

## Practicum 1.1

In dit practicum koppel je de schakelaars aan de leds.

1. start een nieuw project (bijvoorbeeld "leds") en include de *DE0\_pin\_assignments.csv* file
2. voeg een nieuw VHDL bestand toe en gebruik de volgende code om de standaard library te includen:

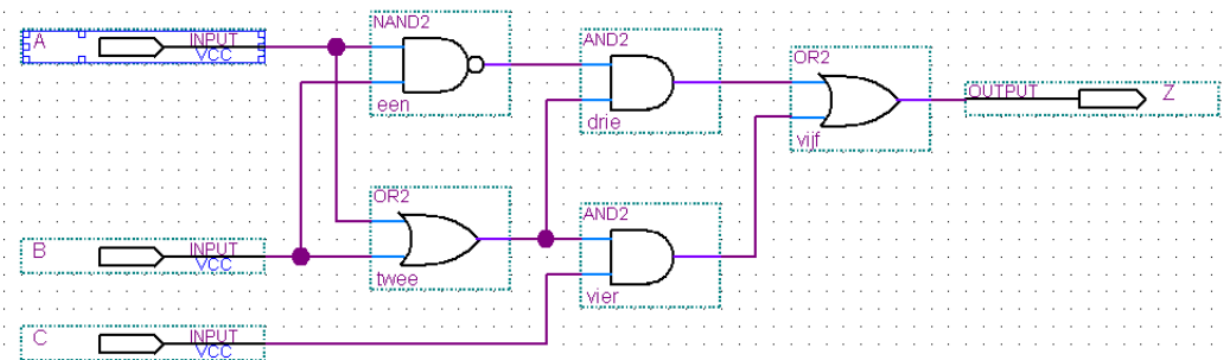
```
library ieee;
use ieee.std_logic_1164.all;
```

3. schrijf een entity met **sw** als input en **ledg** als output, beide *std\_logic\_vectors* met breedte 10.
4. schrijf een architecture met een enkel VHDL statement: `ledg <= sw` (wat betekent dit statement precies?)
5. compile en test het ontwerp op het DE0 board.

## Practicum 1.2

In dit practicum implementeer je een combinatorische schakeling door deze te tekenen in een block diagram.

1. start een nieuw project en include de pin assignments, of gebruik het bovenstaande project;
2. voeg een nieuw *Block Diagram/Schematic File* (bijvoorbeeld *prac12*) en teken het onderstaande schema waarbij je gebruik maakt van de standaard symbolen die je kunt vinden in Altera libraries – primitives – logic.
3. Genereer pins voor de inputs/outputs van de symbolen (rechter muis klik menu) en ken schakelaars `sw[2]`, `sw[1]` en `sw[0]` toe aan `a`, `b` en `c` en `ledg[0]` aan `z`.
4. compileer het ontwerp;
5. simuleer het ontwerp, maak een geschikte wave file om alle mogelijkheden te testen;
6. programmeer en test het ontwerp op het DE0 board;
7. voeg op verschillende punten in het schema extra `ledg` outputs toe om alle tussenliggende resultaten te tonen;
8. compileer, programmeer en test het ontwerp.



## Practicum 1.3

In dit practicum gebruik je data flow voor een implementatie van een combinatorische schakeling.

1. Maak een waarheidstabel voor het bovenstaande schema;
2. Open het voorgaande project;
3. Voeg een nieuw VHDL bestand (*bijvoorbeeld prac13.vhdl*), include de onderstaande entity en schrijft een architecture die het bovenstaande schema beschrijft. Gebruik een enkel VHDL statement;

```
entity lab13 is
  port (sw : in std_logic_vector(2 downto 0);
        ledg : out std_logic_vector(0 downto 0)
  );
end lab13;
```

4. compileer het ontwerp, Pas op: maak de nieuwe file de *top-level entity*;
5. simuleer het ontwerp met dezelfde wave file. Krijg je dezelfde resultaten?
6. programmeer en test het ontwerp op het DE0 board;
7. gebruik de menu optie: *tools - netlist viewers - RTL viewer* om te zien hoe Quartus het ontwerp synthetiseert. Hoe verschilt het van het originele schema?