

# Entwurf und Synthese von Eingebetteten Systemen

Sommersemester 2017

## Übungsblatt 1

Mathematisch-Naturwissenschaftliche Fakultät  
Fachbereich Informatik  
Lehrstuhl Eingebettete Systeme  
Prof. Dr. O. Bringmann

EBERHARD KARLS  
UNIVERSITÄT  
TÜBINGEN



---

Montag, den 8. Mai 2017, 12:00

Die Abgabe der Lösungen erfolgt über das ILIAS-System als gepacktes Archiv (ZIP/TAR.GZ).  
Dieses ZIP enthält die schriftlichen Antworten auf Fragen, die Quellcodes in Dateiform (nicht als Listing im PDF),  
und die Simulationsergebnisse als VCD-Dateien.

**Abgabefrist:** Montag, den 15. Mai 2017, 12:00 Uhr

### Aufgabe 1: Entwurfsmethoden von Eingebetteten Systemen

[10 Punkte]

Bitte beantworten sie nachfolgende Fragen kurz!

- (a) Was ist ein Floorplan und welcher Achse im Y-Diagramm ist er zuzuordnen?
- (b) Was ist eine Netzliste und welcher Achse im Gajski-Kuhn-Diagramm ist sie zuzuordnen?
- (c) Auf welcher Ebene im Y-Diagramm werden unter anderem Register, Addierer und Multiplexer zur Schaltungs-  
darstellung verwendet?
- (d) Durch welchen Prozess können sie eine Verhaltensbeschreibung auf algorithmischer Ebene (z.B. C-Code)  
in eine Strukturbeschreibung auf RT-Ebene abbilden? Auf welche Weise kann die RTL-Beschreibung an-  
schließend von der Struktur- auf die Verhaltenssicht abgebildet werden?
- (e) Was ist eine Intellectual Property (IP)? Welche Arten von IPs gibt es und wie unterscheiden sich diese? Bei  
welcher Chip-Entwicklungsmethodik kommen IPs sehr häufig zum Einsatz?
- (f) Nennen sie drei verschiedene Hardwarearchitekturen für den plattformbasierten Entwurf!

## Aufgabe 2: Simulation von 1-bit-Addierern mit VHDL und Questasim

[10 Punkte]

Nachfolgende Aufgaben erfordern die Ausführung von MentorGraphics Questa Sim. Bitte arbeiten sie sich sorgfältig durch den Anhang dieses Übungsblatts, wo sowohl das Einloggen auf den Übungsrechnern als auch die Simulation von Schaltungen kurz erläutert wird.

- (a) Gegeben sei der folgende VHDL-Code für einen Ein-Bit-Halbaddierer auf Logikebene:

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.all;
3  USE IEEE.std_logic_arith.all;
4
5  ENTITY halfadder IS
6  PORT ( A, B : IN  BIT;
7        S, C : OUT BIT);
8  END halfadder;
9
10 ARCHITECTURE behave OF halfadder IS
11 BEGIN
12     S <= A XOR B;
13     C <= A AND B;
14 END;
```

Speichern sie den Code in einer Datei `halfadder.vhd`.

Simulieren sie den Ein-Bit-Halbaddierer vollständig und manuell über die Kommandozeile von Questasim mit den Befehlen: `force <signal> <value>` und `run 10`

Geben sie ihre Simulationsergebnisse dabei als VCD (Value Change Dump)-Datei ab. Eine Anleitung zum Generieren von Value Change Dumps finden sie im Anhang dieses Übungsblatts.

**Erwartete Abgaben:** VCD-Datei

- (b) Erstellen sie in VHDL die Verhaltensbeschreibung eines Ein-Bit-Volladdierers mit Hilfe des Ein-Bit-Halbaddierers aus Aufgabe 1.

**Erwartete Abgaben:** VHDL-Datei

- (c) Simulieren sie den Ein-Bit-Volladdierer vollständig und manuell über die Kommandozeile von Questasim wie in Aufgabe 1. Geben sie ihre Simulationsergebnisse dabei als VCD (Value Change Dump)-Datei ab.

**Erwartete Abgaben:** VCD-Datei

## Questasim auf den Übungsrechnern des Lehrstuhls verwenden

Die Übungen mit Questasim können sie einerseits auf ihrem privaten Rechner durchführen – hierfür steht Ihnen beispielsweise Studenten-Edition zur Verfügung. Andererseits stellen wir Ihnen für die Übungen auch unsere Übungsrechner inkl. vollwertiger Lizenzen für Questasim zur Verfügung, die sie über eine SSH-Verbindung von jedem beliebigen Rechner verwenden können.

Nachfolgend eine Liste der Übungsrechner, die Sie hierfür verwenden können:

- bolero.informatik.uni-tuebingen.de
- bump.informatik.uni-tuebingen.de
- habanera.informatik.uni-tuebingen.de
- mambo.informatik.uni-tuebingen.de
- yabara.informatik.uni-tuebingen.de
- vegeta.informatik.uni-tuebingen.de

Wir stellen Ihnen nachfolgend für die verbreitetsten Betriebssysteme Lösungen vor, wie Sie unter Zuhilfenahme der Übungsrechner Questasim ausführen können. Hierfür wird der X-Forwarding-Mechanismus von SSH und dem X-Server ausgenutzt, um das Questasim-Fenster auf ihrem Rechner darzustellen. Die tatsächliche Ausführung findet allerdings auf dem Übungsrechner statt. Sämtliche Pfadangaben im gestarteten Questasim beziehen sich dementsprechend auf den Übungsrechner und nicht auf Ihren Rechner, auf dem Sie die SSH-Verbindung gestartet haben.

**Um unabhängig vom verwendeten Übungsrechner auf Ihre Daten zugreifen zu können, speichern Sie alle Daten in ihrem Home-Verzeichnis (/home/nutzername) bzw. einem beliebigen Unterverzeichnis darunter!**

# Starten von Questasim über eine SSH-Verbindung unter Linux und Mac OS X

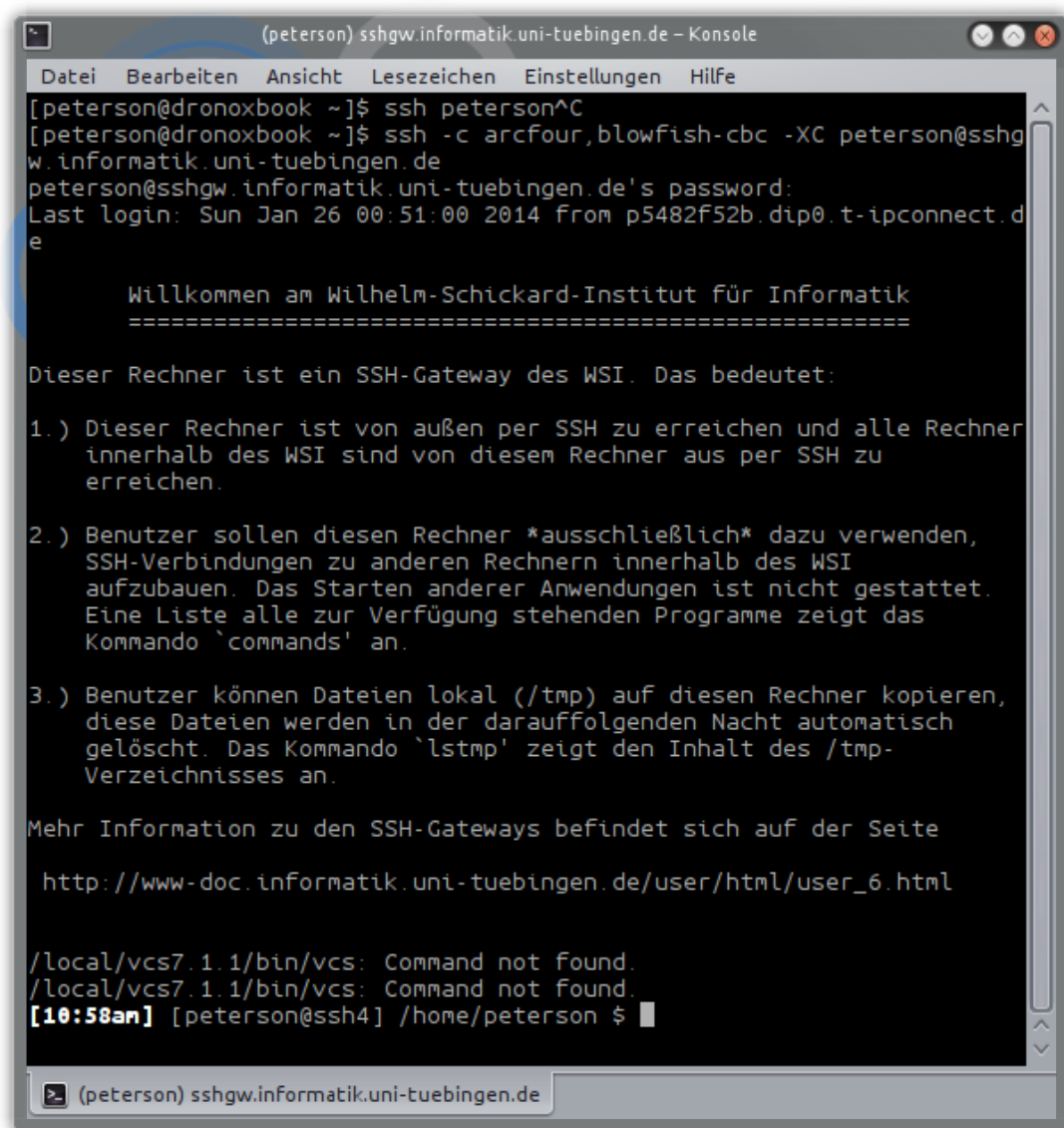
Dieser Abschnitt beschäftigt sich mit dem Starten von Questasim von einem Linux-Rechner aus. Sie benötigen hierfür eine grafische Installation mit X-Server und OpenSSH.

## 1. Mit dem SSH-Gateway verbinden

Im ersten Schritt müssen Sie sich mit dem SSH-Gateway unseres Fachbereichs verbinden – dieses erreichen Sie unter der Adresse `sshgw.informatik.uni-tuebingen.de`. Geben Sie dazu in eine Linux-Konsole ihrer Wahl (z.B. `xterm`, `konsole`, `gterminal`) folgendes ein:

```
# ssh -c arcfour,blowfish-cbc -XC username@sshgw.informatik.uni-tuebingen.de
```

Geben Sie anschließend ihr WSI-Passwort ein (haben Sie initial über den Antrag festgelegt). Sie sollten nun eine Ausgabe wie die folgende sehen.



```
(peterson) sshgw.informatik.uni-tuebingen.de - Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
[peterson@dronoxbook ~]$ ssh peterson^C
[peterson@dronoxbook ~]$ ssh -c arcfour,blowfish-cbc -XC peterson@sshgw.informatik.uni-tuebingen.de
peterson@sshgw.informatik.uni-tuebingen.de's password:
Last login: Sun Jan 26 00:51:00 2014 from p5482f52b.dip0.t-ipconnect.de

Willkommen am Wilhelm-Schickard-Institut für Informatik
=====

Dieser Rechner ist ein SSH-Gateway des WSI. Das bedeutet:

1.) Dieser Rechner ist von außen per SSH zu erreichen und alle Rechner innerhalb des WSI sind von diesem Rechner aus per SSH zu erreichen.

2.) Benutzer sollen diesen Rechner *ausschließlich* dazu verwenden, SSH-Verbindungen zu anderen Rechnern innerhalb des WSI aufzubauen. Das Starten anderer Anwendungen ist nicht gestattet. Eine Liste aller zur Verfügung stehenden Programme zeigt das Kommando 'commands' an.

3.) Benutzer können Dateien lokal (/tmp) auf diesen Rechner kopieren, diese Dateien werden in der darauffolgenden Nacht automatisch gelöscht. Das Kommando 'lstmp' zeigt den Inhalt des /tmp-Verzeichnisses an.

Mehr Information zu den SSH-Gateways befindet sich auf der Seite
http://www-doc.informatik.uni-tuebingen.de/user/html/user_6.html

/local/vcs7.1.1/bin/vcs: Command not found.
/local/vcs7.1.1/bin/vcs: Command not found.
[10:58am] [peterson@ssh4] /home/peterson $
```

Figure 1: Ausgabe nach erfolgreicher Verbindung zum SSH-Gateway

Die ersten Argumente des Befehls sorgen dabei für eine bessere Performance durch eine schnellere Verschlüsselung/Entschlüsselung (`-c arcfour,blowfish-cbc`) und einen komprimierten Datenstrom (`-C`). Die Option `-X` aktiviert den X-Forwarding-Mechanismus.

Ersetzen Sie bitte `username` durch Ihren WSI-Account!

## 2. Mit dem Übungsrechner verbinden

Suchen Sie sich im zweiten Schritt einen Übungsrechner aus der obigen Liste (z.B. samba) heraus und loggen Sie sich vom SSH-Gateway aus auf diesem Rechner ein. Geben Sie hierfür folgendes in der Konsole ein:

```
# ssh -c arcfour,blowfish-cbc -XC username@uebungsrechner
```

Das Feld `username` entspricht dabei noch immer ihrem WSI-Account und `uebungsrechner` dem entsprechenden Hostnamen des Übungsrechners. Ein Beispiel für diese Felder wären folgende:

```
# ssh -c arcfour,blowfish-cbc -XC peterson@samba
```

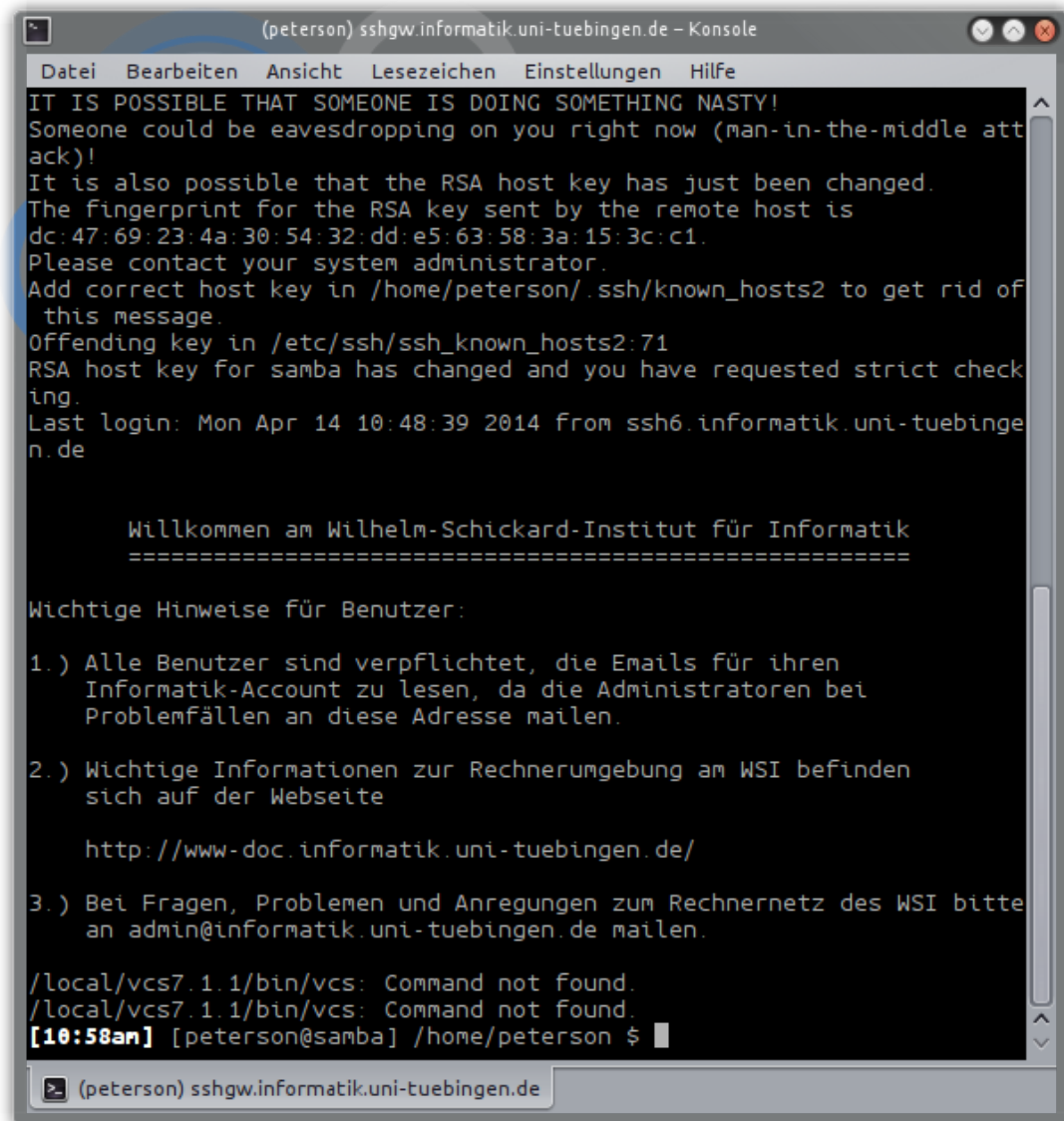


Figure 2: Ausgabe nach erfolgreicher Verbindung zum Übungsrechner (hier: samba)

### 3. Questasim starten

Im ersten Schritt müssen Sie die Umgebungsvariablen setzen. Dies können Sie auch entweder von einem Skript erledigen lassen oder in ihr Shell-Startup-File ( `/.cshrc`) kopieren.

```
# setenv LM\_LICENSE\_FILE 1701@menelaos
# setenv QUESTA "/afs/wsi/es/tools/mentor/questa\_sv\_afv/10.4c\_5/questasim"
# set path = ($path $QUESTA/bin )
```

Starten Sie anschließend Questasim über den Befehl:

```
# vsim
```

Es sollte sich nun folgendes Fenster öffnen:

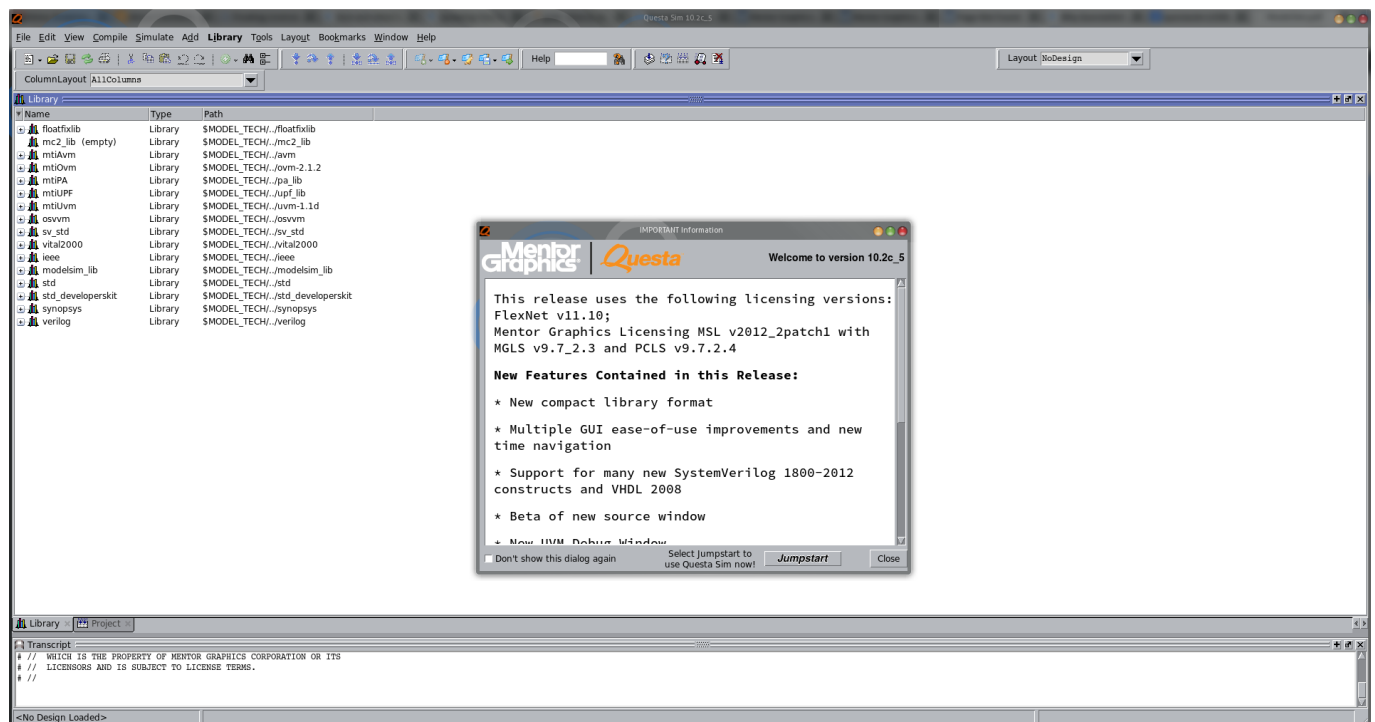


Figure 3: Questasim nach dem Start

# Starten von Questasim über eine SSH-Verbindung unter Windows

Dieser Abschnitt beschäftigt sich mit dem Starten von Questasim von einem Windows-Rechner aus. Sie benötigen hierfür einen X11-Client – in den nachfolgenden Schritten verwenden wir hierfür MobaXTerm Home Edition. **Hinweis:** Die Screenshots beziehen sich noch auf das Starten der alten Version – bitte für die Konsoleneingaben nur den Text beachten, nicht die Screenshots.

## 1. Installation und Start von MobaXTerm

Laden Sie sich die Home Edition von MobaXTerm unter dem obigen Link gratis herunter und führen Sie den Installer aus. Starten Sie anschließend MobaXTerm. Sie sollten folgendes Fenster sehen:

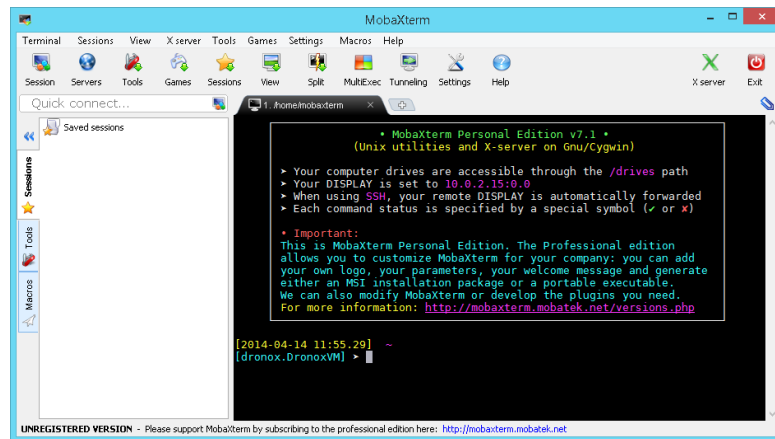


Figure 4: MobaXTerm nach dem Start

## 2. Konfiguration der SSH-Verbindung

Richten Sie nun die SSH-Verbindung zum Übungsrechner ein. Klicken Sie hierfür links oben auf `Session`. Es öffnet sich das nachfolgende Fenster:

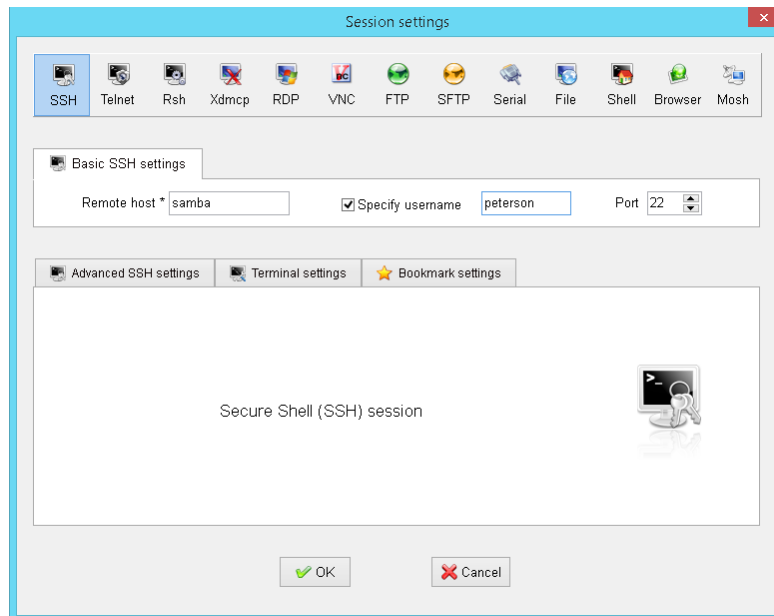


Figure 5: MobaXTerm: SSH-Verbindung konfigurieren (1)

Geben Sie bei `Remote host` den Namen des Übungsrechners aus der obigen Rechnerliste (z.B. `samba`) ein. Vervollständigen Sie auch bei `Specify username` ihren WSI-Account.

Öffnen Sie anschließend die erweiterten Einstellungen durch einen Klick auf `Advanced SSH settings`. Es öffnet sich der folgende Dialog:

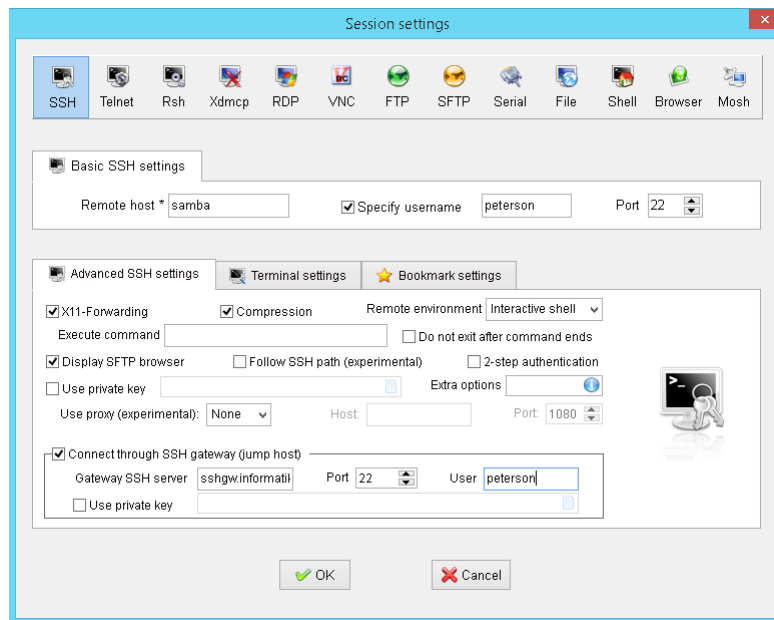


Figure 6: MobaXTerm: SSH-Verbindung konfigurieren (2)

Stellen Sie sicher, dass der Haken bei `X11-Forwarding` und `Compression` aktiviert ist. Aktivieren Sie zudem die Option `Connect through SSH gateway (jump host)`. Tragen Sie bei `Gateway SSH server` das SSH-Gateway (`sshgw.informatik.uni-tuebingen.de`) ein sowie bei `User` ihren WSI-Account.



### 3. Starten der SSH-Verbindung

Bestätigen Sie die SSH-Session anschließend mit einem Klick auf OK.

Geben Sie in dem sich öffnenden Passwort-Dialog ihr WSI-Kennwort ein.

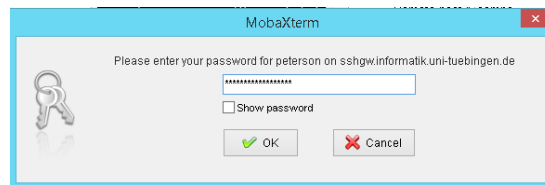


Figure 7: MobaXTerm: SSH-Verbindung starten (1)

Bestätigen Sie solange die Dialoge bis sich folgendes Fenster öffnet und sie vom Rechner samba nochmals nach dem Kennwort gefragt werden.

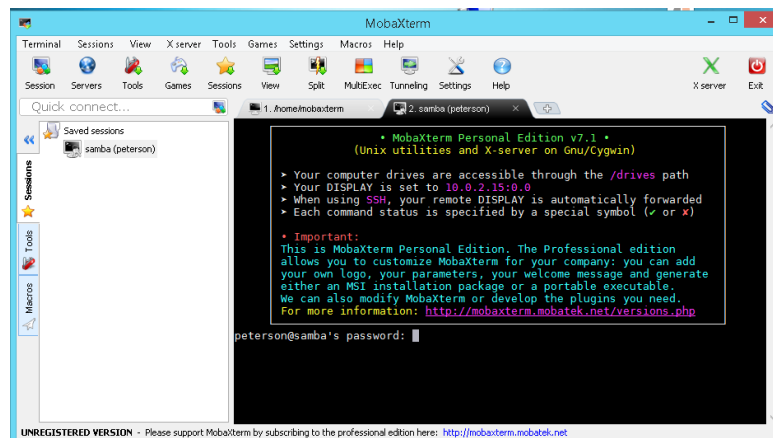


Figure 8: MobaXTerm: SSH-Verbindung starten (2)

Sie sollten nun eingeloggt sein.

#### 4. Starten von Questasim

Zum Starten von Questasim müssen Sie zuerst die Umgebungsvariablen setzen. Dies können Sie auch entweder von einem Skript erledigen lassen oder in ihr Shell-Startup-File ( `./cshrc`) kopieren.

```
# setenv LM\_LICENSE\_FILE 1701@menelaos
# setenv QUESTA "/afs/wsi/es/tools/mentor/questa\_sv\_afv/10.4c\_5/questasim"
# set path = ($path $QUESTA/bin )
```

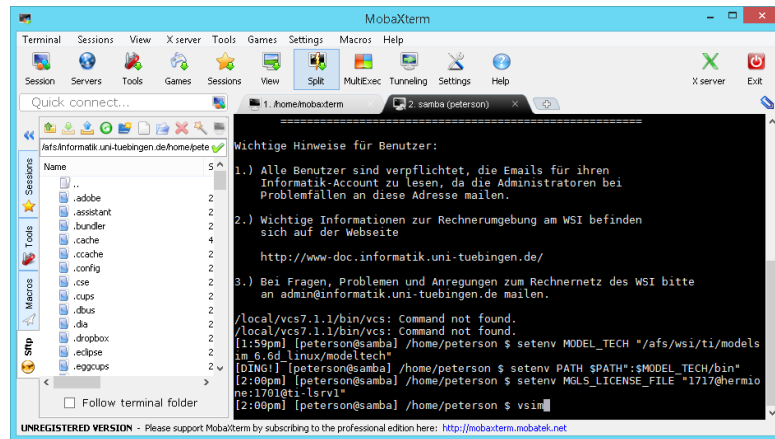


Figure 9: MobaXTerm: Questasim starten (1)

Starten Sie anschließend Questasim über den Befehl:

```
# vsim
```

Es sollte sich nun folgendes Fenster öffnen:

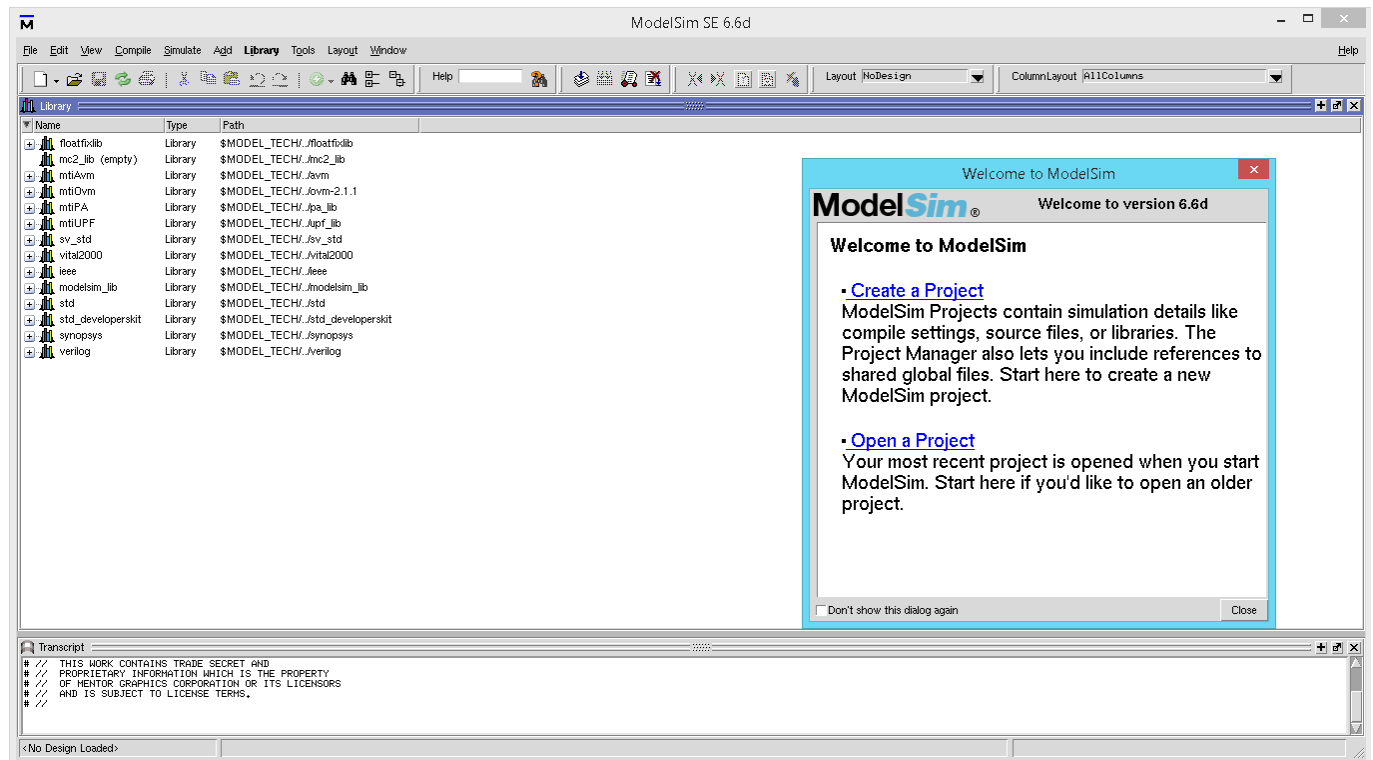
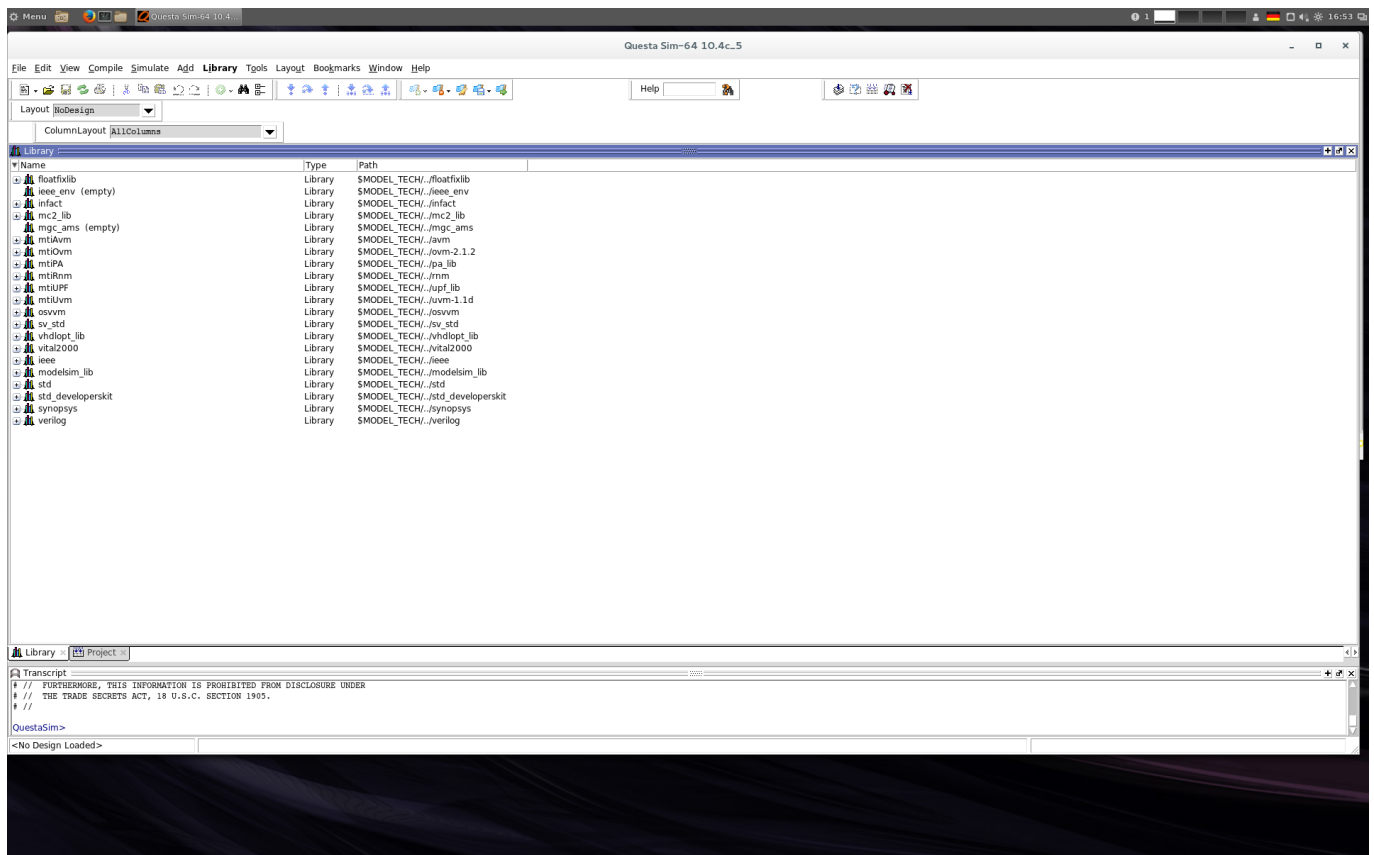


Figure 10: MobaXTerm: Questasim starten (2)

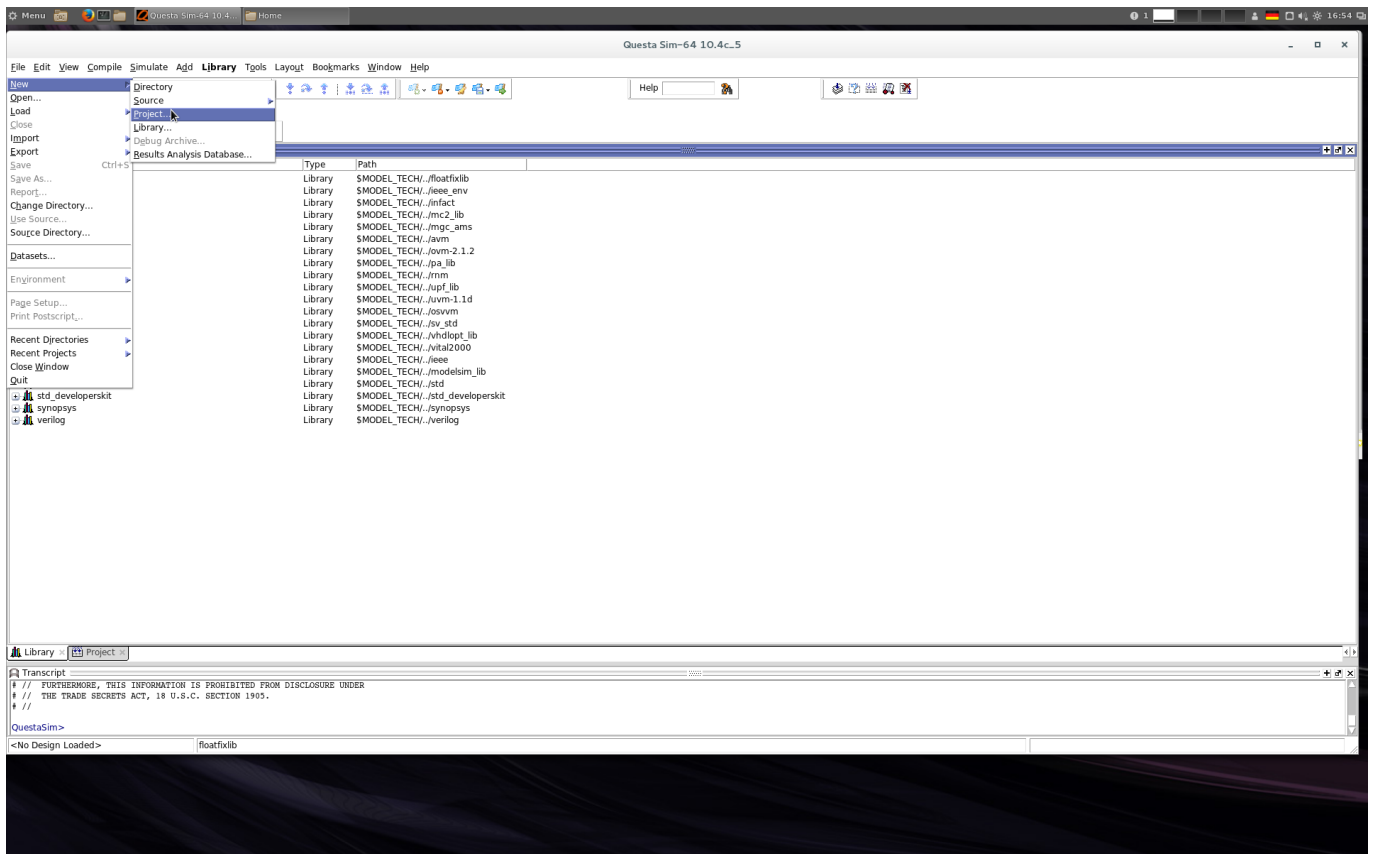
# Simulation mit Questasim

Nachfolgend möchten wir Ihnen eine kompakte Schritt-für-Schritt-Anleitung zur manuellen Simulation eines Halbadierers geben.

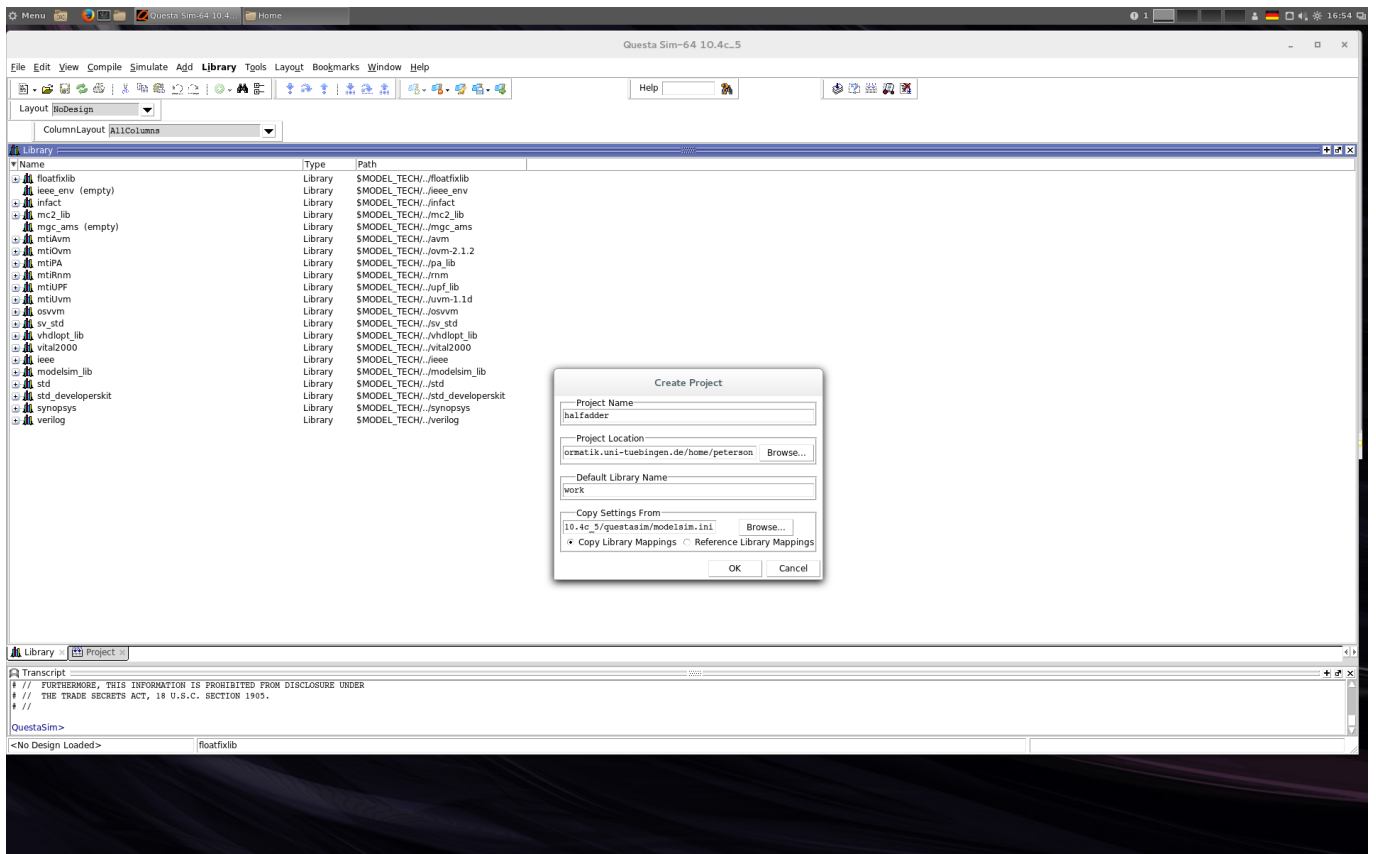
1. Wenn sie Questasim entsprechend des obigen Tutorials gestartet haben sollten sie folgende Oberfläche sehen:



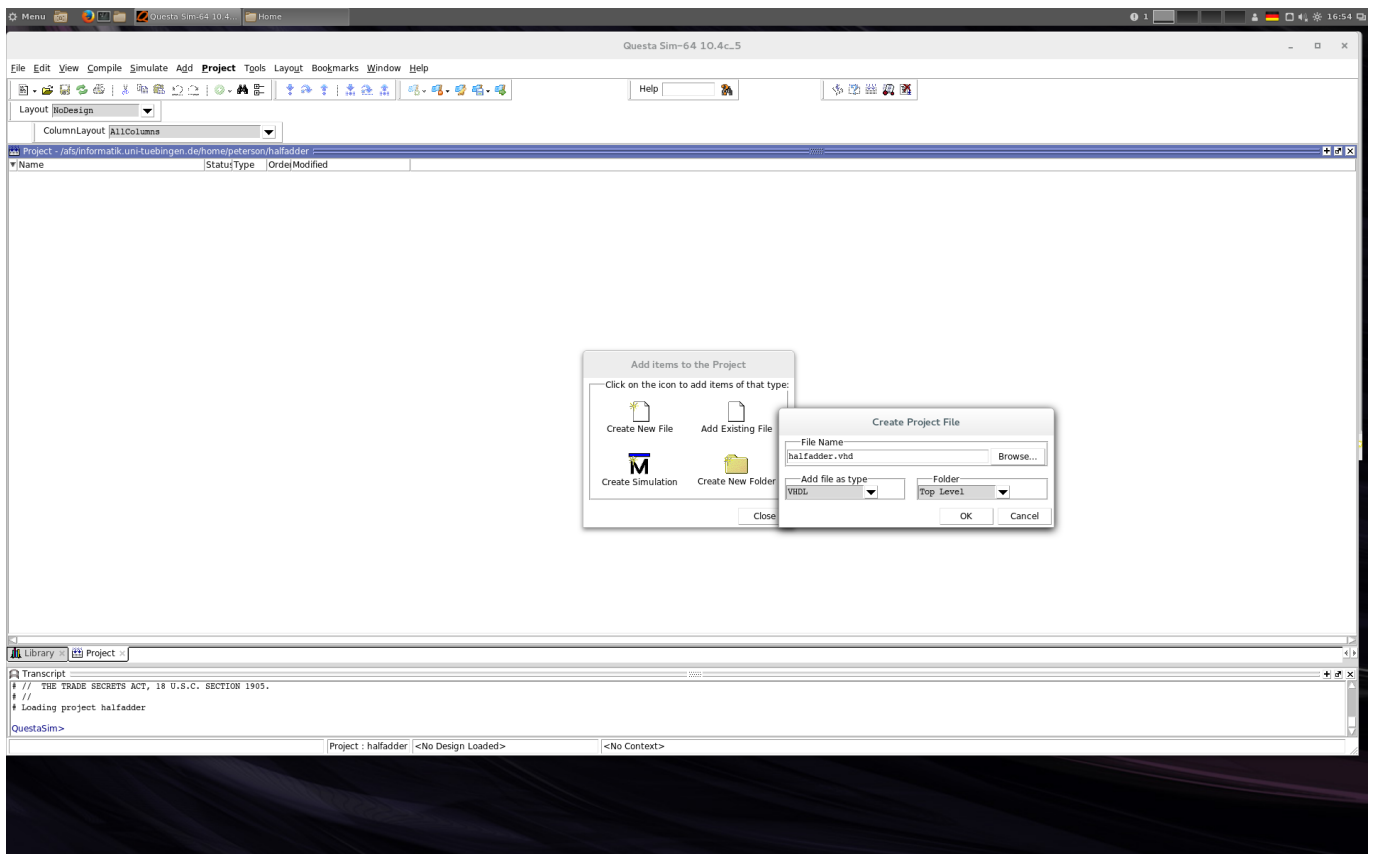
## 2. Erstellen sie nun ein neues Projekt via **File > New > Project**.



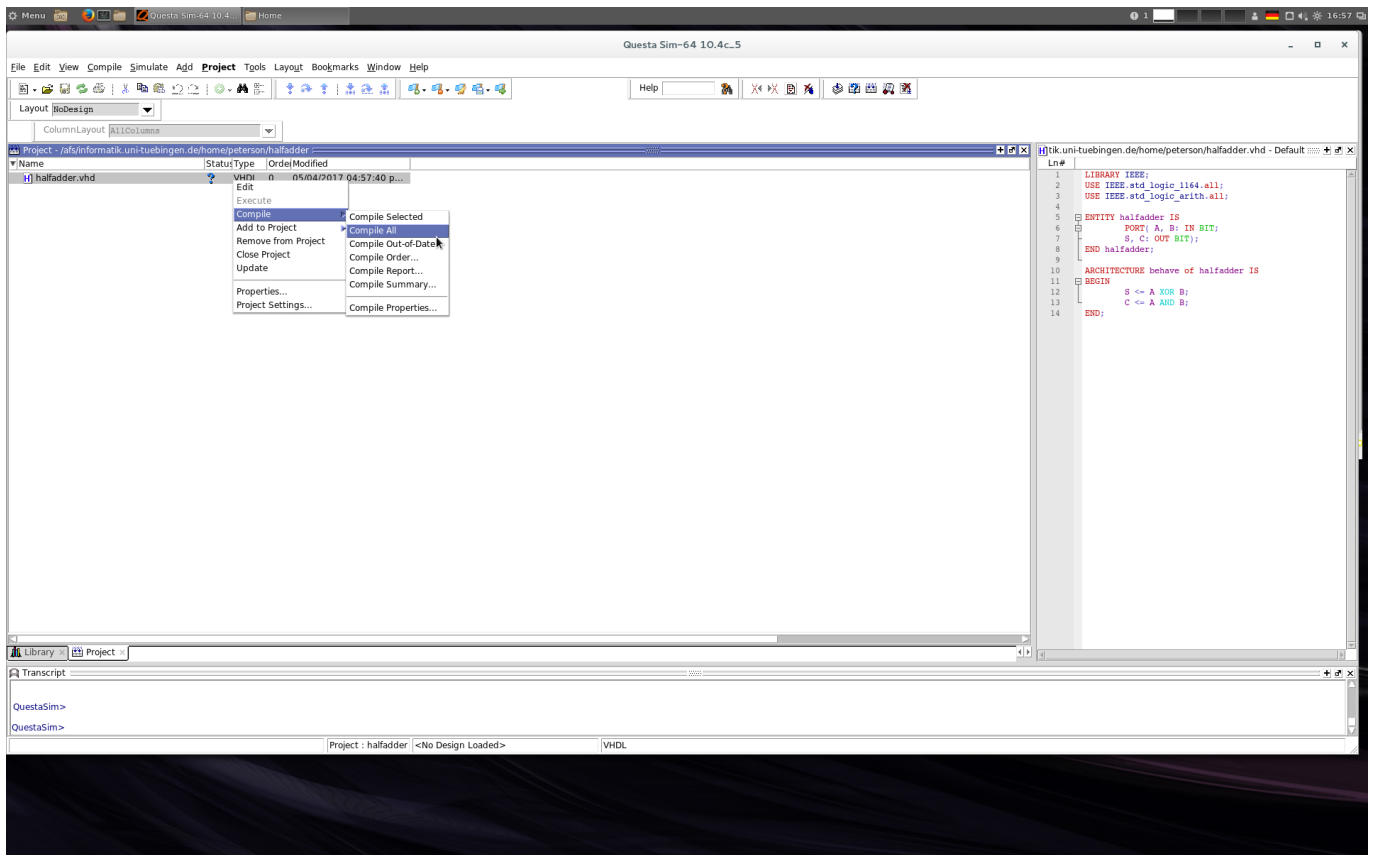
### 3. Geben sie einen sinnvollen Namen und einen Speicherort für das Projekt an.



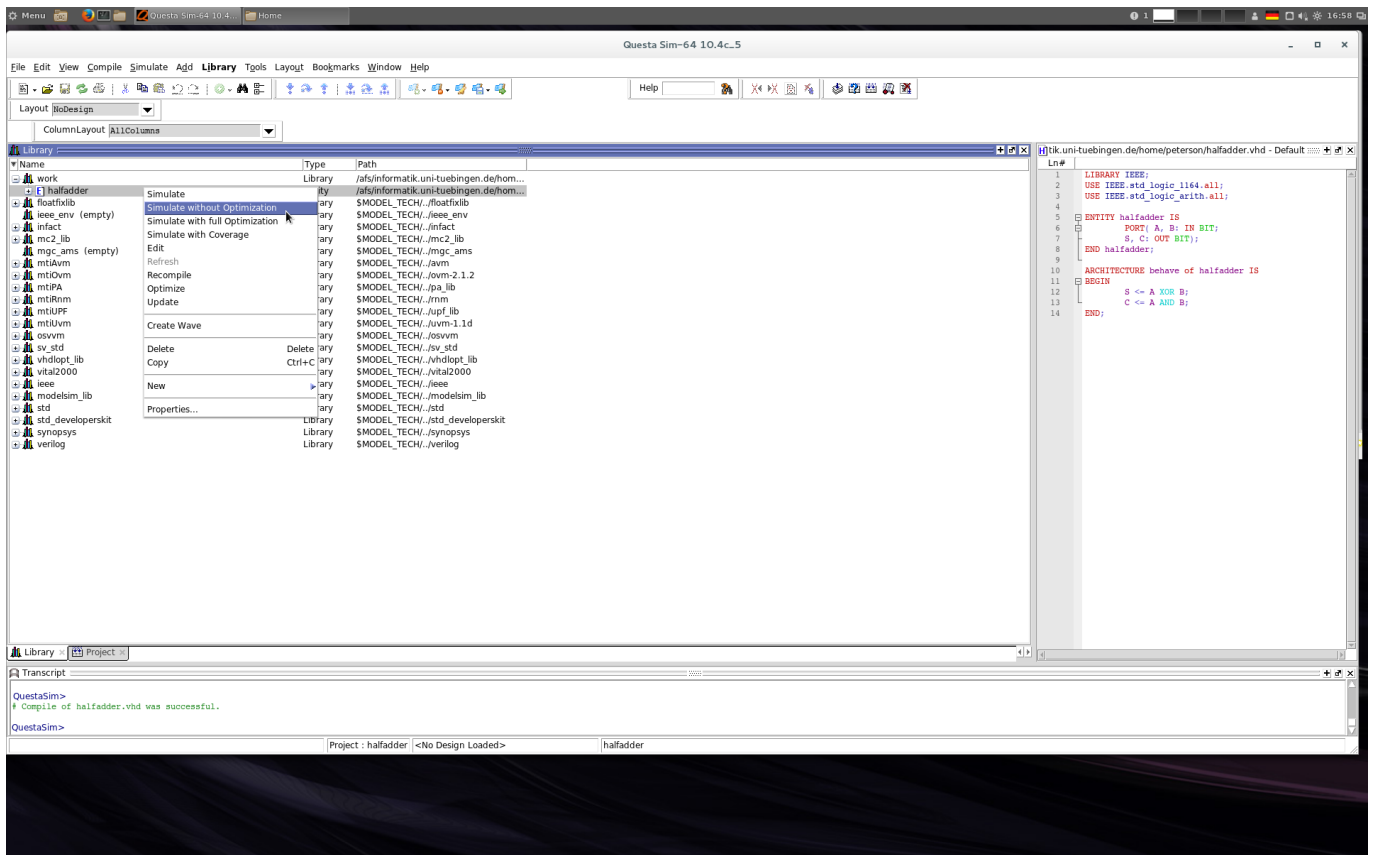
4. Legen sie über das Fenster **Add Items to the Project > Create New File** eine neue Datei mit dem Namen **halfadder.vhd** an.



5. Geben sie im Editorfenster (rechts) den Code für den Halbaddierer ein und speichern sie die Datei. Öffnen sie anschließend das Kontextmenü der Datei mittels Rechtsklick und kompilieren sie die Datei über **Compile > Compile All**. Es sollte im Meldungsfenster (unten) ein ... *was successful* erscheinen.

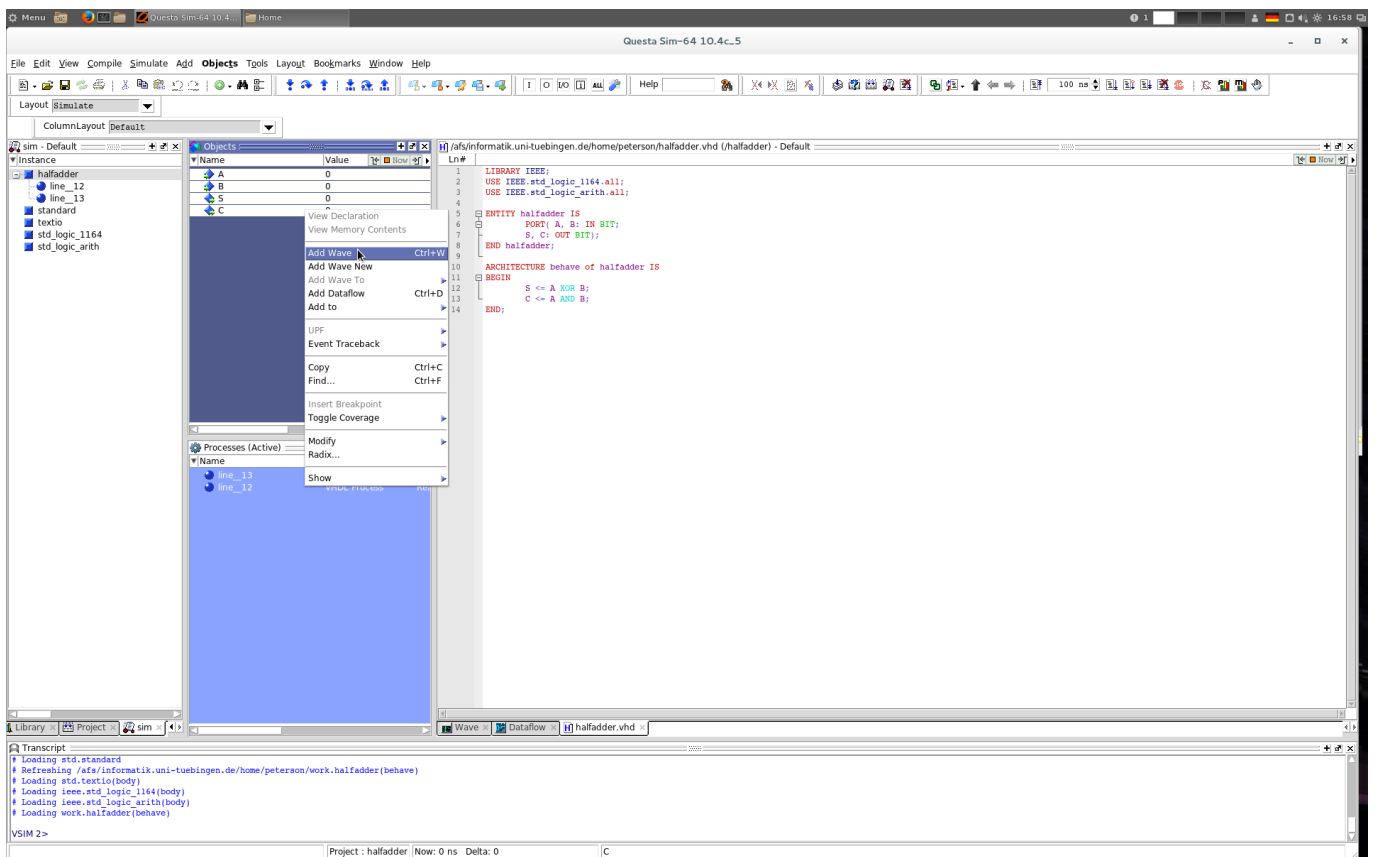


6. Wechseln sie nun den Reiter auf **Library**. In der Library **work** sollten sie eine Komponente **halfadder** finden. Öffnen sie auch hier das Kontextmenü mit der rechten Maustaste und starten sie die Simulation mit Hilfe von **Simulate without Optimization**. Es sollte im Meldungsfenster (unten) eine grüne Erfolgsmeldung erscheinen.

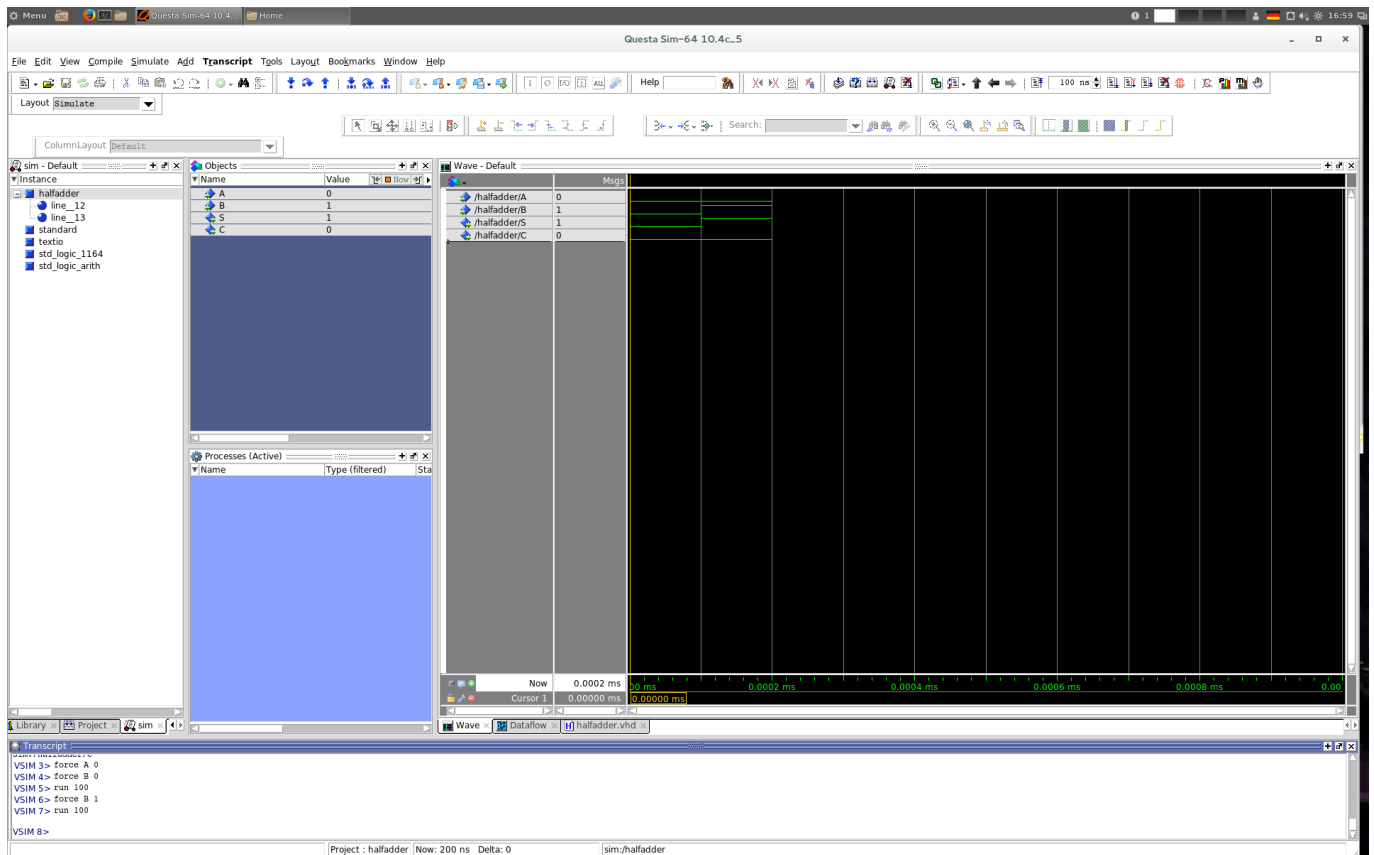




7. Fügen sie jetzt die Signale, die sie grafisch ausgeben lassen wollen während der Simulation mittels Markierung und rechter Maustaste (**Add Wave**) zum Simulationsfenster hinzu.



8. Tippen sie in der Kommandozeile (unten: VSIM x>) die entsprechenden Simulationsbefehle, z.b. `force A 0`, `force B 0`, `run 100` ein. Dies sorgt dafür, dass die Ports A und B auf 0 gesetzt werden und anschließend 100 Zeiteinheiten (per Default: Nanosekunden) simuliert werden. Sie sollten nun die folgende Ausgabe sehen.



## Generierung eines Value Change Dumps (VCD-Datei)

Zur Verifikation Ihrer Simulationsergebnisse benötigen wir den Export der Simulationstraces von Questasim. Hierfür empfehlen wir den Export in das VCD (Value Change Dump)-Format, welches anschließend mit diversen Tools wie z.B. GTKWave betrachtet werden kann.

Um ein VCD-File zu erzeugen, gehen sie in Questasim bitte wie folgt vor:

1. Kompilieren und Laden sie das Design erfolgreich im Transcript Window.
2. Spezifizieren sie über die Simulationskonsole den Namen der zu erzeugenden VCD-Datei  
`vcd file <filename>.vcd`
3. Fügen sie die zu protokollierenden Signale hinzu: `vcd add <path_to_instance>/*`
4. Simulieren sie wie gewohnt Ihre Schaltung.
5. Beenden sie die Simulation: `quit -sim`

Das generierte VCD File können sie sich beispielsweise in GTKWave anschauen:

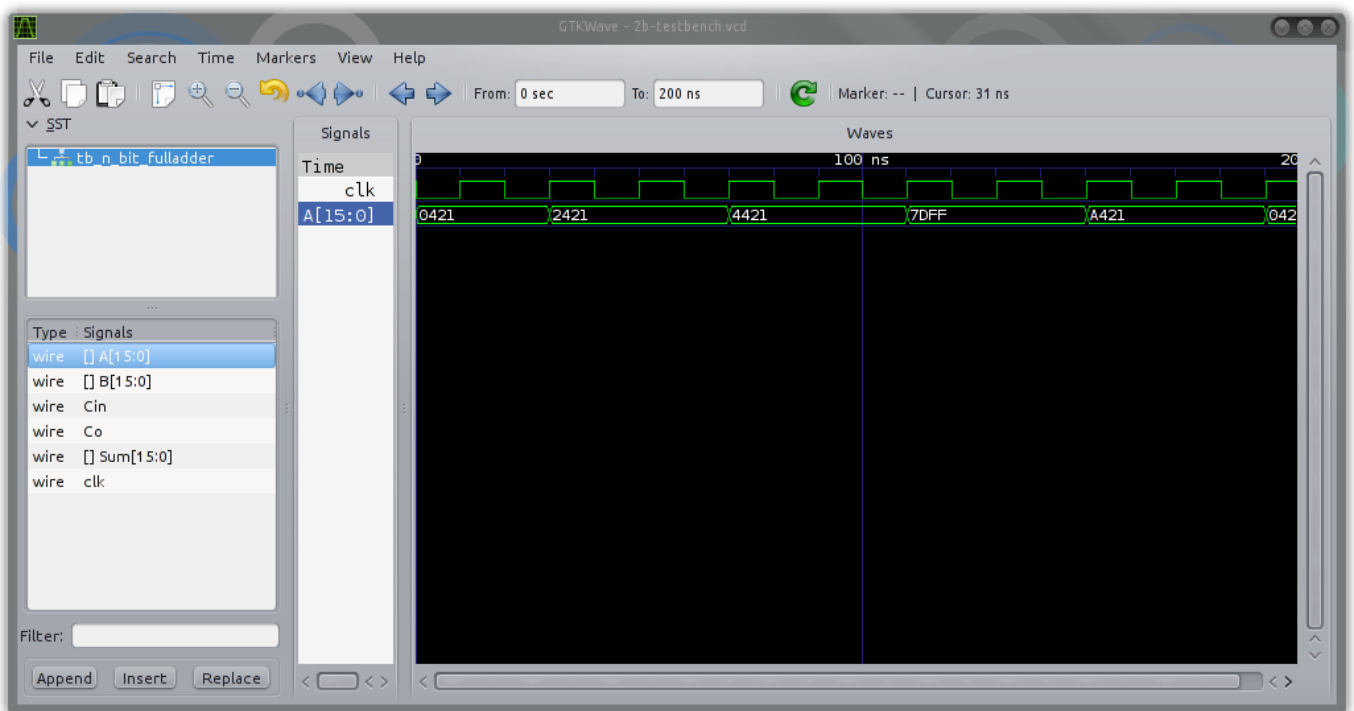


Figure 11: Beispielausgabe von GTKWave