VHDL Opdracht CAO-2

1. Hexadecimal naar 7-segment

In deze opdracht wordt een hexadecimaal naar 7-segment display gerealiseerd op een FPGA (Cyclone II starter kit met een Altera-Cyclone EP2C20F484C7 device).

a) Bestudeer het schema van de Cyclone II starter kit. Welke spanning staat er op de ingang van de pin van de FPGA wanneer een "key" is ingedrukt? Bij welke uitgangsspanning gaat een LED branden van het 7-segment display?

b) Bestudeer het ontwerp (`display.vhd`) en voer een simulatie uit.

c) Voor dit ontwerp is ook een eenvoudige testomgeving aanwezig (`display_test.vhd`). Een testomgeving wordt nog fraaier wanneer je het resultaat met een golden unit kunt vergelijken (komt later). Deze is er niet, dus moet de gebruiker zelf de controle uitvoeren.
Kijk ook eens naar de type conversie in de "port map". (Detail: een ingangssignaal heeft de type conversie rechts van het symbool "=>", bij een uitgangssignaal is staat de type conversie links van het symbool "=>").

Simuleer de testomgeving

d) Voor de realisatie wordt gebruik van Cyclone II starter kit. Op dit bord zijn o.a. de 7-segment displays aangesloten op de pinnen van de FPGA. De constraintfile `display.qsf` beschijft deze verbindingen. Bij Quartus moet deze constraintfile in dezelfde directory staan als het ontwerp. De naam van de constraintfile moet overeenkomen met de naam van de entity.
Let op: de constraintfile wordt door Quartus aangepast! Beewaar zelf altijd en kopie! Gebruik bij een nieuwe syntheserun het originele qsf bestand.

Realiseer het ontwerp `display` op het bordje (in de tutorial is de procedure beschreven) en test de schakeling.

e) In vraag c) is aangegeven dat een test waarbij je een simulatie uitvoert tegen een golden unit handig is. Je kunt wat anders doen, koffie drinken of zo, terwijl de test wordt uitgevoerd. In dit ontwerp beschouwen we de oorspronkelijke functie `hex2display` (in file `display.vhd`) als de golden unit en willen we het gerealiseerde ontwerp op correctheid controleren (synthesetools zijn ook niet foutloos). Quartus genereert een postsimulatiefile met timing (zie o.a. hoofdstuk 4 van de handleiding).

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.*

f) In vraag e) heb je waarschijnlijk entity `display` als component geïnstantieerd of heb je de functie `hex2display` gekopieerd. Maak nu een package met de naam `utilities`. In deze package komt alleen de header van de functie, in de package body de functie en vervolgens zet je boven het ontwerp waarin je deze functie wilt gebruiken "`use work.utilities.all;`".

> *N.B. op Blackboard (course materials) tref je de nodige bronnen aan hoe je met een package kunt werken. Maar met google gaat het natuurlijk ook (bv. http://www.asic-world.com/examples/vhdl/package.html)*

2. Teller

Gevraagd wordt een schakeling waarmee het aantal keren dat een knop wordt ingedrukt te tellen.

a) Maak gebruik van de hexadecimaal naar 7-segment converter uit de vorige opdracht.
b) De schakeling heeft twee 7-segment uitgangen. (Tip: het is niet handig om één teller te gebruiken die tot en met 99 gaat.)
c) Als meer dan 99 keer de knop is ingedrukt blijft het display op 99 staan.
d) De schakeling heeft een asynchrone reset (laag actief).
e) Realiseer je dat het indrukken van een knop vele klokperioden duurt (de klok heeft een frequentie van 50 MHz) en dit moet worden herkend als slechts eenmaal indrukken van de knop.
f) Geef eenvoudige VHDL testomgeving voor dit ontwerp
g) Realiseer je ontwerp op de FPGA. Telt de schakeling goed?

Er is al een entity beschrijving en constraintfile aanwezig.