



Entwurf und Synthese von Eingebetteten Systemen

Sommersemester 2021

Übungsblatt 1

Mittwoch, den 18. Mai 2021, 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: Sonntag, den 30. Mai 2021, 23:55 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 ei-
ne Strukturbeschreibung auf RT-Ebene abbilden? Auf welche Weise kann die RTL-Beschreibung anschließend
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 zwei verschiedene Hardware-Architekturen für den plattformbasierten Entwurf!
- (g) Was bestimmt die maximale Taktrate einer digitalen Schaltung?

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

[10 Punkte]

Nachfolgende Aufgaben erfordern die Ausführung von Cadence Xcelium. 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. Finden und korrigieren Sie den Fehler!

```
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 OR B;
13     C <= A AND B;
14 END;
15
```

- (b) Simulieren Sie den korrigierten Ein-Bit-Halbaddierer vollständig und manuell über die Kommandozeile von Xcelium mit den Befehlen: `force <signal> <value>` und `run 10`. Speichern Sie hierzu den Code in einer Datei `halfadder.vhd`. 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

- (c) Wofür wird ein Reset-Signal verwendet? Warum gibt es kein Reset-Signal für den Ein-Bit-Halbaddierer?
- (d) Erstellen Sie in VHDL die Strukturbeschreibung eines Ein-Bit-Volladdierers mit Hilfe des Ein-Bit-Halbaddierers aus Aufgabe 2(b).

Erwartete Abgaben: VHDL-Datei

- (e) Simulieren Sie die Strukturbeschreibung des Ein-Bit-Volladdierers vollständig und manuell über die Kommandozeile von Xcelium wie in Aufgabe 2(b). Geben Sie Ihre Simulationsergebnisse dabei als VCD (Value Change Dump)-Datei ab.

Erwartete Abgaben: VCD-Datei

- (f) Erstellen Sie in VHDL die Verhaltensbeschreibung eines Ein-Bit-Volladdierers.

Erwartete Abgaben: VHDL-Datei

- (g) Simulieren Sie die Verhaltensbeschreibung des Ein-Bit-Volladdierers vollständig und manuell über die Kommandozeile von Xcelium wie in Aufgabe 2(b). Geben Sie Ihre Simulationsergebnisse dabei als VCD (Value Change Dump)-Datei ab.

Erwartete Abgaben: VCD-Datei

- (h) Sind die Simulationsergebnisse der Strukturbeschreibung und der Verhaltensbeschreibung identisch? Wo können sich Unterschiede ergeben?

Xcelium auf den Übungsrechnern des Lehrstuhls verwenden

Wichtige Information:

Bitte beachten Sie, dass sich aktuell unsere interne Infrastruktur ändert, weswegen leider beachtet werden muss, wann Ihr WSI-Account erstellt wurde. Bereits bestehende WSI-Accounts werden im folgenden als *INF-Accounts* bezeichnet, WSI-Accounts, die **dieses Semester** angelegt wurden, werden als *CS-Accounts* bezeichnet. **Wichtig:** INF-Accounts können die Anleitung weiterhin verfolgen. CS-Accounts müssen alle *informatik.uni-tuebingen.de* durch *cs.uni-tuebingen.de* ersetzen außer bei *menelaos*!

Um die für die Übungen notwendigen Tools starten zu können, ist es notwendig, dass Sie sich in unserem Netzwerk einloggen und einen der Poolrechner verwenden, die nachfolgend gelistet sind:

INF-Accounts:

- samba
- goten
- ducky
- dacky
- veku
- yabara
- cvraman
- mambo
- bolero

CS-Accounts:

- bump
- habanera
- veritas
- goku
- vegeta

Auf diesen Rechnern befinden sich im Netzwerkdateisystem AFS sämtliche Toolinstallationen. Der Lizenzserver für alle Tools läuft auf dem Rechner `menelaos.informatik.uni-tuebingen.de` und dem Port 1701 (dieser wird über eine Umgebungsvariable bekannt gemacht, siehe unten).

Wir stellen Ihnen nachfolgend für die verbreitetsten Betriebssysteme Lösungen vor, wie sie sich auf einem der Poolrechner einloggen können und wie Sie unter Zuhilfenahme der Poolrechner Xcelium ausführen können. Wir empfehlen die Nutzung von X2Go, welche Ihnen einen vollwertigen Linux-Desktop zur Verfügung stellt. Sollte es mit X2Go zu Problemen kommen, gibt es auch eine Lösung mit Hilfe des X-Forwarding-Mechanismus von SSH und dem X-Server. Sämtliche Lösungen funktionieren unter den drei großen Betriebssystemen Windows, Mac OS X und Linux.

Hinweise: Sämtliche Pfadangaben im gestarteten Xcelium 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 des Linux-Desktops über eine X2Go-Verbindung

Dieser Abschnitt beschäftigt sich mit dem Starten von Xcelium über X2Go, welches gratis für Windows, Mac OS X und Linux zur Verfügung steht.

1. Installieren Sie den X2Go-Client

Laden Sie sich X2Go für Ihr Betriebssystem herunter (<https://wiki.x2go.org/doku.php>) und installieren Sie es.

2. Starten Sie X2Go

Starten Sie den X2Go-Client über das Systemmenü.

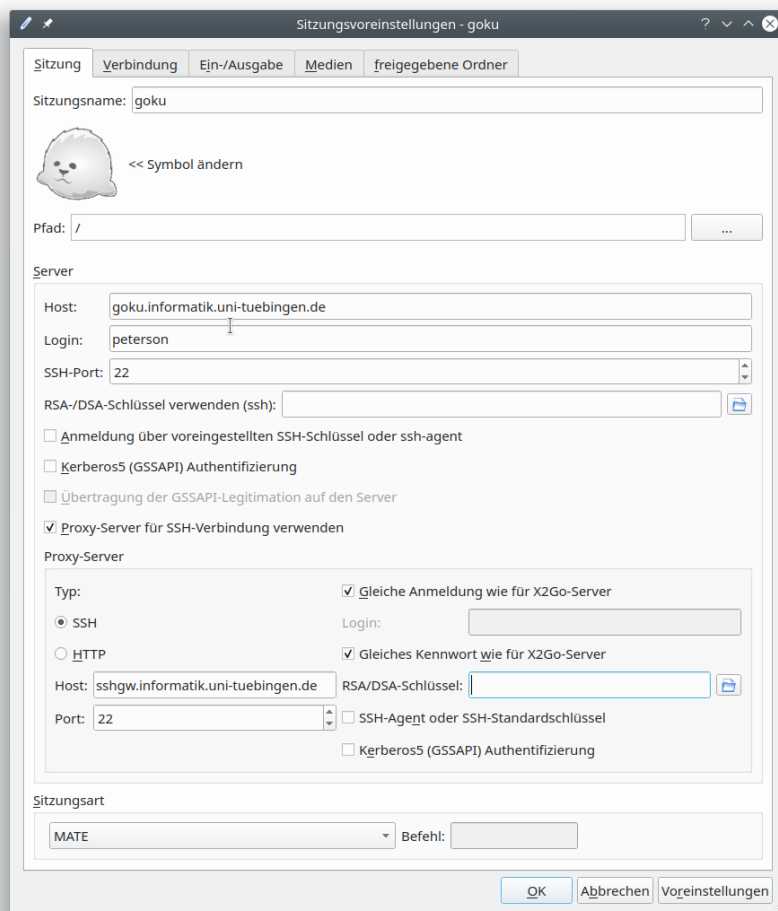
3. Legen Sie eine neue Sitzungskonfiguration an

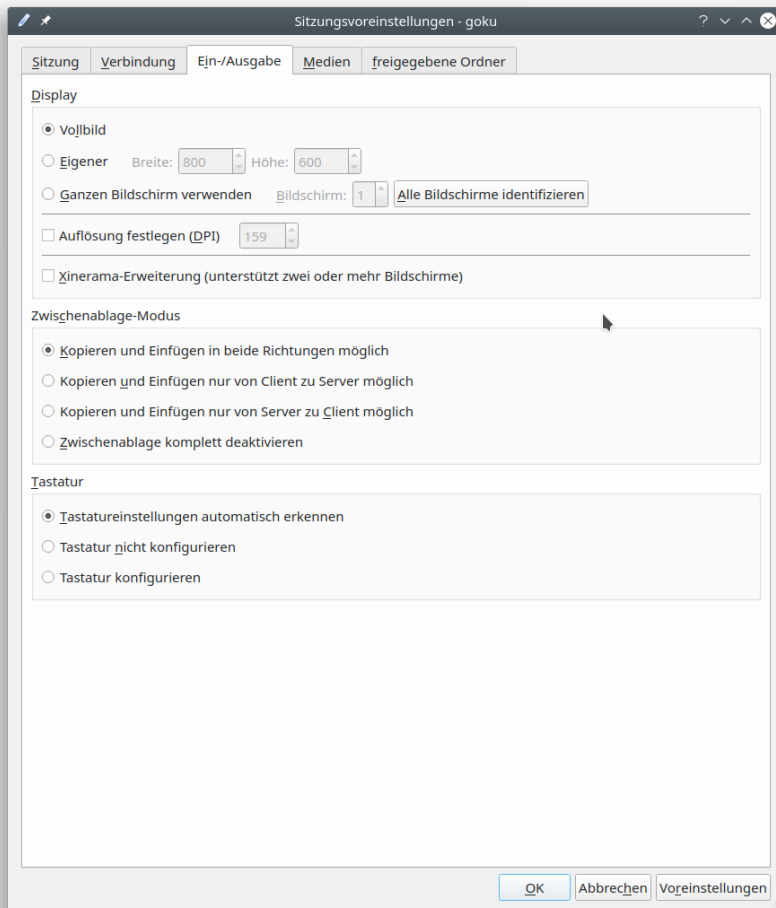
Legen Sie nun über das Menü **Sitzung** → **Neue Sitzung...** eine neue Sitzungskonfiguration an. Geben Sie die Informationen wie im nachfolgenden Screenshot an und ersetzen Sie **peterson** durch Ihren eigenen WSI-Nutzernamen, **goku.informatik.uni-tuebingen.de** durch den Rechner Ihrer Wahl und den Sitzungsnamen durch einen von Ihnen gewählten Sitzungsnamen.

Hinweis: Sollten Sie Probleme mit der Verbindung bekommen, probieren Sie:

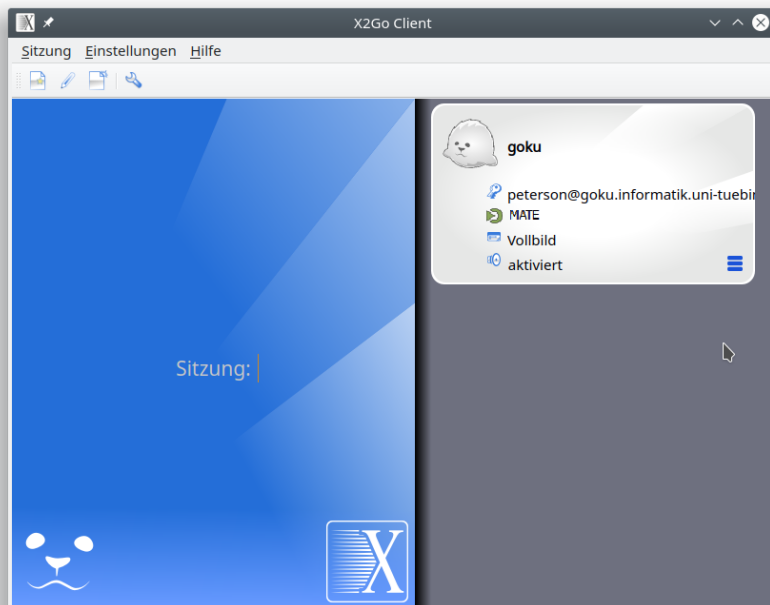
- Die Auswahl eines anderen Zielrechners, siehe obige Liste.
- Das load-balanced SSH-Gateway `sshgw.informatik.uni-tuebingen.de` durch einen der dedizierten Server zu tauschen (und auch dort verschiedene zu testen): `ssh1.informatik.uni-tuebingen.de`, `ssh2.informatik.uni-tuebingen.de`, `ssh3.informatik.uni-tuebingen.de`, `ssh4.informatik.uni-tuebingen.de`, `ssh5.informatik.uni-tuebingen.de`

Wenn sie trotzdem keine stabile Verbindung aufbauen können, wählen Sie den Alternativweg über X-Forwarding.



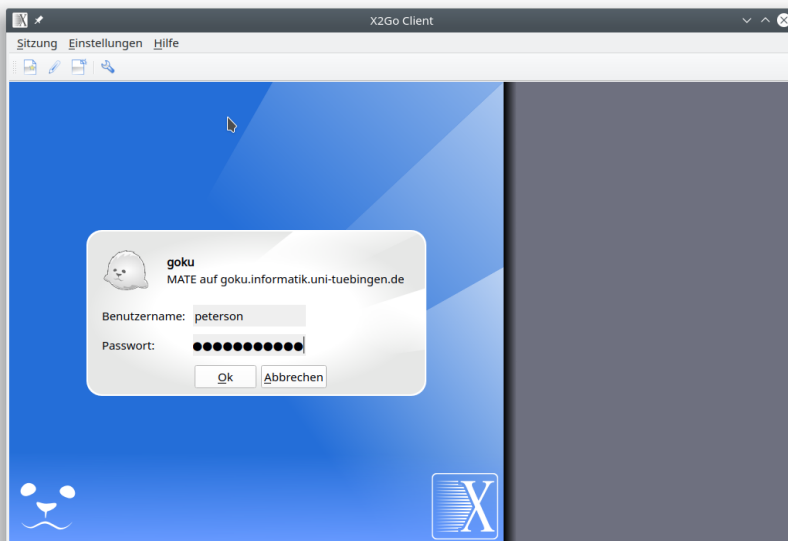


Sie sollten nun Ihre neue Konfiguration sehen:

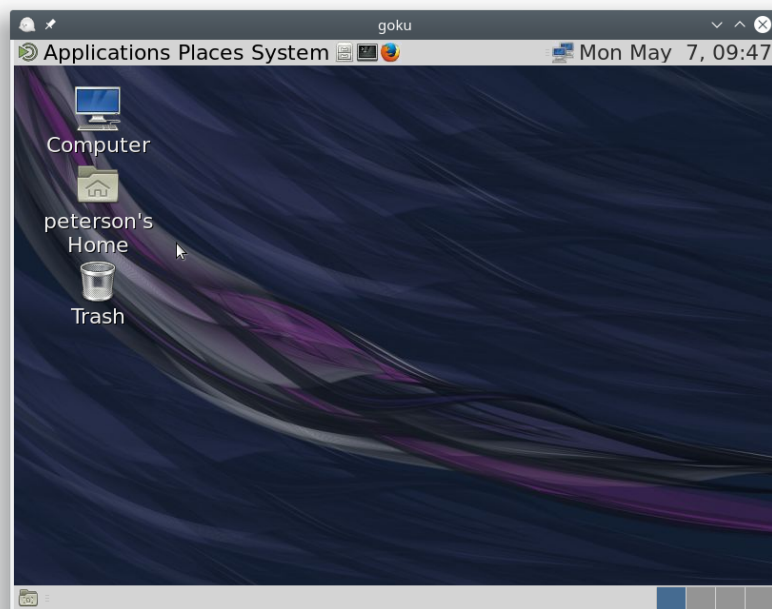


4. Loggen Sie sich auf dem Rechner ein

Starten Sie nun die neue Sitzung durch einen Klick auf die Sitzung oder Eingabe des Sitzungsnamens in der Eingabemaske. Sie sollten nun einen Loginbildschirm sehen:



Wenn alles erfolgreich geklappt hat, dann sehen Sie nun den Desktop – herzlichen Glückwunsch!



Starten einer SSH-Verbindung unter Linux und Mac OS X

Dieser Abschnitt beschäftigt sich mit dem Starten von Xcelium 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.

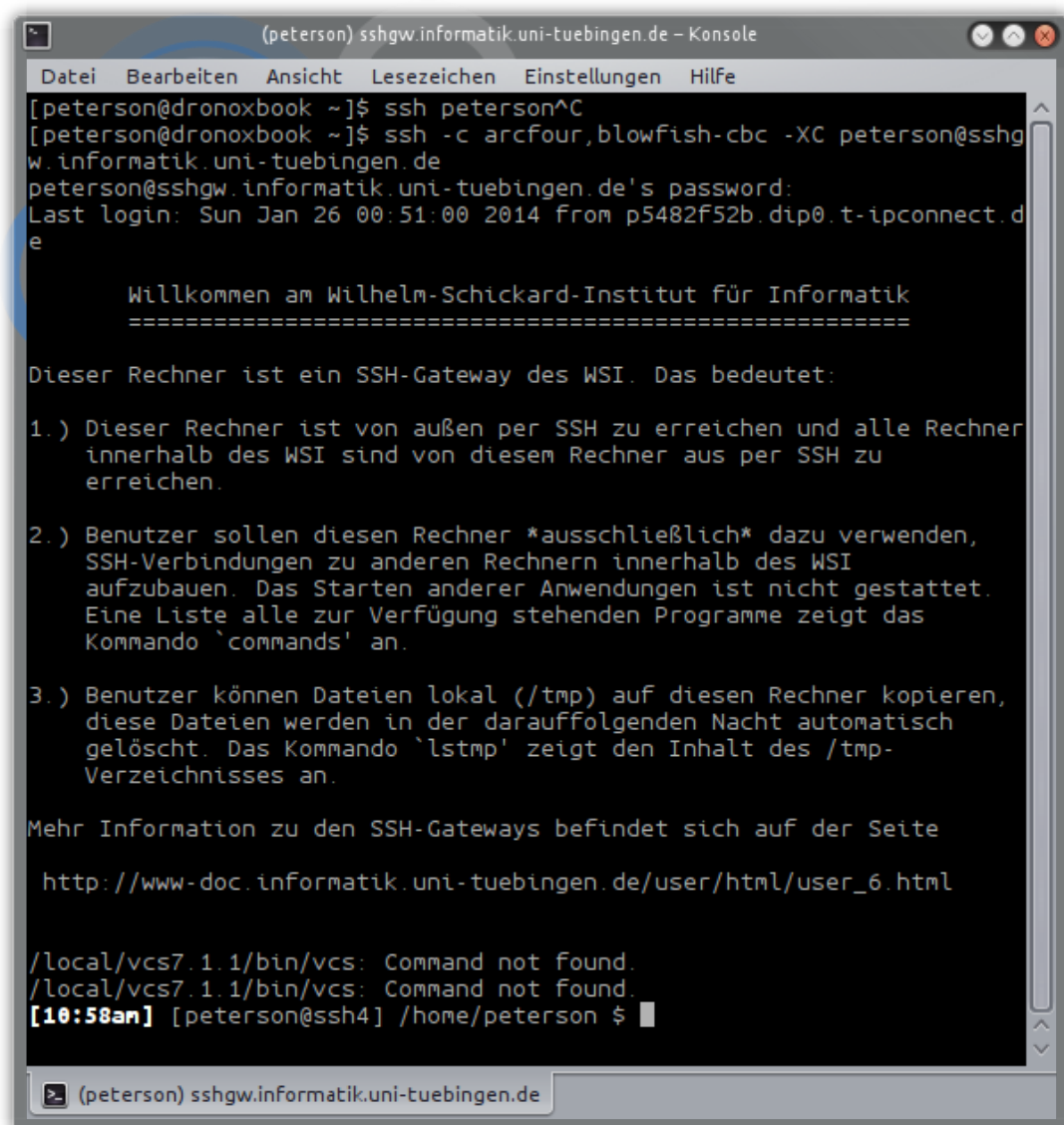


Abbildung 1: Ausgabe nach erfolgreicher Verbindung zum SSH-Gateway

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
```

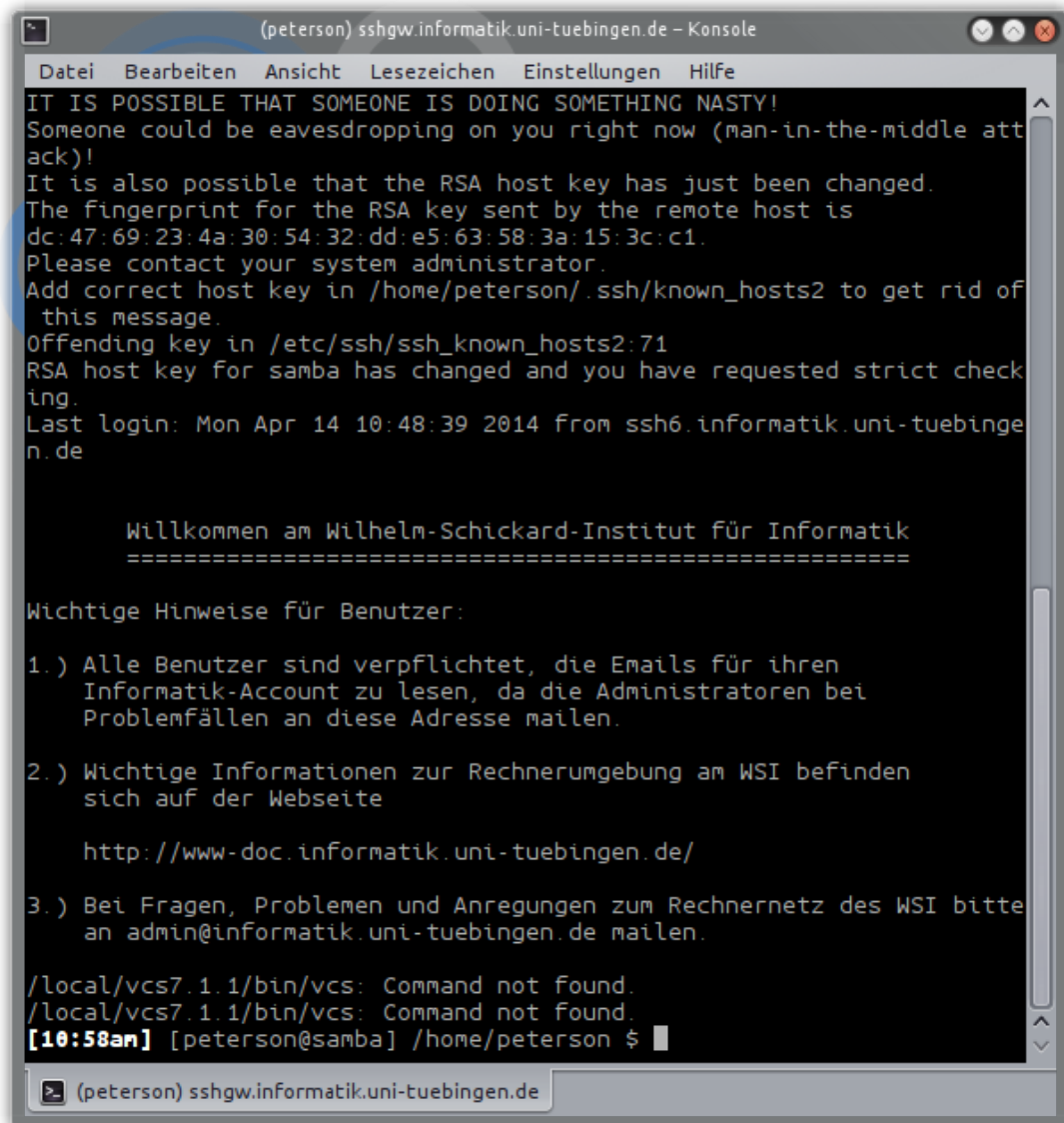


Abbildung 2: Ausgabe nach erfolgreicher Verbindung zum Übungsrechner (hier: samba)

Starten einer SSH-Verbindung unter Windows

Dieser Abschnitt beschäftigt sich mit dem Starten von Xcelium 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:

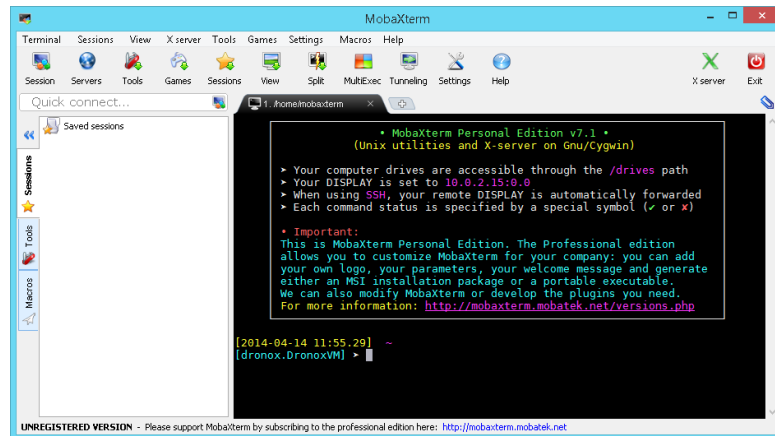


Abbildung 3: MobaXTerm nach dem Start

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:

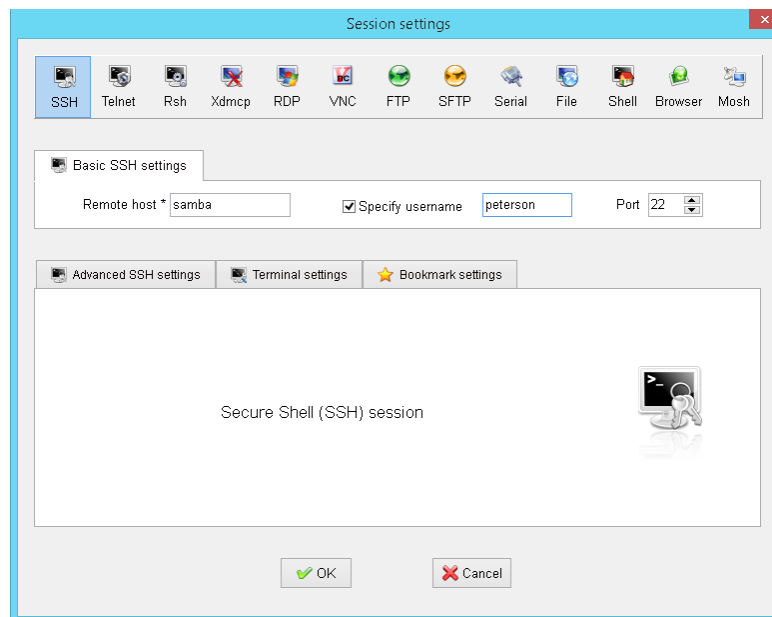


Abbildung 4: MobaXTerm: SSH-Verbindung konfigurieren (1)

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:

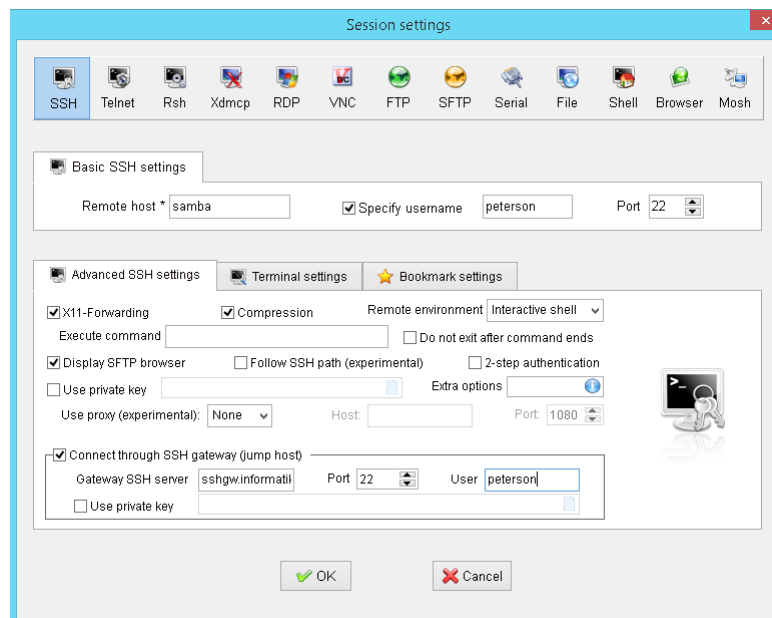


Abbildung 5: MobaXTerm: SSH-Verbindung konfigurieren (2)

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.

Hinweis: In neueren MobaXTerm-Versionen sind die Konfigurationsoptionen für das SSH-Gateway in ein neues Tab (Netzwerkoptionen) gewandert.

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.

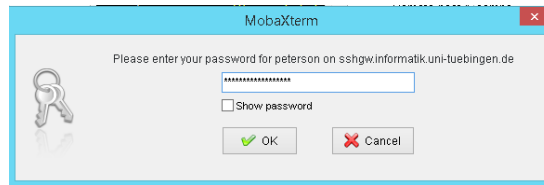


Abbildung 6: MobaXTerm: SSH-Verbindung starten (1)

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

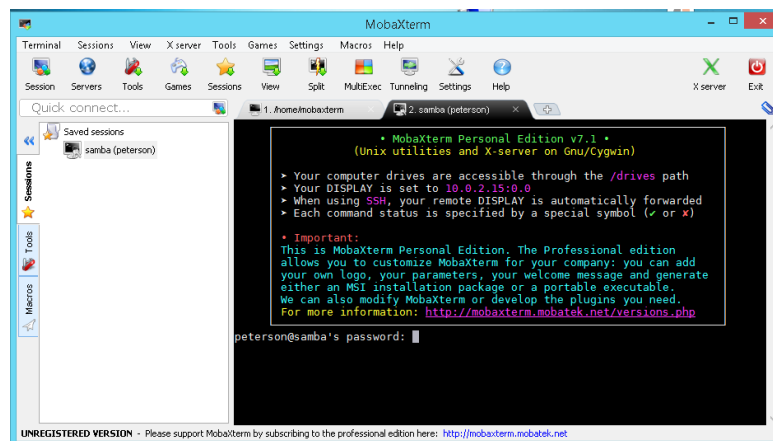


Abbildung 7: MobaXTerm: SSH-Verbindung starten (2)

Sie sollten nun eingeloggt sein.

Simulation mit Xcelium

Nachfolgend möchten wir Ihnen eine kompakte Schritt-für-Schritt-Anleitung für eine interaktive und automatisierte Simulation eines Halbbaddierers geben.

1. Verbinden Sie sich per X2Go oder mittels SSH zu einem Poolrechner.
2. Navigieren Sie in dem Terminal zu dem Ordner in der sich das Makefile und in unserem Fall auch die VHDL-Dateien befinden.
3. Passen Sie das Makefile an den entsprechenden Stellen an. Hierbei werden die VHDL-Source Dateien und Simulations- und Synthesevariablen, wie das Topmodul und Name der Waveform-Trace-Datei, gesetzt.
4. Machen Sie sich mit dem Makefile vertraut.
5. Führen Sie folgende Schritte für die automatisierte Simulation in der bash aus:

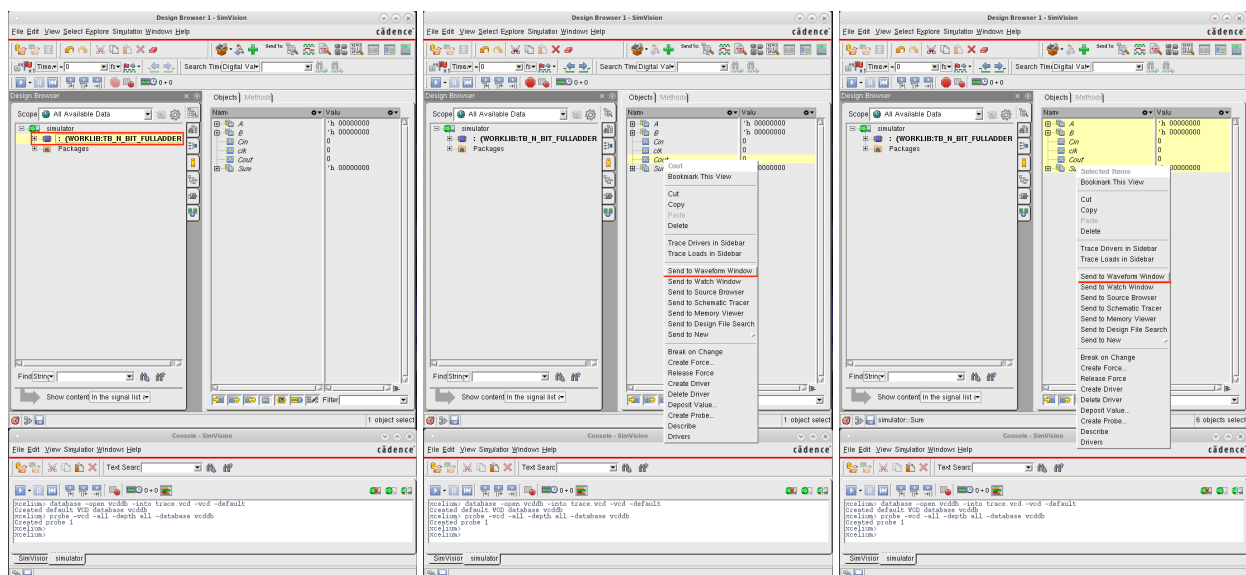
```
make clean # Loescht alle Dateien, die durch das Makefile erstellt wurden.
make sim   # Setzt die notwendigen Umgebungsvariablen und fuehrt die
           # Simulation mit Hilfe einer Testbench/Testtreiber aus.
```

6. Führen Sie folgende Schritte für die interaktive Simulation aus:

(a)

```
make clean # Loescht alle Dateien, die durch das Makefile erstellt
           # wurden.
make sim_gui # Setzt die notwendigen Umgebungsvariablen und startet die
            # Simulationsumgebung Xcelium.
```

- (b) Fügen Sie die Signale zum Waveform-Viewer hinzu. Wählen Sie hierzu im Design Browser das gewünschte Modul aus (siehe Abbildung 8a). Mit einem **Rechtsklick** können Sie einzelne oder alle Signale unter dem Reiter *Objects* auswählen und mittels **Send to Waveform Window** zum Waveform-Viewer hinzufügen (siehe Abbildung 8b, 8c).



(a) Modulwahl

(b) Signalauswahl

(c) Signalauswahl

Abbildung 8: Waveform-Viewer

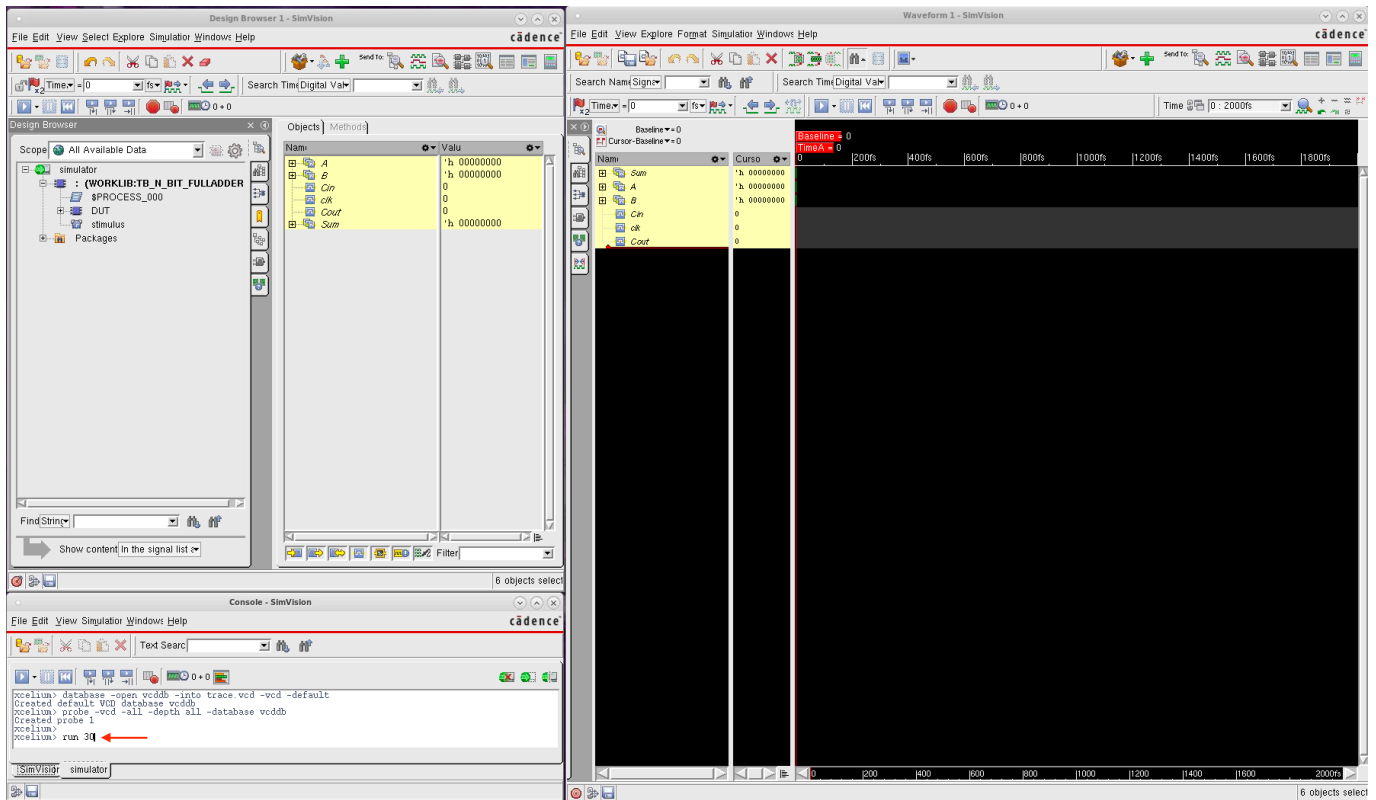


Abbildung 9: Waveform-Viewer

- (c) Es sollte sich nun ein neues Fenster Waveform 1 - SimVision geöffnet haben (siehe Abbildung 9) und die entsprechenden Signal aufgelistet sein.
- (d) Geben Sie nun **"run 30"** ein, um die Testbench für 30 Zeiteinheiten auszuführen. Entsprechend sollten Sie in dem Waveform-Viewer ein Fortschreiten der Signal sehen. Sollte keine Testbench vorhanden sein (zum Beispiel Übungsblatt 1), dann müssen die Signalwerte manuell über den Befehl **"force"** auf einen Wert gesetzt werden. Anschließend kann mit **"run 30"** wieder für die angegebene Zeit simuliert werden. Siehe Abbildung 10.

Beispiele für den **"force"**-Befehl:

```
force :Cin b"1"                # Cin erhaelt den Wert 1
force :A x"ABCDEF"             # A erhaelt den Hex-Wert ABCDEF
force :A b"10101011110011011110111110101010" # Entsprechend binaer
```

- 7. Beide Simulationmethoden erzeugen ein Waveform-Trace-File (.vcd - Value Change Dump). Dieses können Sie mit "gtkwave" öffnen. Sie hierzu Seite 14.

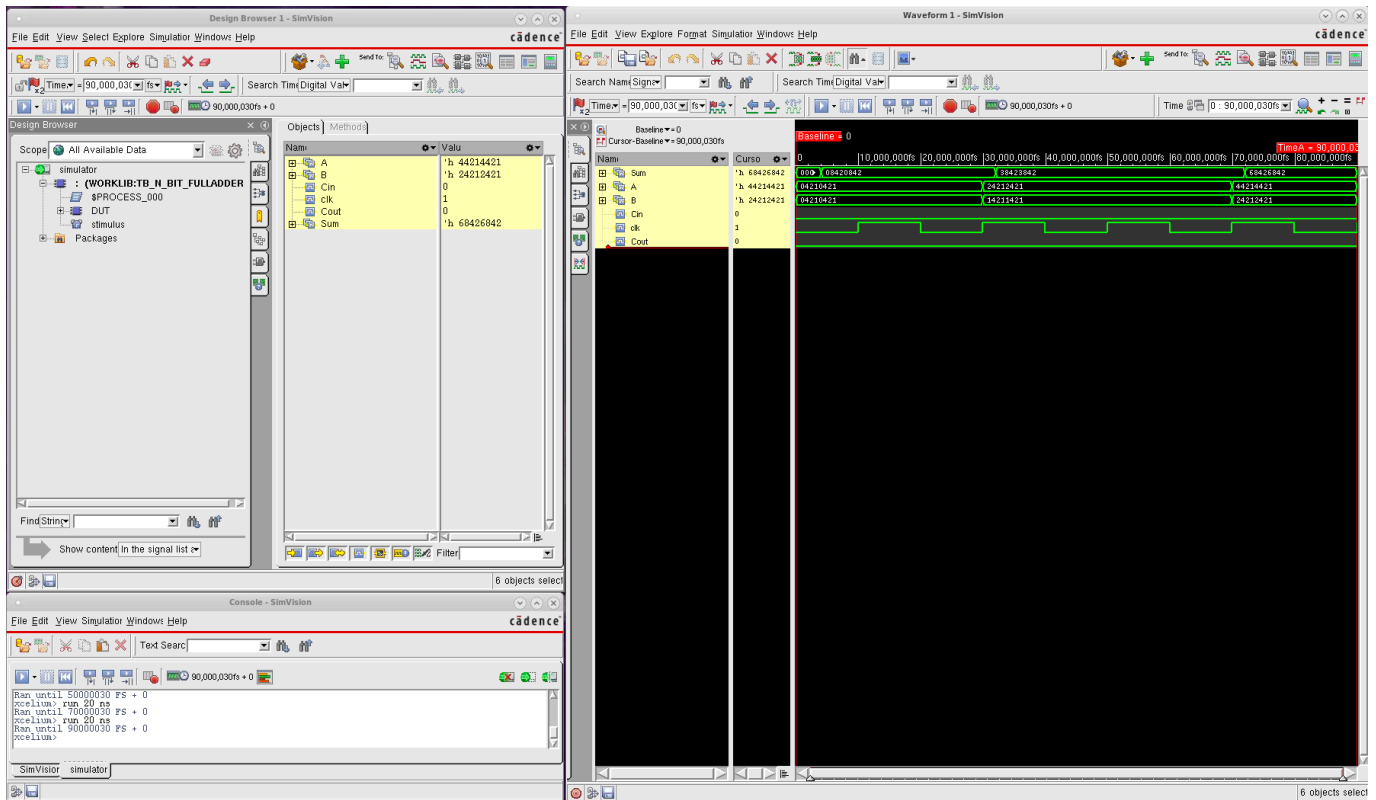


Abbildung 10: Waveform-Viewer

Generierung eines Value Change Dumps (VCD-Datei)

Zur Verifikation Ihrer Simulationsergebnisse benötigen wir den Export der Simulationstraces von Xcelium. 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 Xcelium 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:

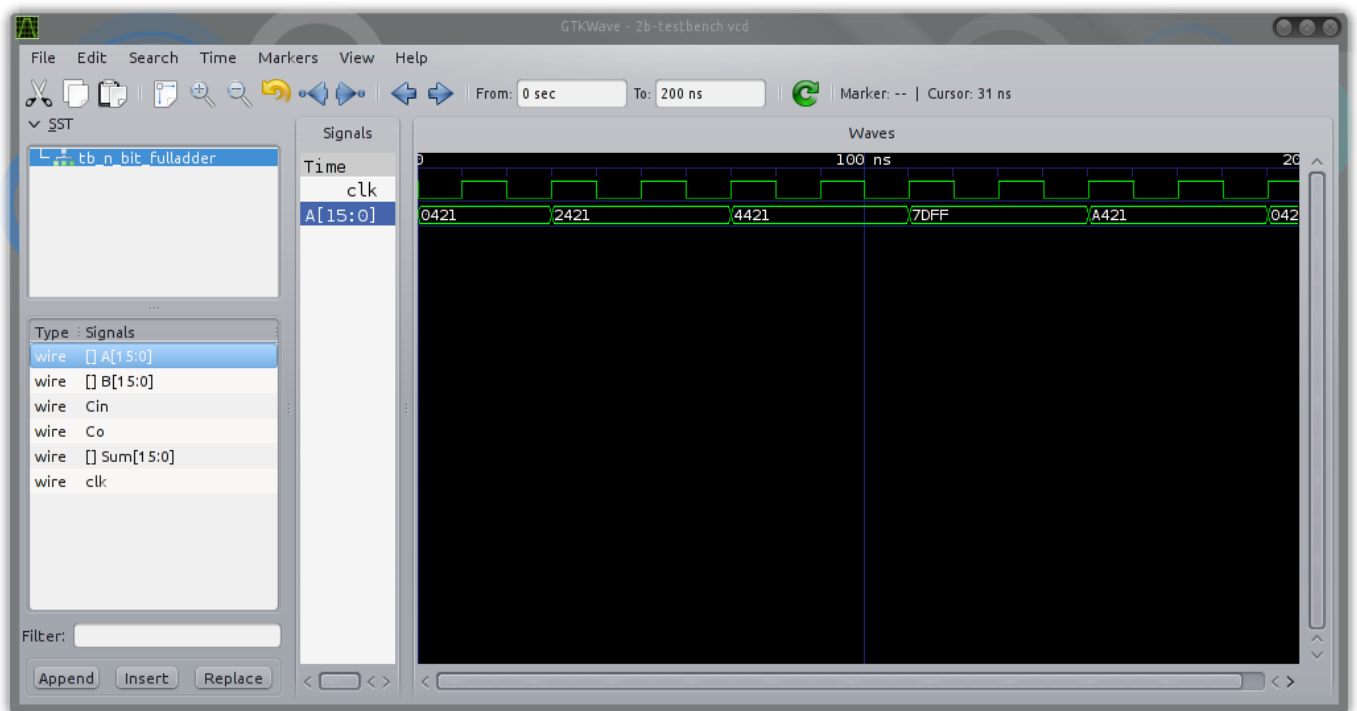


Abbildung 11: Beispielausgabe von GTKWave