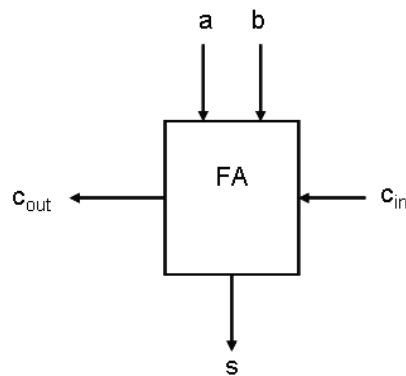


Opdracht week 5 – 4-bit Full Adder

Inleiding

In de digitale techniek worden getallen opgeslagen in het binaire talstelsel. Rekenschakelingen zijn eenvoudig te ontwerpen. Zo kan een ontwerp worden gemaakt voor een full adder voor één bit. Zie figuur 1.



Figuur 1: 1-bit Full Adder

Meerdere van deze 1-bit Full Adders zijn te cascaderen tot grotere optellers, bijvoorbeeld 4-bit. Het ontwerp komt overeen met de manier waarop wij optellen, namelijk kolomsgewijs.

Deze opdracht bevat het ontwerpen en testen van zowel een 1-bit Full Adder als een 4-bit Full Adder. Eerst wordt de 1-bit Full Adder ontworpen en getest, daarna de 4-bit Full Adder.

Leerdoelen

De leerdoelen van deze opdracht zijn:

- Ontwerpen van een schakeling met poorten met hiërarchieën.
- Instellen van de juiste Top Level Entity.
- Invoeren, simuleren en testen van de ontworpen schakeling.

Opdrachten

De volgende opdrachten moeten gedaan worden. Eerst moet het project ingericht worden.

- a) Haal van BlackBoard het zip-bestand `full_adder.zip` binnen en pak het uit in `H:\QUARTUS\INLDIG`. Het zip-bestand bevat het Quartus-project `full_adder`.

Vervolgens moet een 1-bit Full Adder ontworpen en getest worden.

- b) Maak een nieuw Block Design File aan. Voer hier het schema van een 1-bit Full Adder in. Gebruik als ingangen a , b en c_{in} en als uitgangen c_{out} en s (zie figuur 1). Sla dit bestand op als `fa_onebit.bdf`.
- c) Simuleer het schema van de 1-bit Full Adder met ModelSim en controleer de goede werking. De bijbehorende simulatie-script heet `tb_fa_onebit.do`.
- d) Genereer een symbool (Block Symbol File) van de 1-bit Full Adder. Zie tutorial.

Nu moet een 4-bit Full Adder geconstrueerd worden uit losse 1-bit Full Adders.

- e) Maak een nieuw Block Design File aan. Voer hier het schema van de 4-bit Full Adder. Sla dit bestand op als `full_adder.bdf`. De mappings van de ingangen en uitgangen staan in onderstaande tabel:

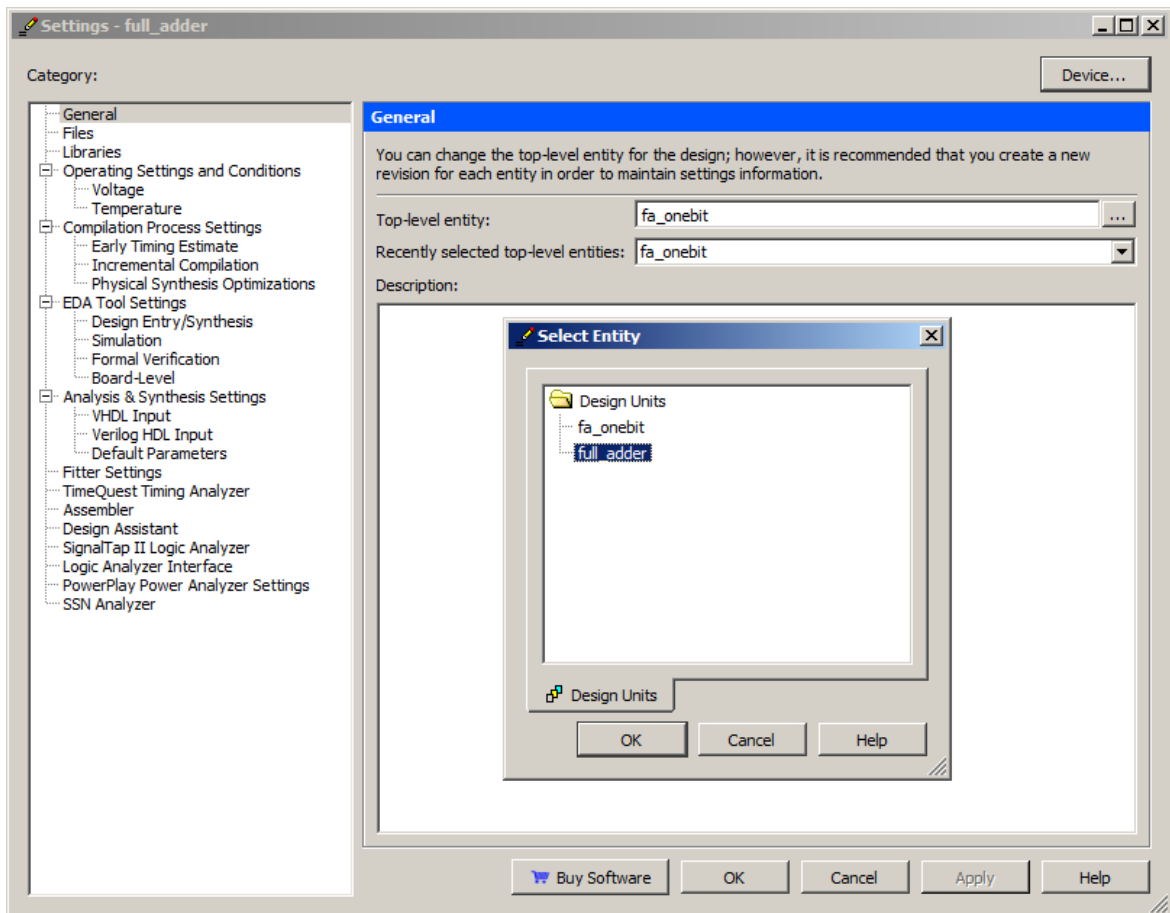
B3	->	SW8	A3	->	SW4	C0	->	SW0
B2	->	SW7	A2	->	SW3			
B1	->	SW6	A1	->	SW2			
B0	->	SW5	A0	->	SW1			
S3	->	LEDG3	C4	->	LEDG4			
S2	->	LEDG2						
S1	->	LEDG1						
S0	->	LEDG0						

(B3 t/m B0 zijn de bits van getal B, A3 t/m A0 zijn de bits van getal A, C0 in de carry-in. S3 t/m S0 zijn de som-bits van uitkomst S, C4 is de uitgaande carry van de 4-bit Full Adder).

- f) Synthetiseer de 4-bit Full Adder middels Start Analysis & Synthesis (Ctrl-K). Hierdoor wordt een *entity* `full_adder` aangemaakt.

Nu moet de 4-bit Full Adder gesimuleerd worden (en later ook geïmplementeerd). Hiervoor moet eerst de juiste *top level entity* geselecteerd worden (dat is bij opdracht f) namelijk nog `fa_onebit`).

- g) Selecteer in het menu Assignments->Settings het tabblad General. Selecteer vervolgens (rechts) bij de regel Top Level Entity de module `full_adder`. Zie figuur 2.
- h) Simuleer de 4-bit Full Adder met behulp van Modelsim. De bijbehorende simulatiescript heet `tb_full_adder.do`.
- i) Test de schakeling met behulp van het DE0-ontwikkelfbord. Voer de volgende berekeningen uit:
 - $+15 + +15$
 - $-1 + -1$
 - $-6 + +6$
 - $+10 + -6$



Figuur 2: *Veranderen van Top Level Entity Name*