

Aufgabe a

```
$ ghdl -s test_file.vhdl # Syntax-Check
```

Dieser Befehl wird für das Prüfen von Syntax von Dokumenten genutzt. Er aktualisiert die Bibliotheken nicht. Analysiert die Dateien, aber generiert kein Code.

```
$ ghdl -a test_file.vhdl # Analyse
```

Der Analysebefehl kompiliert eine oder mehrere Dateien und erstellt eine Objektdatei für jede Quelldatei. Der Analysebefehl wird mit dem Schalter -a ausgewählt. GHDL analysiert jeden Dateinamen in der angegebenen Reihenfolge und stoppt die Analyse im Fehlerfall (die folgenden Dateien werden nicht analysiert).

```
$ ghdl -e test_file # Build
```

Elaboration-Befehl: Unter GNU/Linux erstellt der Elaboration-Befehl eine ausführbare Datei, die den Code der VHDL-Quellen, den Elaboration-Code und Simulationscode enthält, um eine Designhierarchie auszuführen. Unter Windows führt dieser Befehl das Design aus, generiert aber nichts.

```
$ ghdl -r test_file --vcd=testbench.vcd # VCD-Dump
```

Run-Befehl: Unter GNU/Linux ermittelt dieser Befehl einfach den Dateinamen der ausführbaren Datei und führt sie aus. Optionen werden ignoriert. Sie können das Programm auch direkt ausführen.

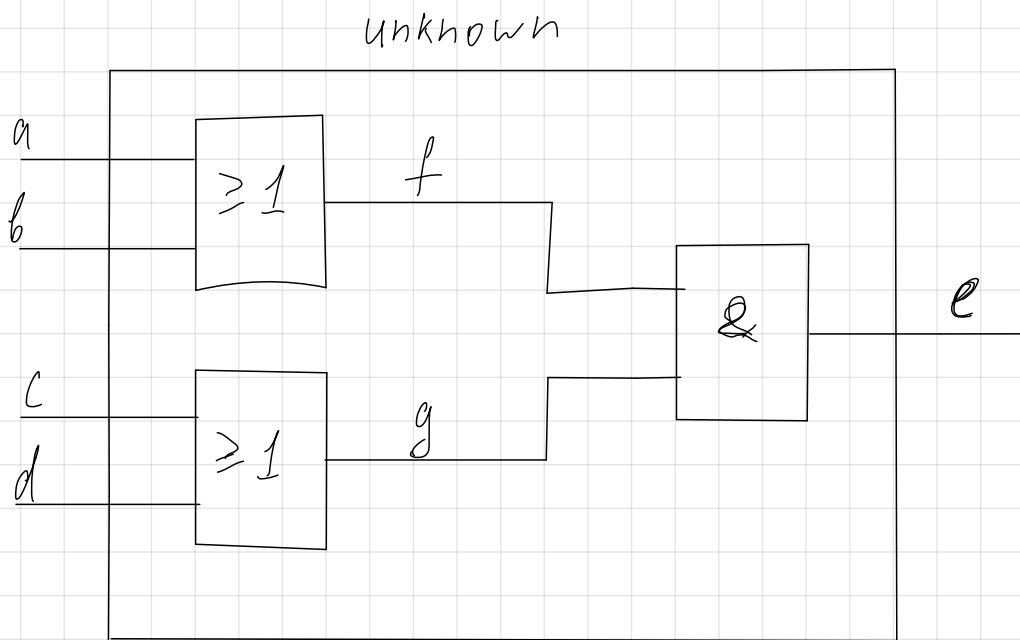
Unter Windows führt dieser Befehl die Simulation aus und startet sie. Daher müssen Sie die gleichen Optionen wie bei der Analyse verwenden.

```
$ gtkwave testbench.vcd # Startet GTKWave
```

Dieser Befehl verhält sich wie der Elaboration-Befehl (-e), gefolgt vom run-Befehl (-r).

Aufgabe d)

Skizze der Schaltung:



Boolsche Funktion:

$$(a \vee b) \wedge (c \vee d) \equiv e$$