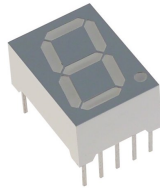


Opdracht week 3 – 7-segment display

Inleiding

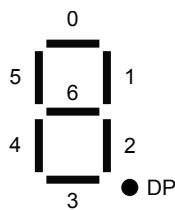
In de digitale techniek worden 7-segment displays gebruikt voor het afbeelden van decimale getallen. De display bestaat uit zeven segmenten (meestal uitgevoerd als leds) en een punt (ook uitgevoerd als led). De zeven segmenten zijn zo geconstrueerd dat ze samen de tien cijfers kunnen weergeven. Het is zelfs mogelijk een aantal letters weer te geven. Zie figuur 1 voor een praktische uitvoering.



Figuur 1: praktische uitvoering 7-segment display

Om tien verschillende cijfers weer te kunnen geven zijn vier bits nodig. De bitcombinaties 0000 t/m 1001 worden hiervoor gebruikt. Dit komt precies overeen met één BCD-cijfer. De bitcombinaties 1010 t/m 1111 worden niet gebruikt, maar zouden gebruikt kunnen worden om hexadecimale cijfers weer te geven.

Om cijfers weer te geven moeten de segmenten worden aangestuurd. Elk segment kan aan of uit staan, dus is een digitale schakeling te ontwerpen die dit doet. Dit is te realiseren met een schakeling die zeven (acht, als de punt wordt meegerekend) uitgangen heeft. Hiervoor wordt aan de segmenten een letter toegekend, zie figuur 2. Vervolgens kunnen de tien cijfers worden geconstrueerd, zie figuur 3.



Figuur 2: de segmenten van de display



Figuur 3: de tien cijfers op de display

De opdracht voor deze week is het invoeren van de schakeling en het beproeven van de 7-segment display.

Leerdoelen

De leerdoelen van deze opdracht zijn:

- Ontwerpen van een schakeling met poorten d.m.v. mintermen.
- Begrip binaire codering, binair tellen.

- Invoeren van het ontworpen schema.
- Simuleren van het ingevoerde schema.
- Testen van het ingevoerde schema.

Opdrachten

De volgende opdrachten moeten gedaan worden:

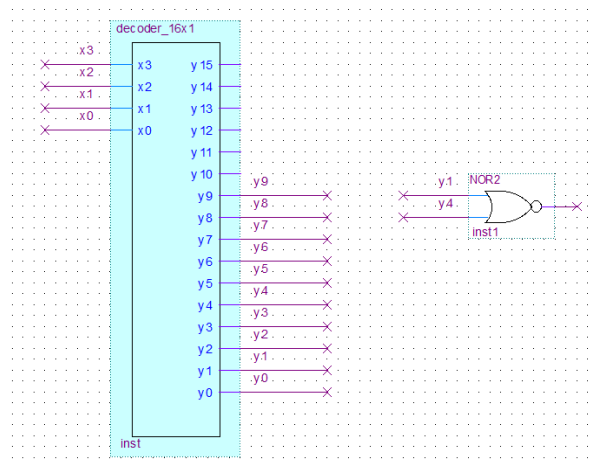
- a) Haal van BlackBoard het zip-bestand `seven_seg.zip` binnen en pak het uit in `H:\QUARTUS\INLDIG`. Het zip-bestand bevat het Quartus-project `seven_seg`.
- b) Maak het schema voor het 7-segment display compleet. Merk op dat er al een bestand met de `seven_seg.bdf` aangemaakt is.
- c) Simuleer het schema voor het 7-segment display met ModelSim. Gebruik het bestand `tb_seven_seg.do`, dit moet aangepast worden voor een goede werking.
- d) Synthetiseer en implementeer het schema en laadt het in het experimenteerbord.
- e) Test het ontwerp op een DE0-bord.

Opmerkingen

Een deel van het schema is al ingevoerd, onder andere het deel voor het genereren van de functies van y_0 t/m y_9 . Dit wordt gedaan door de deelschakeling `decoder_16x1`. Deze heeft vier ingangen en zestien uitgangen.

De y -uitgangen vertegenwoordigen de bijbehorende *mintermen*. Zo is y_0 logisch 1 als de ingangen x_3 , x_2 , x_1 en x_0 allemaal logisch 0 zijn (minterm 0). Alle andere y -uitgangen zijn dan logisch 0. Er is dus op elk moment slechts één y -uitgang logisch 1.

Het invoeren van de complete schakeling zal nog wel problemen geven, er moeten veel poorten worden ingevoerd en nog veel meer verbindingen (wires). Dat levert een onwerkbaar schema op. Het is mogelijk om wires te koppelen zonder ze fysiek te verbinden door ze dezelfde naam te geven. In figuur 4 is te zien dat de signalen y_1 van de decoder verbonden is met signaal y_1 van de NOR-poort. Dit geldt ook voor y_4 .



Figuur 4: de decoder met een poortje

De naamgeving gaat eenvoudig: selecteer een wire, klik rechtermuisknop en klik dan op *Properties*. Onder het tabblad *General* kan je een naam invullen.

Merk op dat één kant van de wires niet verbonden zijn, dat is te zien aan het kruisje.

Let op: de leds van de display zijn laag actief, een logisch 0 zorgt ervoor dat de led gaat branden.

Voor de ingangen moeten weer de schakelaars SW3 t/m SW0 gebruikt worden waarbij SW3 het meest significante bit voorstelt. De uitgangen hebben de namen HEX0_D0 t/m HEX0_D6 waarbij HEX0_D0 gelijk staat aan segment A en HEX0_D6 gelijk staat aan segment G. Zie voor een volledig plaatje bijlage B van de tutorial.