

Laboration i digital konstruktion I: Lab 3

Bengt Oelmann, Mitthögskolan, ITE

1.0 Konstruktion och implementering av en parameteriserbar ALU

1.1 Syfte

Den här laborationen visar hur man kan konstruera parameteriserbara moduler. Med en parameter kan man sätta bitbredden för operanderna till ALU:n

1.2 Bakgrund

I en mikroprocessor är ALU:n (Aritmetisk, Logisk Enhet) den enhet som utför alla aritmetiska och logiska operationer. ALU:n kan utföra ett antal olika operationer. Vilken operation som utförs bestäms av den instruktion som för tillfället exekveras.

1.3 Förberedelser

```
cd ~/VHDL
mkdir lab3
cd lab3
qhlib work
cp ~kurser/digkon1/lab/VHDL_kod/ALU_lab.vhdl
```

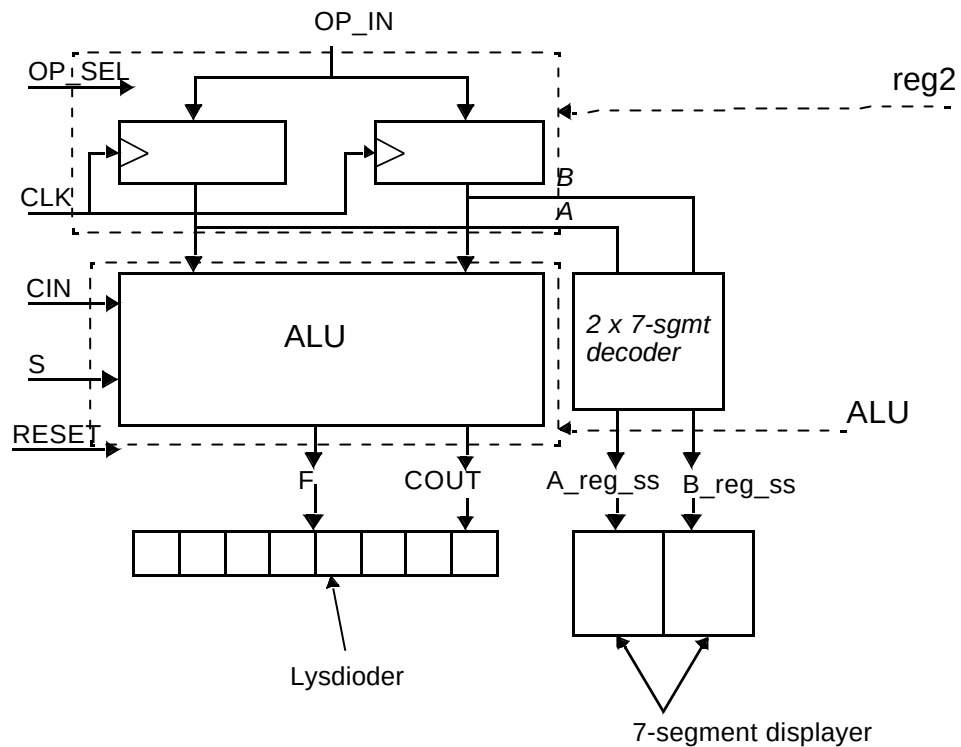
I *ALU_lab.vhdl* finns grunden till VHDL beskrivningen som ska kompletteras samt anpassningar för att kunna använda ALU:n på prototypkortet.

1.4 Funktionell beskrivning av ALU:n

ALU:n i den här laborationen kan utföra 8 operationer på två operand med N-bitars ordbredd. N är en parameter som kan sättas av användaren. Operanderna *A* och *B* läses en åt gången in i var sitt register. Med signalen *OP_SEL* väljs vilket av registren som ska uppdateras, signalen *OP_IN* laddas in i ett av registren. Funktionen som ska utföras sätts med signalen *S*.

Tabell 1: ALU funktioner

S	F
000	00 ... 0
001	$B - A - 1 + \text{CIN}$
010	$A - B - 1 + \text{CIN}$
011	$A + B + \text{CIN}$
100	$A \text{ xor } B$
101	$A \text{ and } B$
110	$A \text{ or } B$
111	11 ... 1



Figuren ovan beskriver strukturen i `ALU_lab.vhdl` som består av instantiering av ALU, reg2 och två stycken 7-segmentavkodare samt kopplingar signal till pinnar på FPGA:n.

1.5 Utförande

1.5.1 VHDL-beskrivning av ALU

Skriv funktionen för ALU:n enligt tabell 1. Signalgränssnittet står beskrivet i `ALU_lab.vhdl`. Koden ska skrivas så att den kan hantera godtyckliga bitbredder styrd av parametern `N`.

1.5.2 VHDL-beskrivning av reg2

Skriv funktionen för registret. Signalgränssnittet står beskrivet i `ALU_lab.vhdl`. Koden ska skrivas så att den kan hantera godtyckliga bitbredder styrd av parametern `N`.

1.5.3 Verifiera med simulering

Sätt parametern `N = 32` och kompilera.

Simulera och kontrollera att samtliga funktioner fungerar som tabell 1 specificerar.

1.5.4 Implementera i FPGA

Sätt parametern `N = 3`.

Läs in filerna *sevenseg_decoder.vhdl* och *ALU_lab.vhdl* i Leonardo och syntetisera så att en XNF-fil skapas.

Kom ihåg att göra följande inställning i Leonardo före syntes:

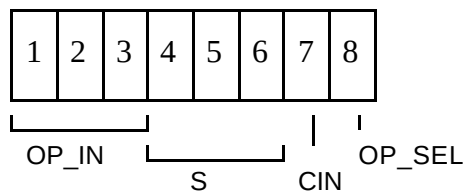
Under fliken *Techn...* FPGA - Xilinx - 4005ePC84

Välj *Advanced settings* längst ner på sidan ...

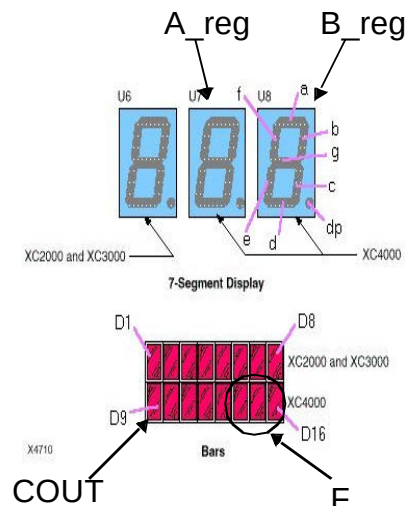
Ta bort kryssen för *Global Buffers*

1.5.5 Ladda ner till prototypkortet

Testa ALU:n på kortet. Indata matas in från följande switchar på SW-3:



Resultaten visas på 7-segment displayen och på lysdioderna:



1.6 Tips

Addition och subtraktion kan göras med + och - tecken om man har *ieee.std_logic_1164.all* inkluderat. Addition och subtraktion görs med signaler/variabler med datatypen *std_logic_vector* men inte med *std_logic*. En konstant bit som '1' skrivs som bitvektor "1". Signalen *CIN* är av *std_logic*. Den skrivs som *std_logic_vector* genom ("&CIN"), d.v.s sammanslagning av en tom vektor med en bit ger en vektor.

1.7 Redovisning

VHDL kod för ALU:n och reg2 ska visas för lab. handledaren

Simulering av 32-bitars varianten ska demonstreras för lab. handledaren

Implementeringen av 3-bitars varianten ska demonstreras för lab. handledaren.

Förklara hur *generics* används.