Eingangsspannung durch eine Variable „U_in“.

Die Unterkapitel tragen folgende Überschriften:

Wie finde ich schnell alle Spannungen im DC-Arbeitspunkt?

Welche Vorbereitungen sind für die HB-Simulation erforderlich?

Wie stelle ich die Ein- und Ausgangsspannung einschließlich der Verstärkung (linear bzw. in dBm / dB) dar?

Ich möchte den Signal / Rauschabstand (= signal to noise ratio) in dB am Ausgang bei f = 1MHz sehen

Wie sieht die Schaltung für einen Parameter-Sweep der Eingangsspannung aus?

Parameter-Sweep der Eingangsspannung: wie sieht dafür die Ausgangsspannung und die Transfer-Funktion aus?

Parameter-Sweep der Eingangsspannung: Ich möchte ausser der Grundwelle auch zwei Oberwellen (= doppelte und dreifache Frequenz) sehen

Parameter-Sweep der Eingangsspannung: wie kann ich den **ungefähren Wert des Output IP3 Punktes ganz schnell abschätzen?**

Parameter-Sweep der Eingangsspannung: Wie verändert sich die **Rauschspannung am Ausgang** mit zunehmender Aussteuerung?

Parameter-Sweep der Eingangsspannung: Wie simuliere ich den **Signal-Rauschabstand am Ausgang?**

Parameter-Sweep der Eingangsspannung: Wie bestimme ich den **Signal-Rauschabstand bei einer Messbandbreite von 1000 Hz?**

Parameter-Sweep der **Frequenz**: Vorbereitung der HB-Simulation

Parameter-Simulation der **Frequenz**: Verlauf der Ausgangsspannung **und der Verstärkung**

Gleichzeitiger Parametersweep von **Frequenz UND Eingangsspannung**

=====

Parameter sweep

SW2
Sim=SW1
Param=U_in
Type=lin
Start=50mV
Stop=1500mV
Points=5

Beispiel: Letztes Kapitel „7.14. Gleichzeitiger Parametersweep von Frequenz UND Eingangsspannung“

Das ist sozusagen die Krönung und wir müssen dazu die Schaltung lediglich um diesen **zusätzlichen Parameter-Sweep für die Eingangsspannung** „U_in“ erweitern.

Gesweept wird von 50 mV bis 1500 mV in 5 Schritten.

Die Folgen sind schon sehr beeindruckend:

