LABORATION 3

LEJON

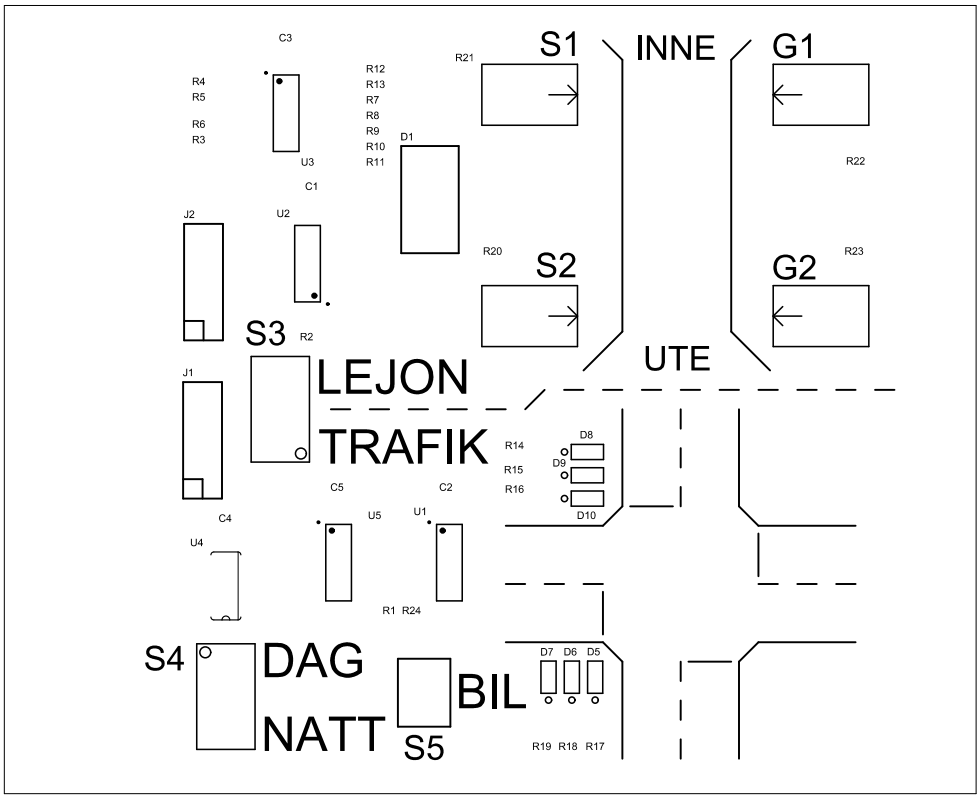
Laborationens syfte

Syftet med laborationen är dels att lära känna laborationsutrustningen och dels att få en uppfattning om hur en digital konstruktion är uppbyggd, i detta fallet med hjälp av en FPGA monterad på Nexys-4 kortet (se laboration 1). Uppgiften skall löses med sekvensnät av Mealy-typ. Uppgiften består av att hantera flera lejon (max 9) i vår lejonbur med tillhörande hage. Till vår hjälp har vi ett externt labbkort som beskrivs nedan.

OBS! Det finns ett antal hemuppgifter som skall vara avklarade och uppvisade för att få lov att genomföra laborationen!

.....
Grupp
.....
Namn
.....
Godkänd

Labbkortet som är anslutet till Nexys4 består av en trafikkorsning och en lejonbur, se figur 3.1. Med strömställaren S3 väljer man om kortet skall fungera som en trafikkorsning (TRAFIK) eller lejonbur (LEJON). Det finns två kontakter på labbkortet märkta `JA` och `JB`. Dessa är anslutna till motsvarande kontakter på Nexys4.



Figur 3.1: Labbkort

`JB` är signaler från Nexys4 till labbkortet enligt tabellen nedan

JB	**Strömställare S3 i läge trafik**	**Strömställare S3 i läge lejon**
D0 (LSB)	Rött ljus huvudgata	D0 (LSB) till 7-segmentsavkodare
D1	Gult ljus huvudgata	D1 till 7-segmentsavkodare
D2	Grönt ljus huvudgata	D2 7-segmentsavkodare
D3	Rött ljus sidogata	D3 (MSB) 7-segmentsavkodare
D4	Gult ljus sidogata	
D5	Grönt ljus sidogata	
D6		
D7 (MSB)		

`JA` är signaler från labbkort till Nexys4 enligt nedan (oberoende av strömställaren S3):

JA	**Funktion**
D0 (LSB)	0 = Dag, 1 = Natt (S4)
D1	1 = Bil på sidogata (S5)
D2	1 = Lejon framför givare G1
D3	1 = Lejon framför givare G2
D4	0
D5	0
D6	0
D7 (MSB)	0

Observera att i denna lab är det endast läge lejon som gäller.

Uppgift 3.1. Lejonburen med ett lejon, teori

I en djurpark finns en avdelning för lejon. Den består av två delar, dels en bur där lejonet kan gå in och dels en inhägnad som är en stor öppen plats där lejonet kan ströva runt. Inhägnaden är kuperad vilket gör den svåröverskådlig. Eftersom djurskötaren vill kunna städa inhägnaden medan lejonet är i buren behöves ett system där en lampa, `Fara`, lyser om något lejon är ute ur buren.

Vid passagen mellan buren och inhägnaden finns två givare, $\mathtt{G}_1$ och $\mathtt{G}_2$. Dessa är realiserade med fotoceller som är placerade med någon decimeters mellanrum och på lagom höjd. Se figur 3.1. Lejonet är ca två meter långa och kan inte vända eller stanna i gången.

Då en ljusstråle bryts ger givaren en logisk etta medan den ger en logisk nolla om den inte bryts. Det innebär att när ett lejon går från buren till inhägnaden ger givarna följande sekvens:

$$(\mathtt{G}_1, \mathtt{G}_2) : 00 \to 10 \to 11 \to 01 \to 00$$

På samma sätt blir sekvensen när ett lejon går från inhägnaden till buren:

$$(\mathtt{G}_1, \mathtt{G}_2) : 00 \to 01 \to 11 \to 10 \to 00$$

Hemuppgift 3.1.1

Rita en graf för ett minimalt sekvensnät som kan hålla reda på *ett* lejon. Insignalerna skall vara signalerna från givarna $\mathtt{G}_1$ och $\mathtt{G}_2$ och utsignalen skall vara `Fara`. Lampan skall vara tänd (logisk etta) då lejonet är i gången eller ute i inhägnaden. Hur många tillstånd behövs?
Denna uppgiften ska inte implementeras på kortet.

Slut på uppgift 3.1

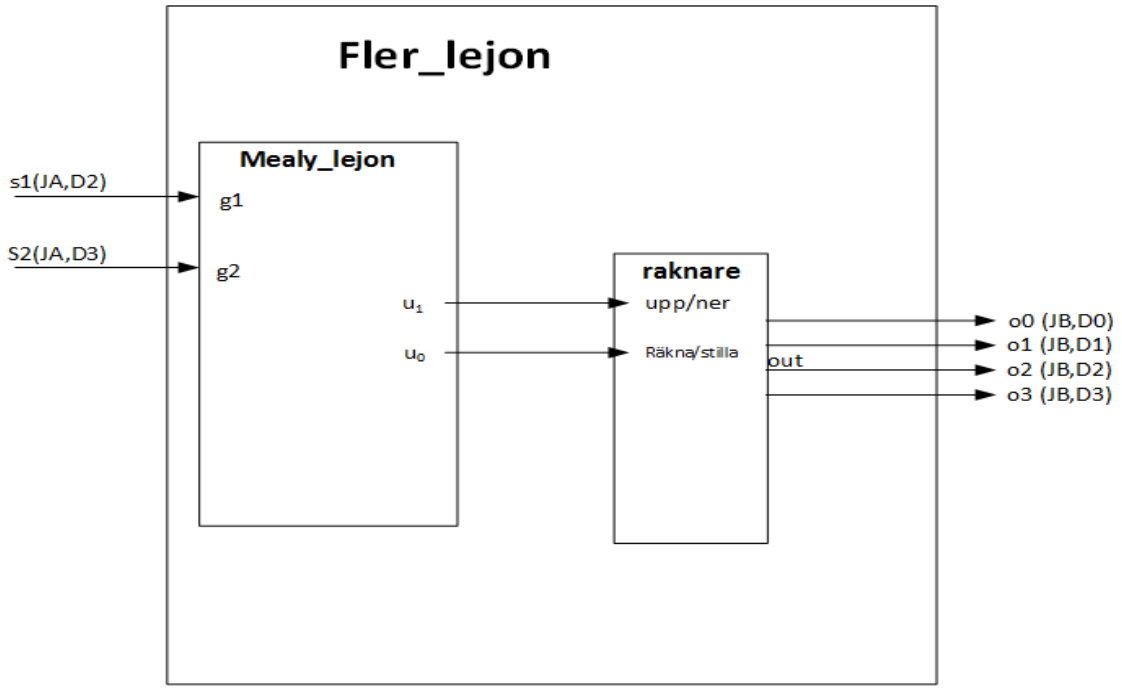
Uppgift 3.2. Lejonbur med flera lejon, VHDL

Uppgifter är som den tidigare beskrivna, men nu har vi flera lejon på avdelningen. Detta betyder att lampan fara ersätts med en 7-segments display som visar antalet lejon ute i hagen. Insignalerna är samma som i föregående uppgift medan utsignalerna är nu två signaler som styr en räknare (vars beteende beskrivs i en senare hemuppgift) Samma förutsättningar som tidigare men nu blir uppgiften:

Hemuppgift 3.2.1

Rita en graf för ett minimalt sekvensnät (Mealy) som kan hålla reda på *flera* lejon (max 9) som befinner sig ute i hagen. Hur många tillstånds behövs nu? Insignalerna är som tidigare g_1 och g_2, medan utsignalerna u_1, u_0 är räknevillkor till räknaren.

Utgå från tillståndsdiagrammet och beskriv i VHDL hur maskinen skall bete sig. Blockschemat bör vara enligt figur 3.2



Figur 3.2: Blockschema

1. Skapa ett nytt projekt i katalogen `C:\Users\stilID\program` (skapa katalogen program om den inte finns) och importera tre källfilerna som ligger under `S:\Courses\eit\EITA15\Laborationer\Lab3`. *Fler_lejon.vhd*, *mealy_lejon.vhd* och *raknare.vhd* innehåller skalet till projektet. I filen *Fler_lejon.vhd* skall inget ändras medan de två övriga behöver kompletteras med vad de skall göra. **Ändra inga signalnamn!**

Hemuppgift 3.2.2

Utgå från tillståndsdiagrammet och beskriv i VHDL hur maskinen skall bete sig (mealy_lejon.vhd). Under *architecture* skriver du den koden som du har kommit fram till. koden skall beskriva en FSM (finite state machin) av typ Mealy. Utsignalerna från FSM-maskinen är två (med villkoren enligt tabellen nedan) och insignalerna är två (g1 och g2)

2. Då vi behöver hålla reda på antalet lejon som befinner sig ute måste en räknare inkluderas i konstruktionen. Filen *raknare.vhd* i projektet skall kompletteras med lämplig VHDL-kod under *architecture*. Detta blir en fyra-bitars räknare med räknevillkoren upp eller ner enligt nedanstående tabell.

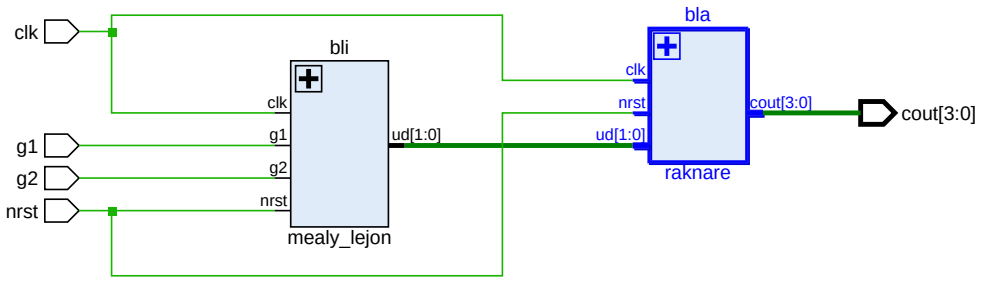
Hemuppgift 3.2.3

Fyll i kodern du behöver för att *raknare.vhd* skall bete sig som en 4-bitars (BCD-kodad) upp eller ner räknare. Insignalerna är u1, u0 och utsignalerna cout(0) ... cout(3) ska driva en 7-segments avkodare som finns på labbkortet. Tänk på att du behöver inte göra en tillståndsmaskin då VHDL språket löser en räknare enkelt.

u1 (msb)	**u0 (lsb)**	**funktion**
-	0	stå stilla
0	1	räkna upp
1	1	räkna ner

Tänk på att biblioteken *use ieee.std_logic_unsigned.all* och *use IEEE.NUMERIC_STD.ALL* kommer att behövas om ni använder aritmetiska funktioner.

3. Nu när designen är nästan klar är det lämpligt att göra en RTL analys av projektet. Du bör ha ett schema enligt figur 3.3



Figur 3.3: RTLschema

4. För att kunna verifiera att de olika delarna fungerar som det är tänkt, är det lämpligt att skapa filer som kan simulera de olika delarna. I katalogen `S:\Courses\eit\`

`EITA15\Laborationer\Lab3` finns det två testfiler *mealy_lejon_tb.vhd* och *raknare_tb.vhd*. Dessa innehåller skalet samt en början på hur man kan skriva sin testbench. I processen stim_proc finns en antydan till hur man kan skriva sina tester. Observera att detta är ett mycket enkelt exempel på VHDL-kod som på inga sätt beskrivet att det är så här man skall skriva. Ett enkelt sätt som är lätt att förstå. Importera till rätt ställe, fyll i vad som behövs och testa respektive dekonstruktion. (titta i labotation 1 om något är oklart)

5. Syntetisera designen och därefter kopplas yttre signaler in. Detta göres med hjälp av filen *Nexys4_Master.xdc* I katalogen `S:\Courses\eit\EITA15\Laborationer\Lab3` finns denna fil med signalerna clk, g1 och g2 deklarerade. Det som behöver läggas till är *cout[0]– cout[3]* samt *nrst*. Figur 3.2 ger en viss ledning hur IO-signalerna skall kopplas.
   Välj Nrst till lämplig tryckknapp på Nexys4 kortet..

Hemuppgift 3.2.4

Fyll i koden så att Nexys4_Master.xcd blir komplett.

6. Generera bitströmmen och överför den till FPGA'n på Nexys-4 kortet. Konsultera laboration 1 för att koppla upp mot kortet. Testa konstruktionen.
   Visa handledarna.

Slut på uppgift 3.2