ModelSim
(Mentor Graphics)

Aufruf

- *Einloggen auf einen Rechner im Hardwarepool, z.B. kojima oder tritton*
- *setupira mentorC4*
  - *setzt die Pfade für die Lizenzdateien*
- *cd <Verzeichnis mit VHDL-Quellen>*
- */usr/local/beckersoftware/modeltech/bin/vsim &*

ModelSim

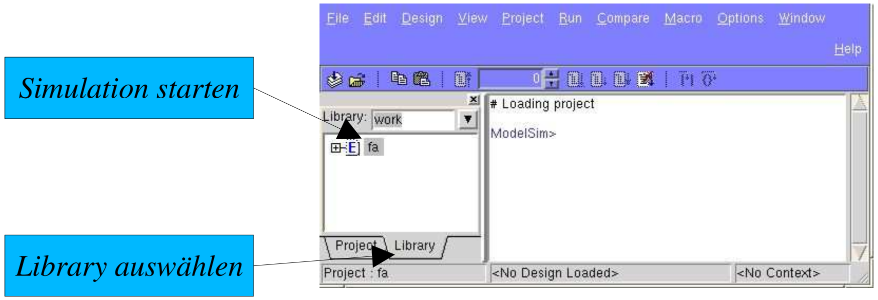
- *Simulation von VHDL und Verilog*
- *Hohe Genauigkeit*
- *Anzeige des Datenflussgraphs*
- *Code Coverage*
- *Debugging - Unterstützung*
- *Kann für Designs mit >1M Gattern verwendet werden*

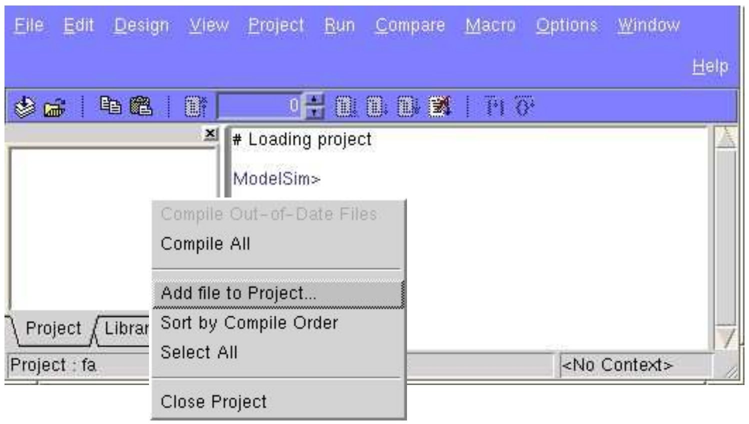
Anlegen eines Projektes

- *Create Project*
  - *Projektname sinnvoll wählen*
- *Files zum Projekt hinzufügen*
  - *Im Fenster »Project« auf rechte Maustaste drücken*
  - *»Add file to project« auswählen und eingeben*
- *Übersetzen*
  - *»Compile all« auswählen*

Simulieren

- *Im Hauptfenster »Library« anklicken*
- *Simulation starten durch Doppelklick auf die top-level Entity*





Weitere Fenster öffnen

- *Im Menü auswählen von »view all«*
  - *Fenster für Prozesse, Signale, Variablen*
  - *Datenfluss*
  - *Waveforms*
- *Anzuzeigende Waveforms über Fenster »signals«, Menüpunkt View → Wave auswählen*

Werte von Eingängen

- *Festlegen der Werte von Eingängen durch*
  - *Edit → Force im Fenster »signals«*
  - *Edit → Clock im Fenster »signals«*
  - *Schreiben einer Testbench*

Testbenches

- *Erzeugen Werte für die Eingangssignale einer Schaltung (Stimuli)*
- *Analogie zum Testen von (gefertigter) Hardware*
- *Schrittweises Verfolgen über »Run → Run -Next« im Hauptfenster*

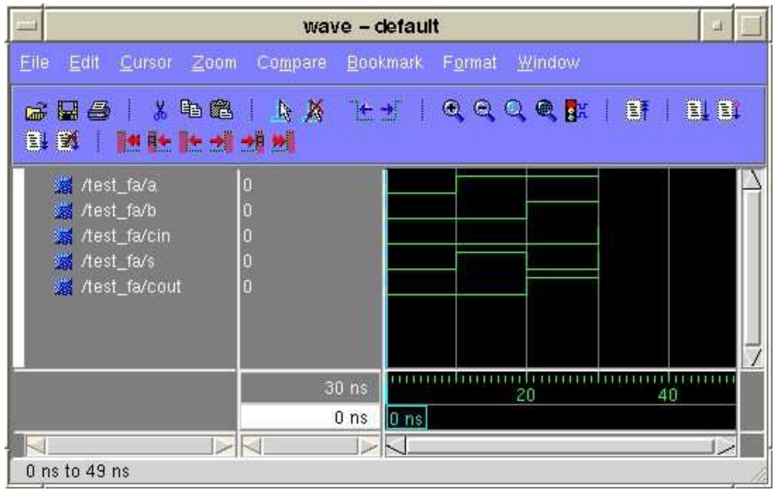
Beispiel

```
library ieee;
use ieee.std_logic_1164.all;

entity test_fa is
end test_fa;

architecture test of test_fa is

component fa
    port(a, b, cin : in  std_logic;
         s, cout   : out std_logic);
end component;

signal a, b, cin, s, cout : std_logic;

begin
```

Beispiel (cont.)

```
begin
dut : fa port map (a, b, cin, s, cout);

stimulus : process
begin
   a <= '0'; b <= '0'; cin <= '0';
   wait for 10 ns;
   a <= '1';
   wait for 10 ns;
   b <= '1';
   wait for 10 ns;
   cin <= '1';
   wait;
end process;
end test;
```

Waveforms



Ausblick

- *Schrittweises Verfolgen der Simulationsschritte (Delta-genau)*
- *Vergleichen zweier Simulationsläufe*
  - *Verifikation durch Simulation*
- *Diagnosemöglichkeit*