

Het testen van de realisatie tegen een golden unit (case display).

Na het synthetiseren van display met Quartus II kan de realisatie (*display.vho* en *display_vhd.sdo*) worden vergeleken met het oorspronkelijke ontwerp (*display.vhd*).

Maak een testomgeving waarin een uitputtende test wordt uitgevoerd en de golden unit wordt vergeleken met het postsimulatie-resultaat. Bestudeer de waveform.

N.B. In hoofdstuk 4 van de tutorial is de procedure voor de postsimulatie beschreven. Let er op dat je de label van de componentinstantiatie invult bij "apply to region". Verder is het belangrijk ModelSim-Altera gebruikt voor de postsimulatie omdat de benodigde bibliotheken dan al aanwezig zijn.

De door Quartus II gegenereerde file *display.vho* bevat o.a. de volgende entity beschrijving:

```
LIBRARY CYCLONEII;
LIBRARY IEEE;
USE CYCLONEII.CYCLONEII_COMPONENTS.ALL;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY display IS
    PORT (
        dig : OUT std_logic_vector(6 DOWNTO 0);
        hex : IN std_logic_vector(3 DOWNTO 0)
    );
END display;
```

N.B.:

1. Omdat het ontwerp is gesynthetiseerd naar een Cyclone II wordt in de postsimulatie ook gebruik gemaakt van deze bibliotheek. Daarom staan er boven de entity een aantal extra library/use clauses.
2. De port declaratie in de entity komt overeen met die in het oorspronkelijke ontwerp. In het algemeen hoeft dat niet zo te zijn. Is in de entity beschrijving een poort opgenomen van het type "integer range 0 to 15" dan kan de tool dit in het postsimulatiebestand realiseren met het type "std_logic_vector(3 downto 0)". Dat is niet handig immers de testomgeving die eerder is gebruikt om het ontwerp te testen kan daarna niet eenvoudig worden gebruikt voor de postsimulatie. Daarom wordt vaak geëist dat op top-niveau alleen gebruik gemaakt mag worden van de types std_logic en std_logic_vector.

In de opdracht moet een testomgeving worden gemaakt waarbij het oorspronkelijke ontwerp wordt vergeleken met de realisatie.

Dit is een kleine uitbreiding op `display_test` (`display_test.vhd`). Hierin wordt alleen het ontwerp getest. Eigenlijk is alleen nog een tweede entity geïnstantieerd nodig voor het testen van de realisatie.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.numeric_std.ALL;
ENTITY display_test IS
END display_test;

ARCHITECTURE exhaustive OF display_test IS
    SIGNAL hex : unsigned(3 DOWNTO 0);
    SIGNAL displ, displ2 : std_logic_vector(6 DOWNTO 0);

BEGIN
    PROCESS
    BEGIN
        FOR i IN 0 TO 15 LOOP
            hex <= to_unsigned(i,4);
            WAIT FOR 20 ns;
            -- user has to verify the result. (or a check with golden unit here)
        END LOOP;
        WAIT;
    END PROCESS;

    golden_unit : ENTITY work.display(bhv)
        PORT MAP (hex => std_logic_vector(hex), dig => displ);

    realization : ENTITY work.display(structure)
        PORT MAP (hex => std_logic_vector(hex), dig => displ2);

END exhaustive;
```

N.B.:

1. Er is nog een klein probleem. In de bestanden `display.vhd` en `display.vho` is een entity `display` aanwezig. Compileer je eerst `display.vhd` en daarna `display.vho` dan zal bij het bij de start van de simulatie (eigenlijk nog de elaboratiefase) worden gemeld:
** Error: (vsim-13) Recompile work.display(bhv) because work.display has changed.
Load interrupted
Error loading design
2. Dit kan worden opgelost door een van onderstaande oplossingen:
 - a. Verwijder de entity beschrijving uit `display.vhd` (of plaats dit in commentaar) en compileer eerst `display.vho` en daarna `display.vhd`
 - b. Ook kan er voor worden gekozen om de naam van de entity te wijzigen (vergeet dan niet om deze ook in de architecture te wijzigen:
architecture bhv of <nieuwe naam entity> is

Postsimulatie geeft onderstaande waveform.

