

PSpice

Einbinden von Bibliotheken

Kathleen Jerchel

9. April 2007

Inhaltsverzeichnis

1	Einleitung	2
2	Theorie	2
2.1	Was braucht man?	2
2.2	Wie beschaffen?	2
2.3	Weitere Vorgehensweise	2
2.3.1	Die Bauteilinformationen habe ich als Text gefunden.	2
2.3.2	Ich habe eine lib-Datei gefunden.	3
2.3.3	Ich habe eine lib- und eine olb-Datei gefunden.	3
3	Praxis	4
3.1	Finden von Bauteilinformationen	4
3.1.1	Google-Suche	4
3.1.2	Simulations- und Bauelementeinformationen auffinden	4
3.2	.lib-Datei erstellen	6
3.2.1	Editor öffnen	6
3.2.2	Infos einfügen	6
3.2.3	Infos als .lib-Datei speichern	6
3.3	.olb-Datei erstellen	7
3.3.1	PSpice Model Editor öffnen	7
3.3.2	.lib-Datei öffnen	7
3.3.3	.olb-Datei erstellen	7
3.4	Einfügen der Bauteile in PSpice	9
3.4.1	Bauelemente in Schaltplan einfügen	9
3.4.2	Bauelemente in Simulationseinstellungen einfügen	9

1 Einleitung

Manchmal sind die Bauteile, welche für ein Projekt benötigt werden gar nicht in den Standardbibliotheken von PSpice enthalten. Da man dennoch vor dem Testaufbau simulieren möchte, wird hier erklärt, wie Bibliotheken gefunden, erstellt und eingebunden werden.

2 Theorie

2.1 Was braucht man?

PSpice braucht zwei Dateitypen um Schaltpläne simulieren zu können: lib- und olb-Dateien. Beides sind Bibliotheken. Die Informationen, wie PSpice mit einem Bauelement bei der Simulation umgehen soll können auch in einer Textdatei stehen. In diesem Fall muss man diese Informationen in die nötigen Dateiformate umwandeln.

2.2 Wie beschaffen?

Die einfachste Möglichkeit ist es auf www.google.de oder einer anderen Suchmaschine nach der Bauteilbezeichnung und „Model“, „PSpice“ und „lib“ zu suchen (z.B. BC547 model pspice). Dann muss man sich aus den Suchergebnissen das Beste aussuchen, was durchaus ein bisschen Zeit in Anspruch nehmen kann. Wichtig ist, dass die lib- und olb-Datei den gleichen Namen haben, z.B. BC547Bnpn.lib und BC547Bnpn.olb.

2.3 Weitere Vorgehensweise

Je nachdem, welche Art der Datei man gefunden hat, muss man verschiedene Schritte unternehmen, um zum Ergebnis zu kommen. Die nächsten Schritte in dieser Anleitung gehen davon aus, dass der Ausgangspunkt eine Textdatei mit den Bauteilsimulationswerten ist. Wenn man andere Dateitypen gefunden hat, kann man Schritte in dieser Anleitung auslassen.

2.3.1 Die Bauteilinformationen habe ich als Text gefunden.

Dieser Text hat in der Regel folgende Form:

```
*  
.MODEL QBC547B NPN(  
+ IS=2.39E-14  
+ NF=1.008  
+ ISE=3.545E-15  
+ NE=1.541  
+ BF=294.3  
+ IKF=0.1357  
+ VAF=63.2  
+ CJS=0  
+ VJS=0.75  
+ MJS=0.333  
+ FC=0.9579 )  
*
```

Die Sternchen markieren nur Anfang und Ende und sind nicht so wichtig. Wenn man einen solchen Text gefunden hat dann öffnet man zusätzlich einen Texteditor und kopiert den Text mit den Bauteilinformationen in ein neues Dokument hinein. Anschließend speichert man dieses Dokument als lib-Datei. Das funktioniert in dem man unter Datei auf Speichern unter klickt, dort unter Dateityp stellt man alle Dateien ein und vergibt unter Dateinamen einen angemessenen Namen. In diesem Fall ist ein angemessener Name BC547Bnpn.lib. Dadurch hat man jetzt aus einer einfachen Textinformation eine lib-Datei gemacht, welche PSpice verarbeiten kann.

2.3.2 Ich habe eine lib-Datei gefunden.

Der Anfang ist gemacht. Jetzt muss aus dieser lib-Datei eine olb-Datei erzeugt werden. In PSpice Student ist auch das Programm PSpice Model Editor Student enthalten. Dieses muss nun gestartet werden. Dann kann man über File Open die lib-Datei öffnen, welche in eine olb-Datei umgewandelt werden muss. Wenn diese Datei geöffnet ist, muss man über File Create Capture Parts ein neues Dialogfenster öffnen. In diesem kann man nun den Pfad der lib-Datei angeben und der Pfad und der Name der zu erstellenden olb-Datei wird automatisch eingefügt. Nachdem man dieses Dialogfenster mit OK geschlossen hat, wird eine olb-Datei hergestellt. Es öffnet sich ein neues Fenster, in dem man nachlesen kann, ob alles geklappt hat. Wenn keine Fehler aufgetreten sind, hat man jetzt zu seiner lib-Datei eine zugehörige olb-Datei.

2.3.3 Ich habe eine lib- und eine olb-Datei gefunden.

Dann müssen diese beiden Bibliotheken jetzt nur noch in PSpice eingefügt werden. Es gibt zwei Orte an denen diese Bibliotheken eingebunden werden müssen. Einmal im Schaltplan bei der Auswahl der einzelnen Bauelemente und bei den Simulationseinstellungen.

1. Schaltplan Wenn man ein neues Bauteil einfügen möchte, muss man auf die Schaltfläche Place part klicken. Danach öffnet sich ein Dialogfenster, in welchem man Bauteile aus einer Liste auswählen kann. Auf der rechten Seite des Dialogfensters gibt es eine Schaltfläche Add Library. Wenn man eine neue Bibliothek hinzufügen möchte, muss man darauf klicken und im sich öffnenden Fenster die gewünschte olb-Bibliothek auswählen. Nachdem man bestätigt hat, kann man auch Bauelemente aus dieser Bibliothek auswählen.

Simulationseinstellungen Unter den Simulationseinstellungen gibt es einen Punkt der Libraries heißt. Diesen wählt man aus und trägt unter Filename die lib-Datei ein, welche neu eingefügt werden soll. Wichtig ist, dass dort wirklich der gesamte Pfad steht. Im Fall unseres BC547B NPN-Transistors: C:\Dokumente und Einstellungen\User\Desktop\BC547Bnpn.lib. Anschließend klickt man auf die Schaltfläche Add as Global und die lib-Bibliothek wird der Liste der Bibliotheken hinzugefügt. Dann müssen diese Einstellungen noch übernommen werden und natürlich auch die anderen Simulationseinstellungen gemacht werden, um anschließend simulieren zu können.

3 Praxis

Damit diese Anleitung nicht so theoretisch ist, wird an dem Beispiel des BC547b NPN-Transistors das Finden, Erstellen und Einfügen der Bibliotheken noch mal gezeigt.

3.1 Finden von Bauteilinformationen

3.1.1 Google-Suche

In Abbildung 1 kann man sehen, dass nach „bc547 pspice model“ auf www.google.de gesucht wurde. Das erste Suchergebnis ist schon sehr vielversprechend, also folgt man dem Link.



Abbildung 1: Suchergebnis auf www.google.com

3.1.2 Simulations- und Bauelementinformationen auffinden

Man wird von Google auf die Philips-Semiconductor-Webseite weitergeleitet. Auf dieser Seite muss man erstmal ein bißchen nach dem PSpice-Model suchen, wird aber am Ende der Seite fündig. Siehe Abbildung 2. Dort klickt man wieder auf den Link. Wichtig: Nicht aufgeben, wenn man nicht gleich das Ergebnis hat, was man wollte. Bei der Suche nach Bibliotheken und generell bei der Benutzung des Programms PSpice muss man sehr geduldig sein.

Auf dieser Seite kann man nun endlich den Transistor den wir suchen (BC547) aussuchen. In dem kleinen Fenster, das auch in Abbildung 3 zu sehen ist, kann man die Parameter des Transistors sehen. Das sind genau die Informationen, die wir brauchen um eine Bibliothek zu erstellen. Jetzt geht man wie folgt vor:

1. in das kleine Fenster klicken
2. mit Strg+a alles in dem kleinen Fenster markieren
3. mit Strg+c alles was markiert ist kopieren

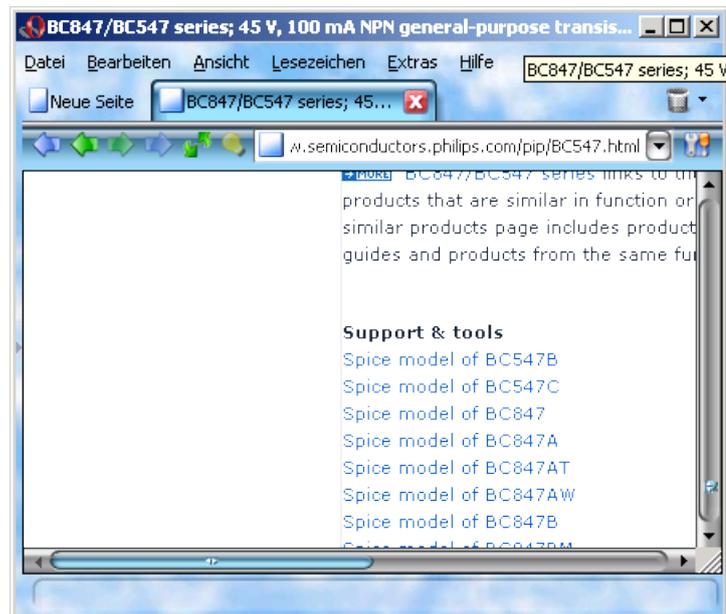


Abbildung 2: PSpice-Modells auf der Philips-Seite

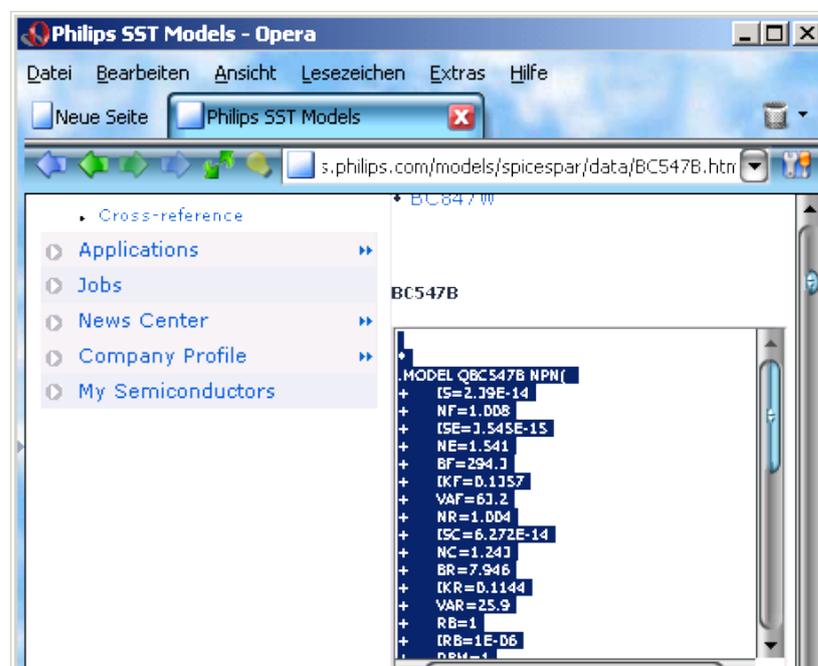


Abbildung 3: BC547-Modell auswählen

3.2 .lib-Datei erstellen

3.2.1 Editor öffnen

Als nächstes öffnet man einen Texteditor, am besten den von Windows. Den findet man unter Start → Programme → Zubehör → Editor.

3.2.2 Infos einfügen

Durch die Tastenkombination Strg+v fügt man die eben aus der Internetseite kopierten Model-Informationen in den Editor ein. Aussehen sollte das dann wie in Abbildung 4.

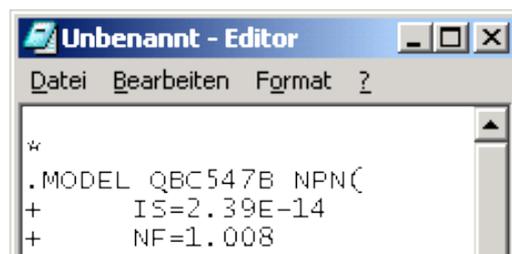


Abbildung 4: In Editor eingefügtes Model

3.2.3 Infos als .lib-Datei speichern

Jetzt muss man das ganze noch als .lib-Datei speichern. Das geht eigentlich ganz einfach, in dem man unter Datei → Speichern unter... klickt. Dann öffnet sich ein Dialogfenster wie in Abbildung 5.

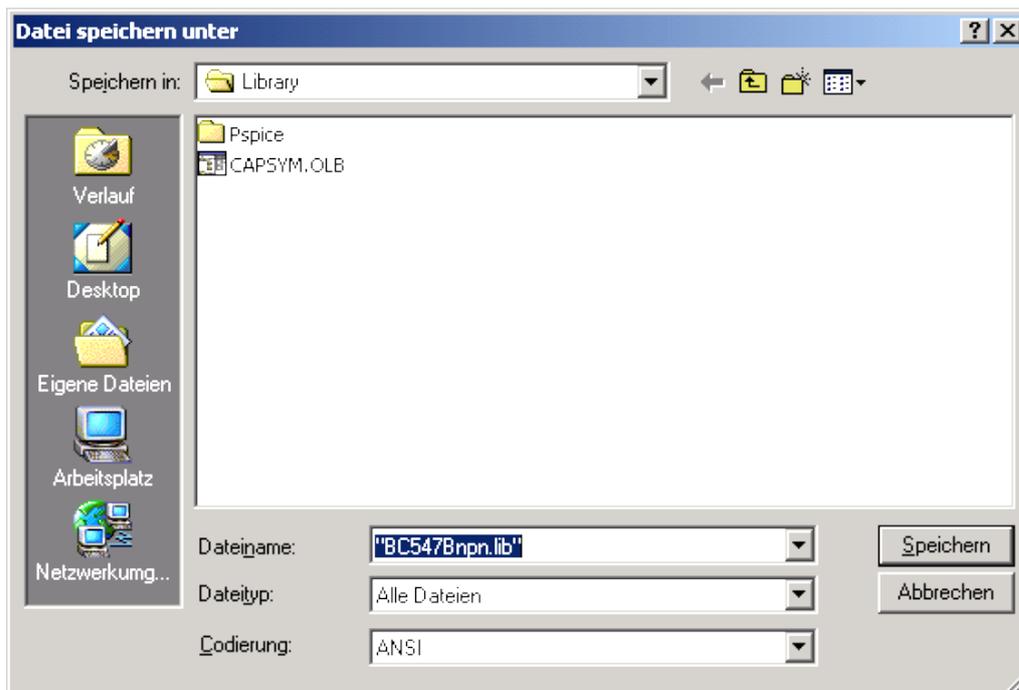


Abbildung 5: Speichern unter... im Editor

Hier muss man darauf achten, dass als Dateiendung `.lib` steht. Denn das ist das Format mit dem der Model-Editor von PSpice umgehen kann. Manchmal funktioniert die Auswahl „Alle Dateien“ beim Editor nicht. Um zu forcieren, dass der Editor die Datei wirklich als `.lib`-Datei speichert, kann man den Dateinamen inklusive der Dateiendung in Anführungszeichen setzen. Hier wurde der Name der Datei `BC547B.lib` gewählt. Diese Datei in einem Verzeichnis Deiner Wahl speichern. Am Besten speichert man jedoch in `C:\Program Files\Orcad_Demo\Capture\Libraries`. Das ist das Verzeichnis, in dem für gewöhnlich die Bibliotheken abgespeichert sind.

Wenn man die Datei richtig gespeichert hat, dann erscheint der richtige Dateiname mit der richtigen Dateiendung oben im Programmfenster, siehe Abbildung 6.

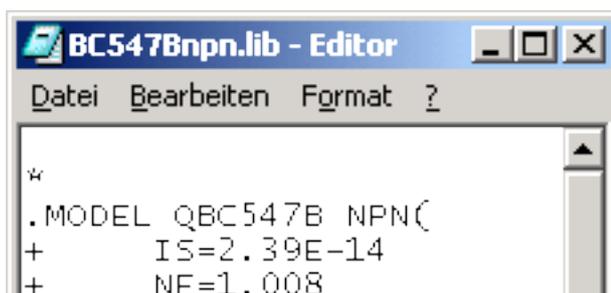


Abbildung 6: gespeicherte `.lib`-Datei

3.3 .olib-Datei erstellen

Nachdem die Datei `BC547B.lib` erstellt wurde müssen wir noch die dazugehörige `.olib`-Datei erstellen. Dies wird in den folgenden Abschnitten erklärt.

3.3.1 PSpice Model Editor öffnen

Man muss jetzt den PSpice Model Editor öffnen, der ebenfalls mit PSpice installiert wird. Man findet ihn also unter `Start → Programme → PSpice Student → PSpice Model Editor Student`. Jetzt mit Doppelklick das Programm starten.

3.3.2 .lib-Datei öffnen

Nachdem das Programm PSpice Model Editor gestartet wurde, muss das `BC547.lib` Model geöffnet werden. Dies erreicht man, indem man mit `File → Open` das `BC547.lib` Model in dem jeweiligen Verzeichnis öffnet und dann im Model Editor dieses Model im linken Fenster noch auswählt. Wenn man alles richtig gemacht hat, muss der Model Editor aussehen wie in Abbildung 7.

3.3.3 .olib-Datei erstellen

Jetzt wird die `BC547B.lib`-Bibliothek erstellt, in dem man über das Menü unter `File` auf `Create Capture Parts` klickt. Dann öffnet sich ein neues Fenster, welches man in Abbildung 8 sehen kann. Über `Browse` in der oberen Zeile (Enter Input Model Library) muss man nun die `BC547B.lib` auswählen. Das Programm füllt

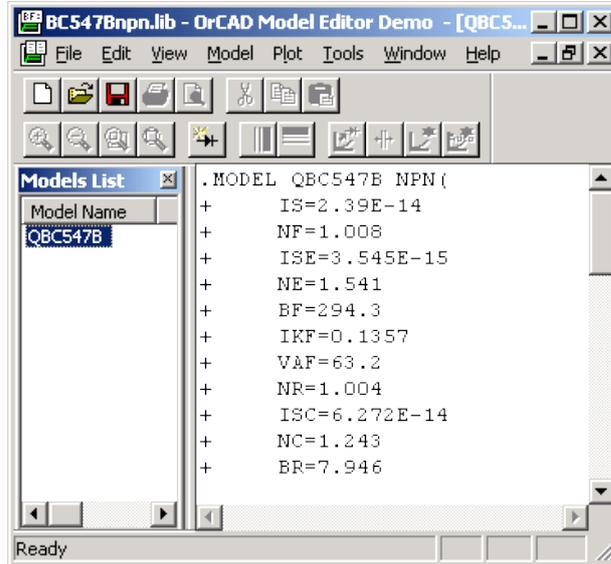


Abbildung 7: geöffnete .lib-Datei im PSpice Model Editor

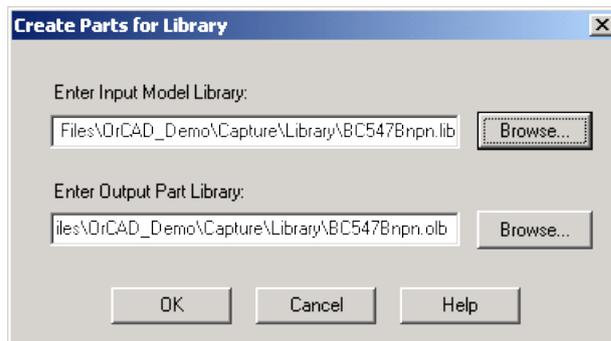


Abbildung 8: Creature Capture Parts...

dann automatisch die untere Zeile (Enter Output Part Library) aus. In der Abbildung 8 kann man auch die fertig ausgefüllten Zeilen sehen. Mit OK bestätigt man den Vorgang.

Im nächsten Schritt öffnet sich ein Ergebnisfenster, welches anzeigt, ob bei der Erstellung der BC547B.olb irgendwelche Fehler aufgetreten sind. Dieses Ergebnisfenster ist in Abbildung 9 zu sehen. Falls keine Fehler aufgetreten sind kann man den Vorgang mit OK beenden. Wenn Fehler aufgetreten sind, sollte man die Prozedur wiederholen. Wenn das auch nicht funktioniert, muss man sich andere Bauteilinformationen suchen.

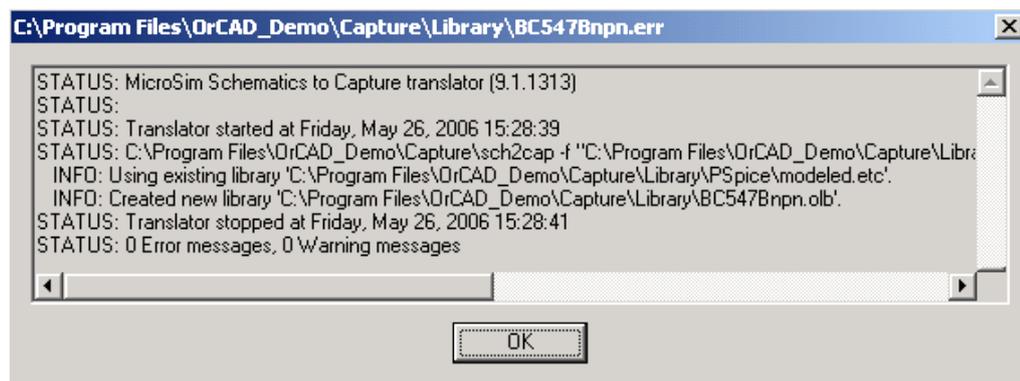


Abbildung 9: Erstellung der .olb-Datei - Ergebnisfenster

3.4 Einfügen der Bauteile in PSpice

Da wir jetzt die Model-Bibliothek und auch die Bauteil-Bibliothek erstellt haben, müssen wir nur noch unser neu erstelltes Bauelement in PSpice einfügen.

3.4.1 Bauelemente in Schaltplan einfügen

Das Einfügen des neuen Bauelements wird ebenfalls über die Schaltfläche Place Part gemacht. Wenn man darauf klickt öffnet sich ein neues Fenster, in welchem man nun auf Add Libraries klicken muss. In dem sich nun öffnenden Fenster kann man die Bibliothek BC547B.olb auswählen und hinzufügen. Anschließend kann man auch Bauelemente aus dieser Bibliothek auswählen.

Wenn man das Bauteil ausgewählt hat, kann man es wie gewohnt im Schaltplan platzieren. Der fertige Schaltplan mit dem neu eingefügten Bauteil BC547B ist in Abbildung 10 zu sehen.

3.4.2 Bauelemente in Simulationseinstellungen einfügen

Wenn man das Bauelement in dem Schaltplan eingefügt hat, muss man außerdem in den Simulationseinstellungen festlegen, welche Bibliotheken verwendet werden.

Zu diesem Zweck öffnet man die Simulationseinstellungen. Wenn man noch keine Simulation gemacht hat, dann wählt man über das Menü den Punkt PSpice → New Simulation. Wenn man bereits eine Simulation gemacht hat, dann wählt man den Punkt PSpice → Edit Simulation. Auf jeden Fall kommt man zu einem Fenster, welches aussieht wie in Abbildung 11. In diesem Fenster gibt es einen Reiter, der Libraries heißt. Wenn man diesen Reiter auswählt, dann kann man

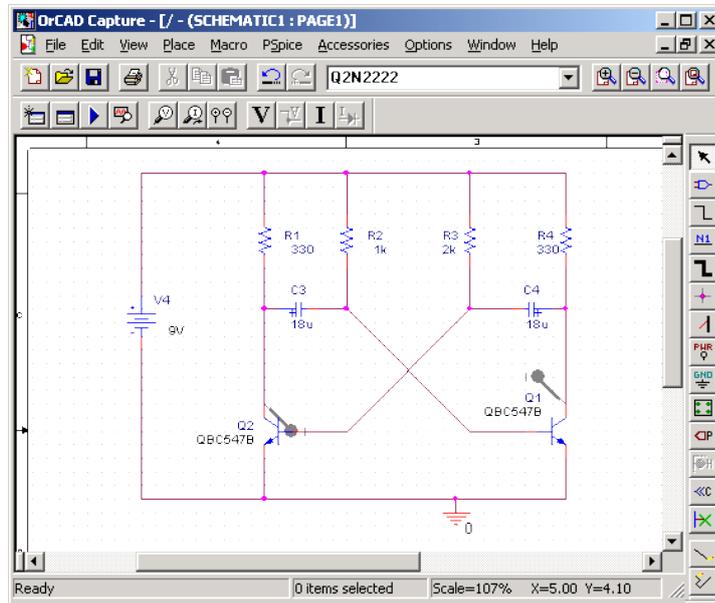


Abbildung 10: eingefügtes Bauteil in Schematic

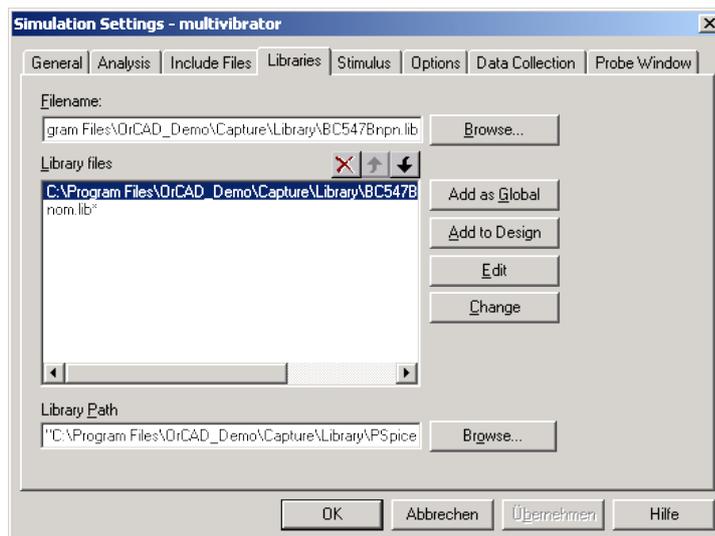


Abbildung 11: Simulationseinstellungen

dort unter Filename den Pfad angeben, in dem die BC547.lib abgespeichert ist. Wenn man den genauen Pfad nicht kennt, kann man über Browse nach der Datei suchen. Nachdem man die Datei gefunden hat, muss man sie auswählen und als globale Bibliothek definieren. Das macht man, in dem man einfach auf den Button Add as Global klickt. Die Datei wird automatisch in das untere Fenster übernommen.

Jetzt müssen eventuell noch die restlichen Simulationseinstellungen vorgenommen werden. Danach kann man die Einstellungen mit OK speichern.

Wenn man jetzt simuliert, dann wird als Ergebnis die Abbildung 12 ausgegeben.

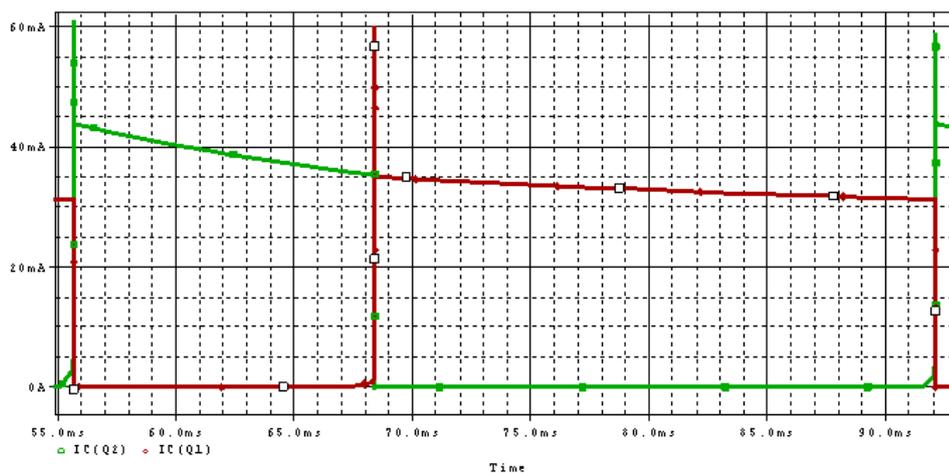


Abbildung 12: Simulationsergebnis