

Opdracht week 3 – schuifregister

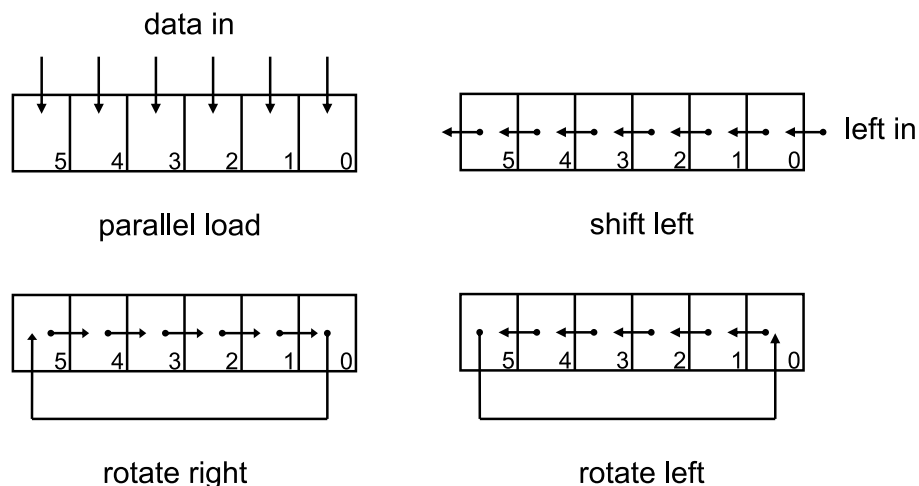
Inleiding

In de digitale techniek worden schuifregisters gebruikt voor tal van zaken. Zo kan je het gebruiken voor seriële communicatie (Ethernet, RS232, USB, SATA) of vermenigvuldigen met en delen door 2.

De opdracht voor deze week is het beschrijven en beproeven van een 6-bit schuifregister met de volgende werkmodi:

- één bit naar links schuiven en aanvullen met een extern databit.
- één bit linksom roteren
- één bit rechtsom roteren
- parallel laden van het register met externe data

De vier werkmodi zijn uitgebeeld in figuur 1. Merk op dat de parallele uitgangen niet zijn getekend.



Figuur 1: De vier werkmodi van het schuifregister

Natuurlijk heeft het schuifregister een klokingang. De vier werkmodi worden aangeboden als een 2-bits waarde (vector). De externe data wordt aangeboden als een 6-bits binaire waarde (vector). De stand (inhoud) van het schuifregister wordt zichtbaar gemaakt via de uitgangen als een 6-bits vector. Het schuifregister moet een asynchrone reset hebben.

Een schuifregister bestaat uit een serieschakeling van D-flipflops. Dat houdt in dat alle acties zoals hierboven beschreven moeten worden uitgevoerd onder klokflanksturing.

De opdracht voor deze week is het ontwerpen en beproeven van een 6-bit schuifregister.

Leerdoelen

De leerdoelen van deze opdracht zijn:

- Opstellen van de juiste entity-beschrijving.
- Ontwerpen/beschrijven van kloksynchrone logica
 - Beschrijven van een asynchrone reset.
 - Beschrijven van de klokflank (opgaand of neergaand).
 - Beschrijven diverse acties die op de klokflank uitgevoerd moeten worden.

Opdrachten

De volgende opdrachten moeten gedaan worden:

- Stel een lijst op met alle in- en uitgangen van het te ontwerpen schuifregister. Geef tevens aan uit hoeveel bits elk signaal bestaat.
- Voer de beschrijving van het schuifregister in. De signalen zijn van het type `std_logic` of `std_logic_vector`, tenzij anders aangegeven.
- Voer de juiste pinkoppelingen in. De pinaansluitingen staan vermeld in bijlage B van de tutorial.
- Synthetiseer en implementeer de VHDL-code en laadt het in het experimenteerbord.
- Test het ontwerp op een DE0-bord.
- Optioneel*: Breidt de mogelijkheden van het schuifregister uit met een *hold* en *shift right* mogelijkheid. Hiervoor is een extra stuursignaal nodig.

Noot: bij deze opdracht wordt nog niet gesimuleerd.

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 (de interne stand van het schuifregister) en die interne stand toekennen aan de uitgangen.

Schuiven in VHDL kan heel makkelijk door het gebruik van *vector slices* en de *concatenation operator*. Zo kan je naar links schuiven met

```
reg <= reg(6 downto 0) & '0';
```

als signaal `reg` een 8-bits schuifregister is. Er wordt hier een 0 ingeschoven. Zie hoofdstuk 8 van het dictaat voor meer informatie over schuifregisters.

De klokgenerator op het DE0-board loopt op 50 MHz. Dat is veel te snel voor mensen om goed te kunnen testen. Verbind daarom (via de Pin Planner) de klokingang van de VHDL-beschrijving met `BUTTON2`. Nu kan je een kloksignaal aanbieden door op het knopje te drukken.