

Opdracht week 5 en 6 – vier-cijfer-teller

Inleiding

In de digitale techniek worden tellers gebruikt voor tal van zaken. Zo kan je het gebruiken om een aantal gebeurtenissen te tellen, of om tijd te meten.

De opdracht voor deze week is het ontwerpen (en beschrijven) van een vier-cijferige teller, waarbij hiërarchie (structural VHDL) een belangrijke rol speelt. Bij deze opdracht wordt gebruik gemaakt van een prescaler (zie project), een omschakelbare BCD/HEX tellers (vier stuks) en de eerder ontworpen 7-segment decoder (ook vier stuks).

Ter voorbereiding op de opdracht is een Quartus II project beschikbaar dat de prescaler beschrijft. Dit project kan gecompileerd worden en in een DE0-ontwikkelbord geplaatst worden. Bestudeer de beschrijving van de prescaler en zorg dat je snapt hoe de werking ervan is. Tevens is een project geplaatst waarin al een begin is gemaakt met het beschrijven van de vier-cijfer-teller. Het geheel werkt nog niet. Lees de bijbehorende opdracht voor meer informatie.

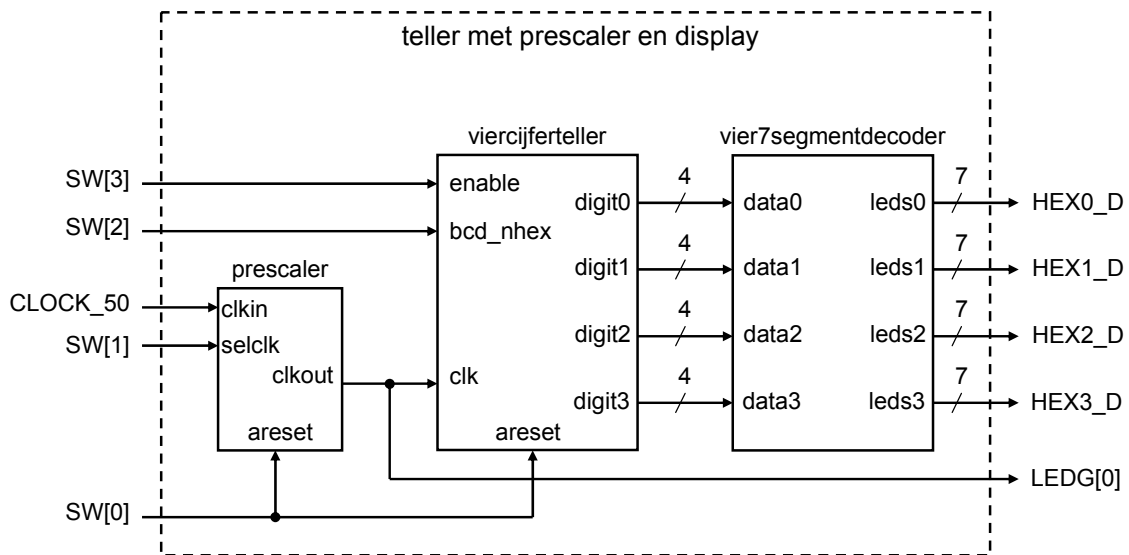
De vier-cijfer-teller als geheel werkt als volgt:

- De vier-cijfer-teller moet kunnen tellen in het BCD- en hexadecimale talstelsel.
- De vier-cijfer-teller bestaat uit vier identieke, gecascadeerde tellers. De tellers moeten dus als BCD- of hex-tellers ingesteld kunnen worden.
- De stand van de vier-cijfer-teller (BCD of hexadecimaal) moet op de vier 7-segment displays worden afgebeeld.
- De telfrequentie kan ingesteld worden op 1 Hz of 10 Hz. Aangezien de klokfrequentie van het DE0-board 50 MHz bedraagt, moet een prescaler gebruikt worden om de klokfrequentie te verlagen. Deze prescaler is gegeven.
- De vier-cijfer-teller moet herstart kunnen worden door middel van een asynchrone reset.
- De vier-cijfer-teller moet kunnen tellen of gestopt worden door middel van een enable-sigitaal.

In deze opdracht is het gebruik van structural VHDL een noodzaak. In onderstaande figuren wordt één en ander duidelijk.

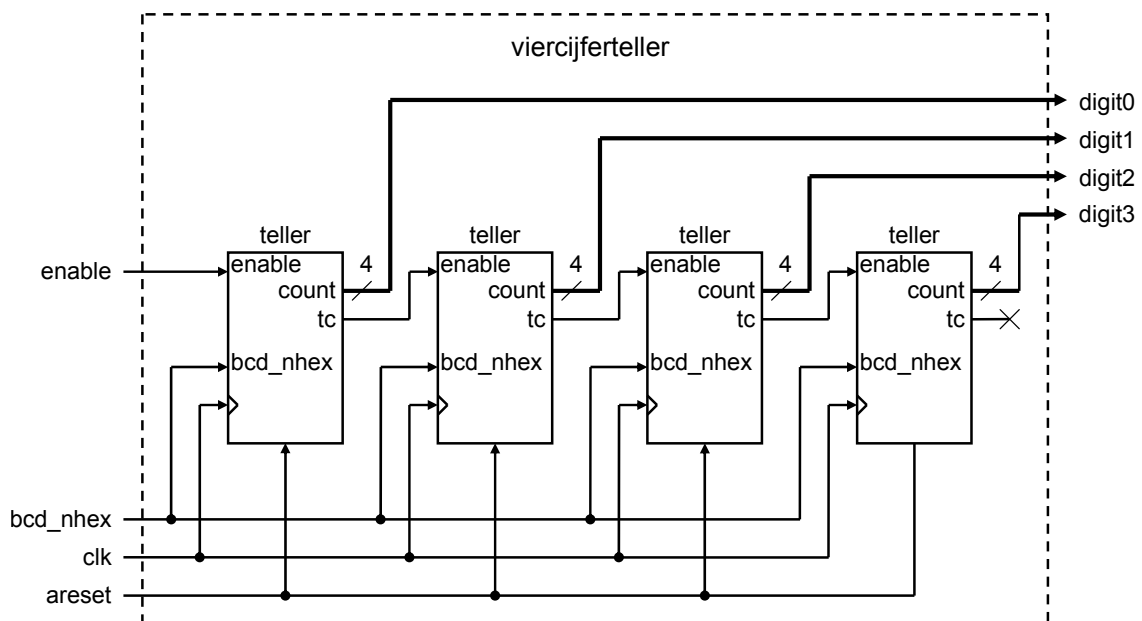
Op het hoogste niveau, de zogenoemde *top level*, zijn drie componenten te onderscheiden: een prescaler, een vier-cijfer-teller en een 7-segment display, zie figuur 1. De signaالنamen aan de rand komen overeen met de pinbenamingen van het DE0-bordje. Bij een aantal signalen is het aantal bits opgegeven, deze signalen moeten als vector worden gebruikt.

De vier-cijfer-teller en 7-segment display bestaan zelf weer uit componenten.



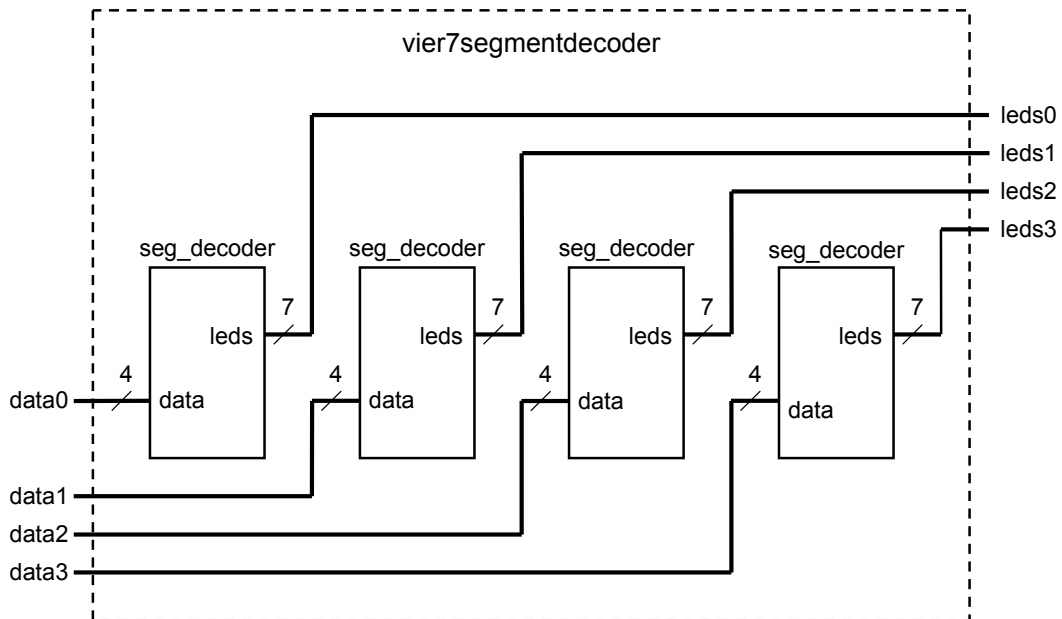
Figuur 1: Top-level blokdiagram met prescaler, teller en display

De viercijferteller (zie figuur 2) bestaat uit vier identieke, gecascadeerde, omschakelbare BCD/hex-tellers, met een *enable* en een *terminal count*. Merk op dat de terminal count van de meest significantie sectie niet naar buiten wordt gevoerd.



Figuur 2: Blokdiagram van de vier-cijfer-teller.

De vier7segmentdecoder (zie figuur 3) bestaat uit vier identieke 7-segment decoders. Deze zijn al tijdens een eerder practicum beschreven en getest.



Figuur 3: Blokdiagram van de vier 7-segment decoder.

Het is duidelijk dat dit een wat groter project is. Het kost aanzienlijk veel tijd om alles in VHDL te coderen en natuurlijk het project helemaal in te richten.

Op blackboard vind je een Quartus-project waarin het één en ander is ingericht. Je kan je nu concentreren op de VHDL-beschrijvingen van de tellers, van de `viercijferteller` en `vier7segmentdecoder`.

Opmerking: voor de tellers moeten unsigned signals/variables gebruikt worden.

Leerdoelen

De leerdoelen van deze opdracht zijn:

- Gebruik van structural VHDL voor het beschrijven van hiërarchieën.
- Scheiding structural VHDL en hardware-genererende VHDL.
- Gebruik van datatype `unsigned` voor rekenkundige bewerkingen.
- Opzetten en uitvoeren van een simulatie van VHDL-code.
- Beschrijven van het gedrag van een teller (BCD/hex, enable).

Opdrachten

De volgende opdrachten moeten gedaan worden:

- Haal het bestand `digse1_viercijfertellermetprescalerendisplay.zip` van BlackBoard en pak het uit in `H:\QUARTUS\DIGSE1`. Het zip-bestand bevat het Quartus-project `viercijfertellermetprescalerendisplay`. Pak het zip-bestand uit en plaats het in een map op de disk.
- Maak de VHDL-code voor `teller` en de 7-segmentsdecoder compleet,

- c) Maak de structural VHDL-beschrijvingen voor `viercijferteller`, `vier7segmentdecoder` en `viercijfertellermetprescalerendisplay` compleet.
- d) Schrijf een testbench en een ModelSim command file voor de **teller** (dus niet voor het hele ontwerp).
- e) Simuleer de teller.
- f) Compileer de VHDL-code en laadt het in het DE0-bordje. De pinaansluitingen zijn al ingevuld.
- g) Test het ontwerp op een DE0-bord.

Opmerkingen

In VHDL mogen uitgangen (port-beschrijvingen) alleen maar links van de *signal assignment operator* staan, d.w.z. dat er alleen waarden aan toegekend kunnen worden. Ze kunnen niet aan de rechterkant van de signal assignment operator staan want dan worden ze “gelezen”, d.w.z. ze dienen als ingang voor een of andere digitale schakeling. Je zal hiervoor dus een intern signaal of variabele moeten aanmaken die interne stand toekennen aan de uitgangen (noot: in VHDL-2008 mag dit weer wel).

De *terminal count* is een combinatorische uitgang en moet dus buiten de klokflank beschreven worden.

Zie hoofdstuk 8 van het dictaat voor meer informatie over tellers.