

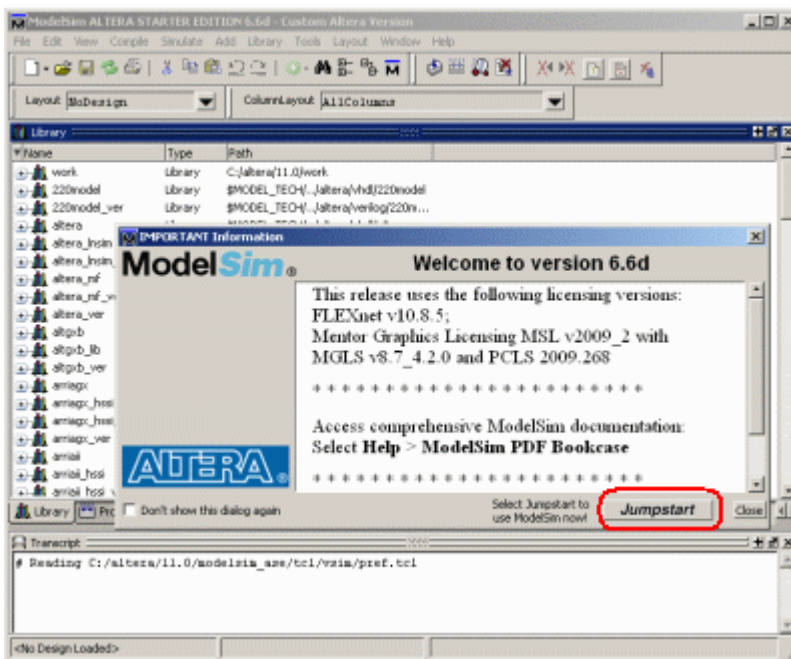
# Testbänk i ModelSim



*Välj rätt programversion - i skolan finns flera olika installerade under startmenyn!*

Altera 13.0.1.232 Web edition\  
ModelSim-Altera Starter Edition 13.0.1.232\  
ModelSim-Altera 10.1d(Quartus II 13.0sp1)

Starta ModelSim.

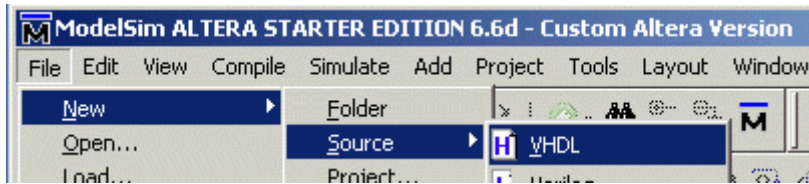


Klicka på **Jumpstart**. Denna gång väljer vi länken "Open Project" för att fortsätta med det gamla MAXsim-projektet.




## Testbänken

Förutom VHDL-koden för låset, behöver vi nu ytterligare en VHDL-fil för testbänkskoden.

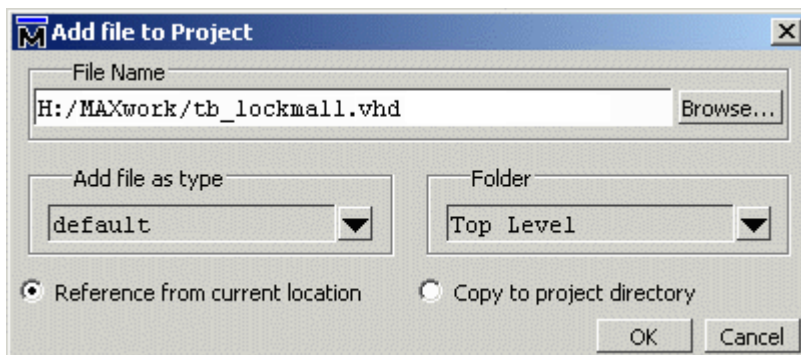
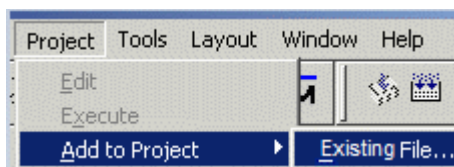


Skapa en ny tom VHDL-fil. Kopiera och klistra in innehållet från den givna filen `tb_lockmall.vhd` och spara sedan filen under samma namn, `tb_lockmall.vhd`, bland de övriga filerna i projektet.

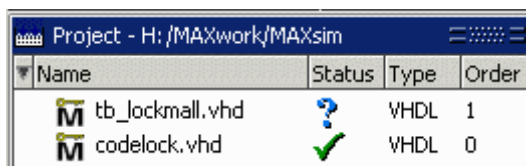
Alternativt kan man kopiera över hela filen `tb_lockmall.vhd` till mappen med de övriga filerna i projektet.



 [tb\\_lockmall.vhd](#) ([tb\\_lockmall.vhd.txt](#))

## Lägg till VHDL-filen till projektet







Projektet har nu två filer. Filen `tb_lockmall.vhd` är inte kompilerad ännu, vilket syns på det blåa frågetecknet.



Name	Status	Type	Order
 tb_lockmall.vhd	?	VHDL	1
 codelock.vhd	✓	VHDL	0

Välj menyvalet **Compile** och alternativet **Compile All**. Nu kompileras filen `tb_lockmall.vhd`. Av innehållet i den filen framgår att den styr `codelock.vhd`, så det är därför den har det högsta ordningsnumret.

Name	Status	Type	Order
 tb_lockmall.vhd		VHDL	1
 codelock.vhd		VHDL	0

## Ladda Designen till simulatören.

Vi vill att testbänkens meddelanden ska visas (som pilar) överst i **Wave**-fönstret. Därför skriver vi i **Transcript**-fönstret:

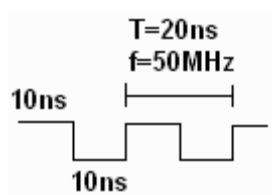
```
vsim -msgmode both -displaymsgmode both tb_codelock
```

Därefter lägger man till signalerna i **Wave**-fönstret.

```
add wave *
```

## Starta simuleringen

I filen `tb_codelock.vhd` står raden `clk <= not clk after 10 ns;`. Den skapar de klockpulser med periodtiden 20 ns som vi använt tidigare.



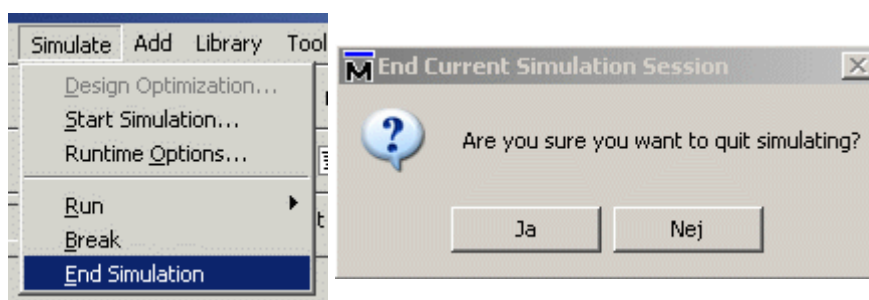
Vi kan direkt starta simuleringen med kommandot `run 3us` i **Transcript**-fönstret. Tiden 3  $\mu$ s räcker för att prova igenom alla tänkbara tangent-tryckningskombinationer.

I **Transcript**-fönstret kan vi läsa det uppmuntrande budskapet `Note: Lock tries to open for the right sequence!`.

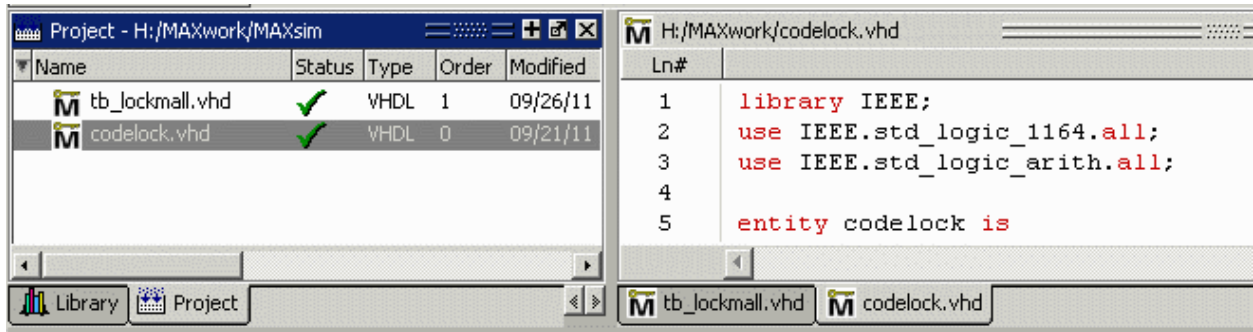
I **Wave**-fönstret väljer man **View**, **Zoom**, och **Zoom Full** och kan då se hela förloppet.




De gröna pilarna visar *när* den önskade händelsen händer!  
Avsluta nu simuleringen.



## Avslöja den felaktiga koden!



Nu gäller det att i **Project**-fönstret dubbelklicka på filen `codelock.vhd` så att den visas i texteditorn och där går att förändra. Kopiera och klistra in innehållet från filen `lockmall_with_error.vhd`. Spara sedan filen under samma namn som tidigare och kompilera om allt.

 [lockmall\\_with\\_error.vhd](#) ([lockmall\\_with\\_error.vhd.txt](#))

Simulera sedan på samma sätt som tidigare. I **Transcript**-fönstret kan Du läsa meddelanden från testbänken när önskade eller oönskade händelser sker.

Den röda pilen i **Wave**-fönstret visar *när* den oönskade händelsen händer!



## VHDL-testbänksfilen

Du behöver naturligtvis inte kunna skriva en VHDL-testbänksfil efter en kort första kurs om Digital Design. Tag ändå tillfället i akt att gå igenom filen och se om Du kan förstå hur den är tänkt!

Se **VHDL testbänksfilen**.

William Sandqvist [william@kth.se](mailto:william@kth.se)