

Practicum 2.1: Zeven segment displays

In deze opdracht maak je een driver voor het zeven segment display.

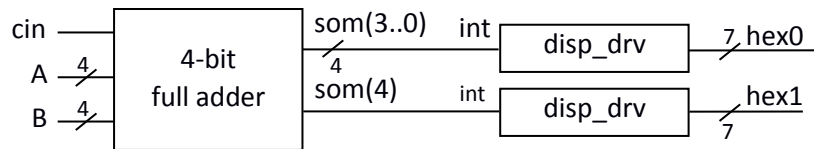
1. zoek uit hoe je de 7 segments displays kan aansturen (zie hoofdstuk 4.3 van *DE0_User_manual_2012.pdf*).
2. start een nieuw project en import (*Assignments – Import Assignments...*) de *DE0_pin_assignments.qsf* file.
3. voeg een VHDL file toe (bijvoorbeeld “disp_drv.vhd”) met de volgende entity:

```
-- 7 segment decoder
-- input: 4-bit number 0 to F
-- output: 7 led segments
entity disp_drv is
    port ( C : in std_logic_vector(3 downto 0);
          display : out std_logic_vector(6 downto 0)
        );
end entity disp_drv;
```
4. schrijf een architecture (genaamd “with_select”) voor deze entity die gebruik maakt van een “with select when” constructie. Let op: een segment licht op wanneer er een 0 naar het segment wordt geschreven;
5. simuleer het ontwerp met de waveform monitor (of een aparte testbench).
6. creëer een symbolfile voor deze entity/file (gebruik create/update in het file menu).
7. voeg een new Block Diagram/Schematic file toe aan het project en gebruik het symbool om het eerste zeven segment display *HEX0(0 TO 6)* aan te sturen met schakelaar 0 t/m 3 (*SW(0 TO 3)*).
8. compileer het ontwerp.
9. programmeer en test het ontwerp met het DE0 board. Een binair getal op de schakelaars moet leiden tot het corresponderende hexadecimale getal op het display.
10. verander het ontwerp zodat de schakelaars 0-3 het display *HEX0* aansturen en zodat schakelaars 4-7 *HEX1* aansturen. Met schakelaars 8 en 9 kan je *HEX2* aansturen (alleen voor cijfer 0..3).
11. compileer, programmeer en test.

Practicum 2.2: full adder

12. Start een nieuw project, voeg een VHDL file toe en schrijf/implementeer een entity en architecture (dataflow) voor een 1-bit full-adder, gebruik `std_logic` voor de ports.
13. Voeg een VHDL file toe en schrijf een 4-bit ripple full adder (entity en architecture). Gebruik port mapping om 4 stuks 1-bit full adder te instantiëren en gebruik een `std_logic_vector` als input en output;
14. Voeg een VHDL file toe die een 4-bit ripple full adder component en twee `disp_drv` componenten gebruikt om de som van twee 4-bits getallen (en een *carry-in*) op HEX1 en HEX0 te tonen. Gebruik port mapping voor deze VHDL file. Sluit de input aan op de schakelaars volgens onderstaande tabel:

Cin	A	B
sw0	sw4-sw1	sw8-sw5



15. compileer, programmeer en test op het DE0 board;