

F3_001 Test av design i Questa

For testing benyttes verktøyet Questa. Dette er et programverktøy som alle VHDL konstruktører benytter, blir og kalla Modelsim.

Utgangspunkt:

I Quartus er det opprettet et prosjektet *mux2*. Vhdl-filen heter *mux2.vhd*. Prosjektet er syntetisert og uten syntaks feil.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

--Multiplekser.
--INN:har et velgersignal og to 1-bit-signallinjer
--UT: 1-bit-signallinje
ENTITY Mux2 IS
    PORT(s :IN std_logic;
          x: in std_logic;
          y: in std_logic;
          m:OUT std_logic);
END ;

Architecture behavior OF Mux2 IS
begin
    -- m <= (x and not(s))OR ((y and s) );
    with s select
        m <= x WHEN '0',
            y WHEN '1',
            '0' When others;
END;
```

Test av design: Lag testbenk:

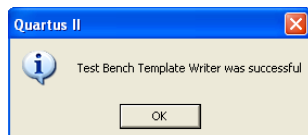
Det neste som må gjøres er å teste koden og kontrollere at den fungerer som den skal. Uttesting i Questa benyttes som dokumentasjon på konstruksjonen.

Dersom Questa **ikke** starter, gå til " Dersom ikke Questa starter.", som står på siste siden.

Å lage testbenk-malen.

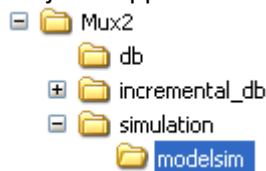
Denne lages i Quartus. Forsikr deg om at du har kjørt "Analysis & Synthesis", eller har kompilert prosjektet.

Velg så *Processing/Start/Start test bench template writer* .

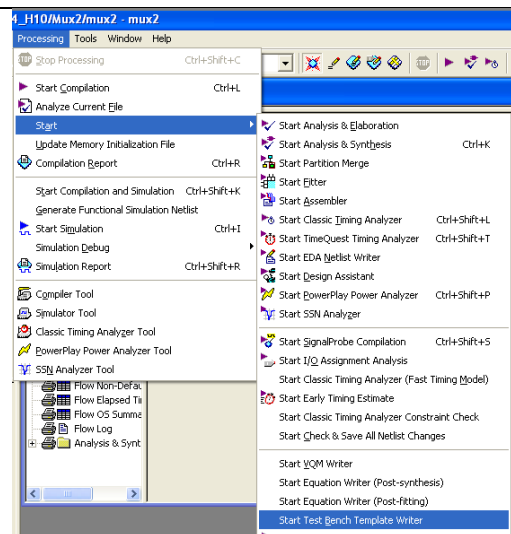


klikk ok

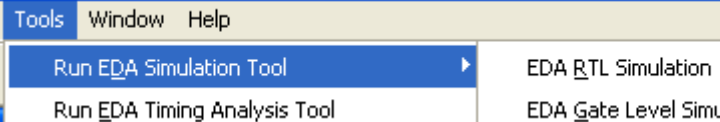
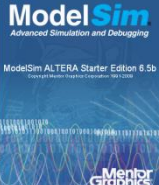
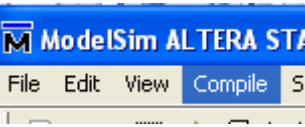
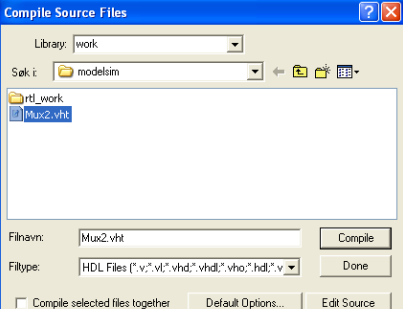
Prosjektmappen ser nå slik ut:



I mappen modelsim ligger nå filen *mux2.vht*.



NB! Dersom du senere gjør endringer i entity i *mux2.vhd*, må du kjøre dette på nytt, eller gjøre nødvendige endringer i *mux2.vht* . Dersom du kjører *Processing/Start/Start test bench template writer* , overskrives filen *mux2.vht*.

<p>Til venstre ser du filen mux2.vht, som ble opprettet da <i>Processing/Start/Start test bench template writer</i> ble kjørt.</p> <p>Den skal vi bruke i Questa, og den skal benyttes til å teste kretsen vi har laget.</p> <p>For å gjøre det, skriver vi inn kommandoer i prosessen Init. Init er et navn som kan velges fritt. Init er "merkelappen" på prosessen. Om vi vil, kan den endres til test_mux2.</p> <p>De kombinatoriske uttrykk vi skriver for å teste mux2, skal stå etter BEGIN og før WAIT.</p> <p>WAIT kan bare brukes i Process.</p>	<pre> 27 LIBRARY ieee; 28 USE ieee.std_logic_1164.all; 29 30 ENTITY Mux2_vhd_tst IS 31 END Mux2_vhd_tst; 32 ARCHITECTURE Mux2_arch OF Mux2_vhd_tst IS 33 -- constants 34 -- signals 35 SIGNAL m : STD_LOGIC; 36 SIGNAL s : STD_LOGIC; 37 SIGNAL x : STD_LOGIC; 38 SIGNAL y : STD_LOGIC; 39 COMPONENT Mux2 40 PORT (41 m : OUT STD_LOGIC; 42 s : IN STD_LOGIC; 43 x : IN STD_LOGIC; 44 y : IN STD_LOGIC 45); 46 END COMPONENT; 47 BEGIN 48 i1 : Mux2 49 PORT MAP (50 -- list connections between master ports and signals 51 m => m, 52 s => s, 53 x => x, 54 y => y 55); 56 init : PROCESS 57 -- variable declarations 58 BEGIN 59 -- code that executes only once 60 WAIT; 61 END PROCESS init; 62 always : PROCESS 63 -- optional sensitivity list 64 -- () 65 -- variable declarations 66 BEGIN 67 -- code executes for every event on sensitivity 68 WAIT; 69 END PROCESS always; 70 END Mux2_arch; </pre>
<p>Start Questa fra Quartus: Tools/Run EDA Simulation Tool/ EDA RTL Simulation</p>	
<p>Questa starter. Mux2.vhd blir kompilert ved oppstart. Gå tilbake til Quartus og rett opp feilene dersom du får feilmeldinger.</p>	
<p>Kompiler I Questa: start med å velge Compile</p>	
<p>Pass på at du velger rett fil, Mux.vht. Når kompileringen er ferdig, ser Transcript vinduet slik ut:</p>	

```

# Errors: 0, warnings: 0
#
vcom -reportprogress 300 -work work {C:/Users/esk/Box Sync/Qua
# Questa Intel Starter FPGA Edition-64 vcom 2021.2 Compiler 20
# Start time: 16:24:32 on Aug 24,2022
# vcom -reportprogress 300 -work work C:/Users/esk/Box Sync/Qu
# -- Loading package STANDARD
# -- Loading package TEXTIO
# -- Loading package std_logic_1164
# -- Compiling entity mux2_vhd_tst
# -- Compiling architecture mux2_arch of mux2_vhd_tst
# End time: 16:24:32 on Aug 24,2022, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0

Questa> ]

```

<No Design Loaded>
<No Context>

Lukk Compile-vinduet ved å klikke på *Done*.

Editor filen:
Vi kan nå begynne å sette inn kommandoer.

Åpn Mux2.vht og skriv inn der pilen viser, det som står i ruten.
Lagr.
Kompilerer på nytt
(Husk alltid å lagre før du kompilerer på nytt!!)

```

init : PROCESS
-- variable declarations
BEGIN
    -- code that executes only once
WAIT;
END PROCESS init;

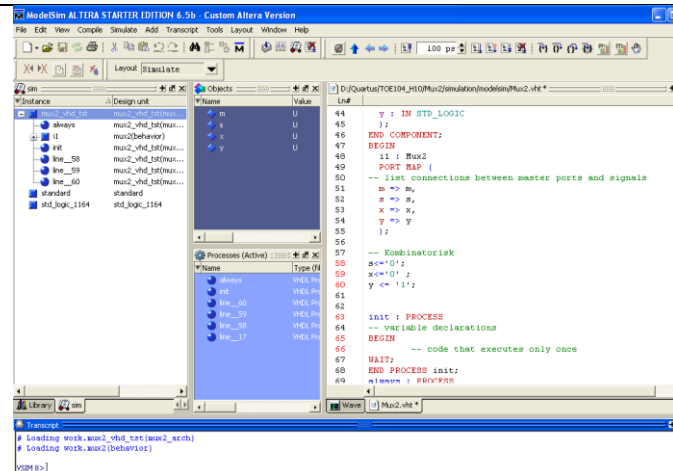
```

62 s <= '0';
63 x <= '0';
64 y <= '1';

Simulere:
Klikk på Simulate. Velg Start Simulation

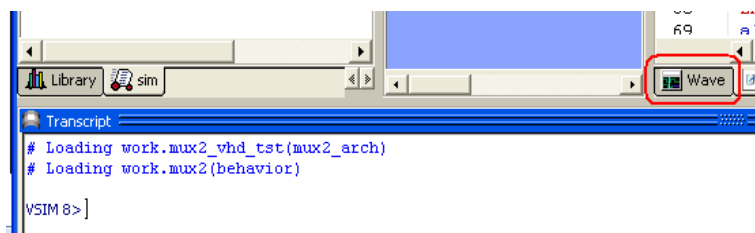
I vinduet som kommer opp, skal du stå i Design.
Klikk på work
mux_vhd_tst.
OK

Questa ser nå slik ut:

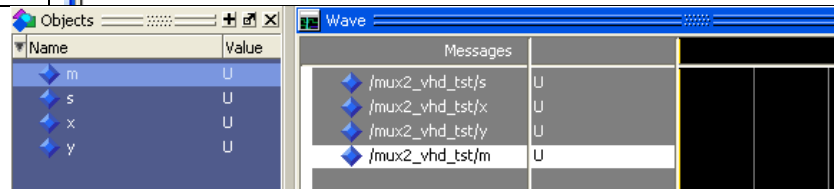


Klikk på Wave.

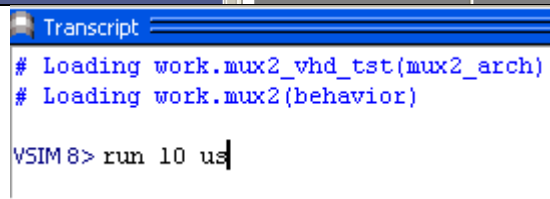
Dersom Wave ikke er aktivt, finner du det her:
View/Wave



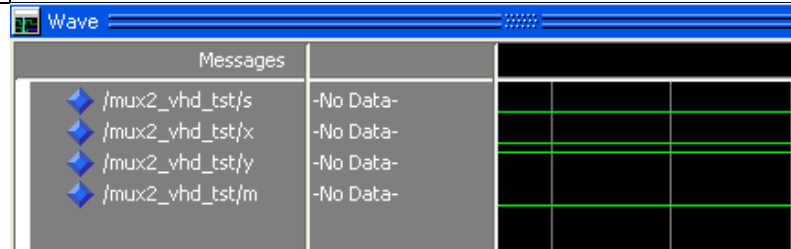
Dra s,x,y og m over til Wave



Vi setter simulereringstiden til 10 usek
NB!: husk mellomrom mellom 10 og us

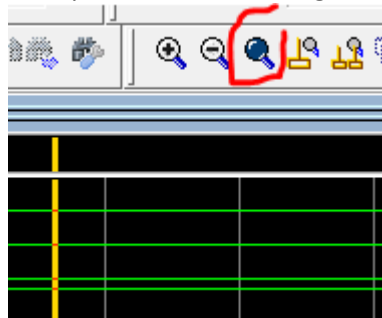


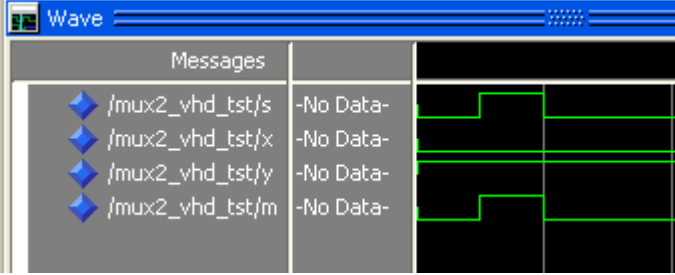
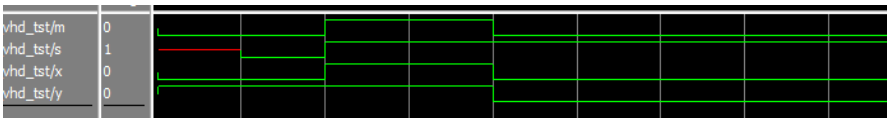
Analyse av resultatet:
OK, for vi satte s=0, x=0,y=1.
Da skal m være lik x. Det er den.

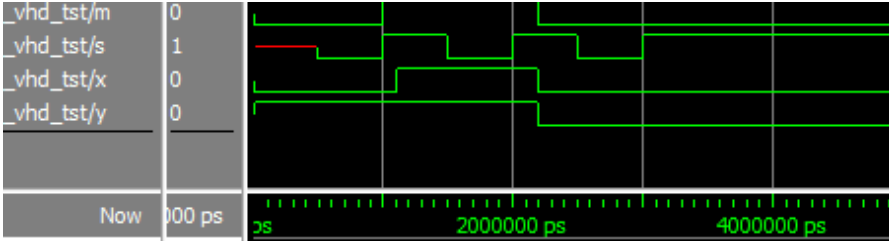
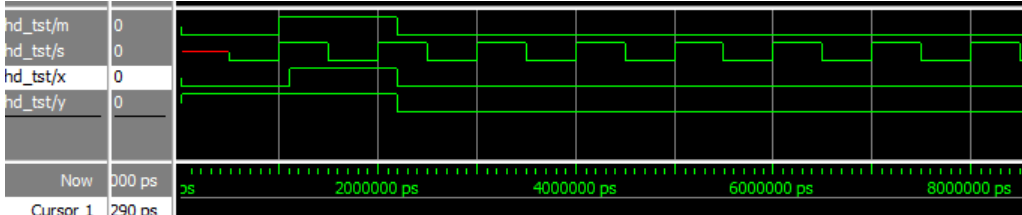


For å være sikre på at du ser alt:

Klikk på området med de grønne linjene, klikk deretter på forstørrelsesglasset.



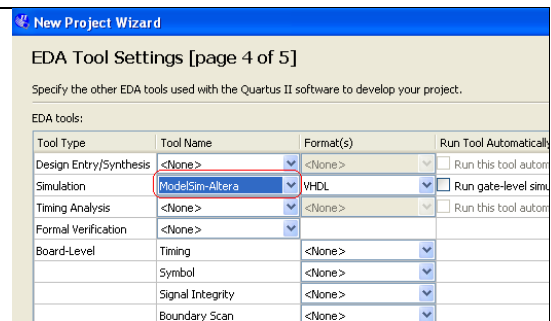
<p>Utvid simuleringskoden slik figuren til høyre viser.</p> <p>Lagr filen</p> <p>Kompiler.</p>	<pre> 57 init : PROCESS 58 59 BEGIN 60 s <= '0'; 61 x <= '0'; 62 y <= '1'; 63 Wait for 1 us; 64 s <= '1'; 65 Wait for 1 us; 66 s <= '0'; 67 WAIT; 68 69 END PROCESS init; </pre>
<p>Simuler.</p> <p>Klikk på Wave</p> <p>Trekk signalene du vil se på over i Wave.</p> <p>La velgersignalet s stå øverst, deretter andre innsigna, l og til sist utsignalet m</p> <p>Skriv run 10 us i Transcript-vinduet, etter VSIM.</p>	
<p>Legg til kode</p> <p>Nå gjøres følgende forandring:</p> <p>Vi har en egen prosess for velgersignalet s</p> <p>Prosessen gis navet <i>Velger</i>.</p> <p>Skriv inn koden</p> <p>Lagr</p> <p>Kompiler</p> <p>Simuler</p> <p>Klikk på Wave</p> <p>Flytt over signalene du vil se på resultatet</p> <p>S beskrives som et periodisk signal</p>	<pre> velger : PROCESS BEGIN s <= '0' after 500 ns, '1' after 2*500 ns; WAIT; END PROCESS velger; init : PROCESS BEGIN x <= '0'; y <= '1'; wait for 1 us; x <= '1'; y <= '1'; wait for 1 us; x <= '0'; y <= '0'; WAIT; END PROCESS init; END mux2_arch; </pre>  <p>Legg merke til at s får verdi først etter 500 ns. Og settes høy etter 1000 ns</p> <pre> s <= '0' after 500 ns, '1' after 2*500 ns, '0' after 3*500 ns, '1' after 4*500 ns, '0' after 5*500 ns, '1' after 6*500 ns;; </pre> <p>For at x og y ikke skal skifte samtidig som s, forskyves x og y:</p>

	<pre> init : PROCESS BEGIN x <= '0'; y <= '1'; wait for 1100 ns; x <= '1'; y <= '1'; wait for 1100 nus; x <= '0'; y <= '0'; WAIT; END PROCESS init; </pre> 
En liten endring i koden gjør at s gjentas periodisk.	<pre> velger : PROCESS BEGIN s <= '0' after 500 ns, '1' after 2*500 ns; WAIT for 2*500 ns; END PROCESS velger; init : PROCESS BEGIN x <= '0'; y <= '1'; wait for 1100 ns; x <= '1'; y <= '1'; wait for 1100 ns; x <= '0'; y <= '0'; WAIT; END PROCESS init; </pre> 

Dersom Questa ikke starter:

Sjekk at questa –Altera ble valgt

Når et prosjekt blir opprettet, er det mulig å angi hvilket simuleringsverktøy som skal benyttes. Velg da Questa Intel FPGA under Simulation.

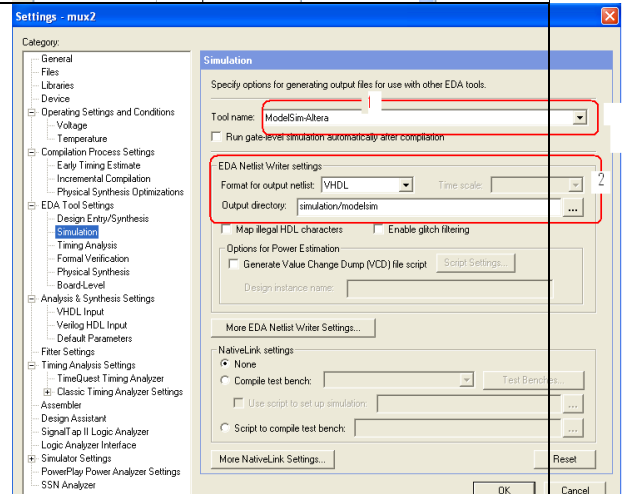


Valg av simuleringsverktøy kan også gjøres etter at prosjektet er opprettet. Det gjøres slik:

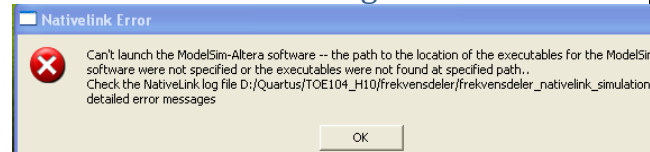
Velg:

Assignment/EDA Tool Setting/Simulation
Under Simulation velges Questa Intel FPGA.

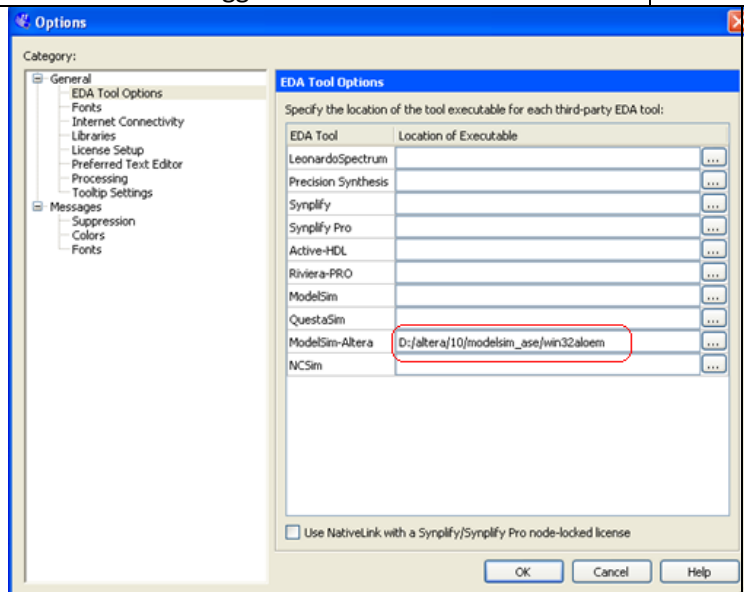
Felt 2, EDA Netlist Writer settings, skal komme opp av seg selv.



Får du denne feilmeldingen?



Sjekk om Quartus vet hvor oppstartsfilen til ModelSimAltera ligger :



I Quartus må vi sette inn hvor ModelSim Altera oppstartsfile er plassert. Dette gjøres under:

QuartusII Tools -> Options -> General -> EDA Tool Options .

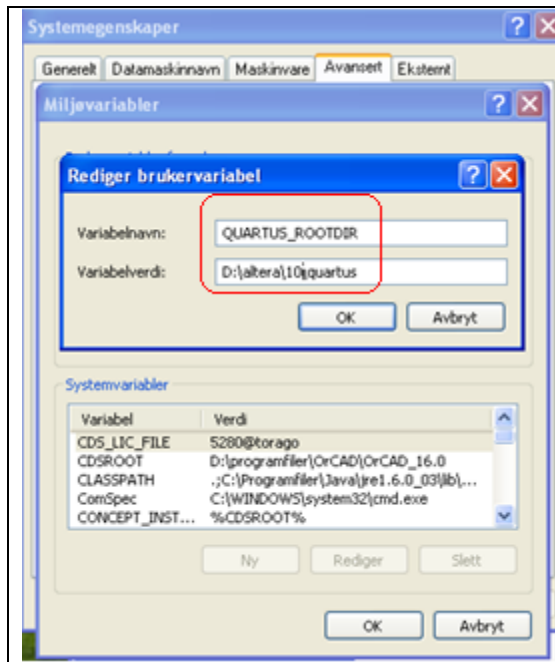
I dette tilfellet er ModelSimAltera installert i D:\altera\10.

Bla deg nedover i katalogene til

/modelsim_ase/win32aloem

Test om ModelSim starter nå.

Hvis ikke se etter om du har rette miljøvariabler:



Din variabelverdi:

Her må du sette miljøvariabler i Min



datamaskin.

Høyreklikk og velg egenskaper ->Avansert ->Miljøvariabler -> Rediger under Brukeregenskaper. Sett inn peker på hvor quartus er installert. I dette tilfellet er Quartus installert under D:\altera\10