

# Laboration i digital konstruktion I: Lab 2

## 1.0 Konstruktion och implementering av FSM

### 1.1 Syfte

Den här laborationen visar hur man implementerar en tillståndsmaskin i en FPGA utifrån en specifikation i form av tillståndsdiagram.

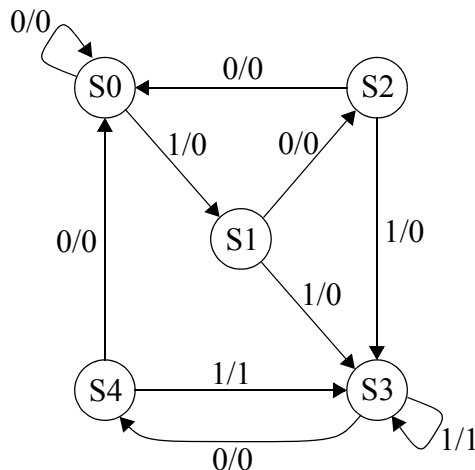
### 1.2 Förberedelser

```
cd ~/VHDL
mkdir lab2
cd lab2
qhlib work
cp ~kurser/digkon1/lab/VHDL_kod/fsm3_tb.vhdl .
cp ~kurser/digkon1/lab/VHDL_kod/lab_controller.vhdl .
```

### 1.3 Utförande

#### 1.3.1 VHDL-beskrivning av tillståndsmaskin

Skriv en syntetiserbar kod för följande tillståndsmaskin:



Format: insignal/utsignal

Insignal: A

Utsignal: Z

Tillståndskodning:

S0 : 000

S1 : 001

S2 : 010

S3 : 011

S4 : 100

Modul-gränssnittet ska vara enligt följande:

```
entity fsm3 is
  port (
    clk, a, reset : in std_logic;
    s             : out std_logic_vector(2 downto 0);
    z             : out std_logic);
end fsm3;
```

utgången s är tillståndet.

#### 1.3.2 Verifiera

Kompilera *fsm3.vhdl*

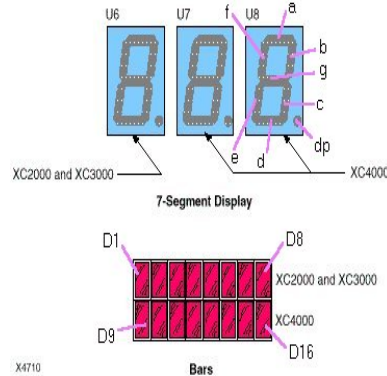
Kompilera testbänken *fsm3\_tb.vhdl*

Simulera med testbänken och kontrollera att fsm3 fungerar som den ska.

## Laboration i digital konstruktion I: Lab 2

### 1.3.3 Anpassa för FPGA prototypkort

För att kunna se hur tillståndsmaskinen fungerar måste vi koppla utgångarna till lysdioder och 7-segment displayer samt ingångarna till switchar.

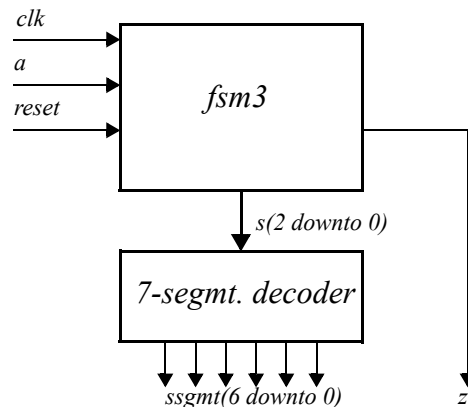


Tillståndet  $s$  ska visas på 7-segment displayen på decimalform, d.v.s då man är i tillståndet  $S_0$  visas 0,  $S_1$  visas 1, ...,  $S_4$  visas 4. Utsignalen  $z$  visas på lysdioden D9.

För att kunna visa decimala tal på 7-segment displayen så måste man ha en kodare som tänder rätt segment för att visa en viss decimal siffra.

Insignalen  $a$  tas från switchen SW3-1 och klocksignalen tas från tryckknappen *SPARE*.

Anpassningen som måste göras ser ut som följer:



I VHDL-koden *lab\_controller.vhdl* finns kopplingen från in/utsignal till/från entiteten från/till switchar och lysdioder på kortet. Med *attribute pin\_number* styrs detta. Här finns också instantiering av 7-segment avkodaren och *fsm3*, som Ni har skrivit.

### 1.3.4 Syntetisera VHDL *lab\_controller*

#### 1.3.4.1 Följande tre filer ska läsas in i Leonardo:

- *fsm3.vhdl*
- *sevenseg\_decoder.vhdl*
- *lab\_controller.vhdl*

#### 1.3.4.2 Ändra inställningar i Leonardo

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

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

Ta bort kryssen för *Global Buffers*

Välj från meny-raden: *Flows* → *Synthesis Wizard*

*Encoding Style* : *Binary*

*Device Settings* : är redan ifylld, men kontrollera att det är OK.

*Global* : *Specify Clock Frequency* : *10 MHz*

*Output File* : *Format XNF*, kontrollera också att filnamnet är OK.

### 1.3.4.3 Ladda ner konstruktionen till kortet

Från *Leonardo* har vi fått en *XNF*-fil som verktyget *xilinx* utgår från för *place & route* och därefter laddar ner till kortet.

Starta *xilinx* och öppna ett nytt projekt med *lab\_controller.xnf* som in-fil.

Välj från meny-raden : *Design* → *Implement*

Om allt har gått bra så står det *Implemented, OK* i konstruktionsträdet. Som resultat får man en fil, *lab\_controller.bit* som laddas ner på kortet.

Slå på spänningen till kortet.

Välj från meny-raden : *Tools* → *Hardware Debugger*

Välj i *Hardware Debugger* meny-rad : *Download* → *Download Design*

Efter ett 10-tal sekunder är nerladdningen färdig och tillståndsmaskinen är på plats och ska fungera.

## 1.4 Redovisning

VHDL-kod för tillståndsmaskinen ska visas för lab. handledaren

FPGA implementationen ska demonstreras för lab. handledaren