

PROGRAMMERING AV MCU

Laborationens syfte

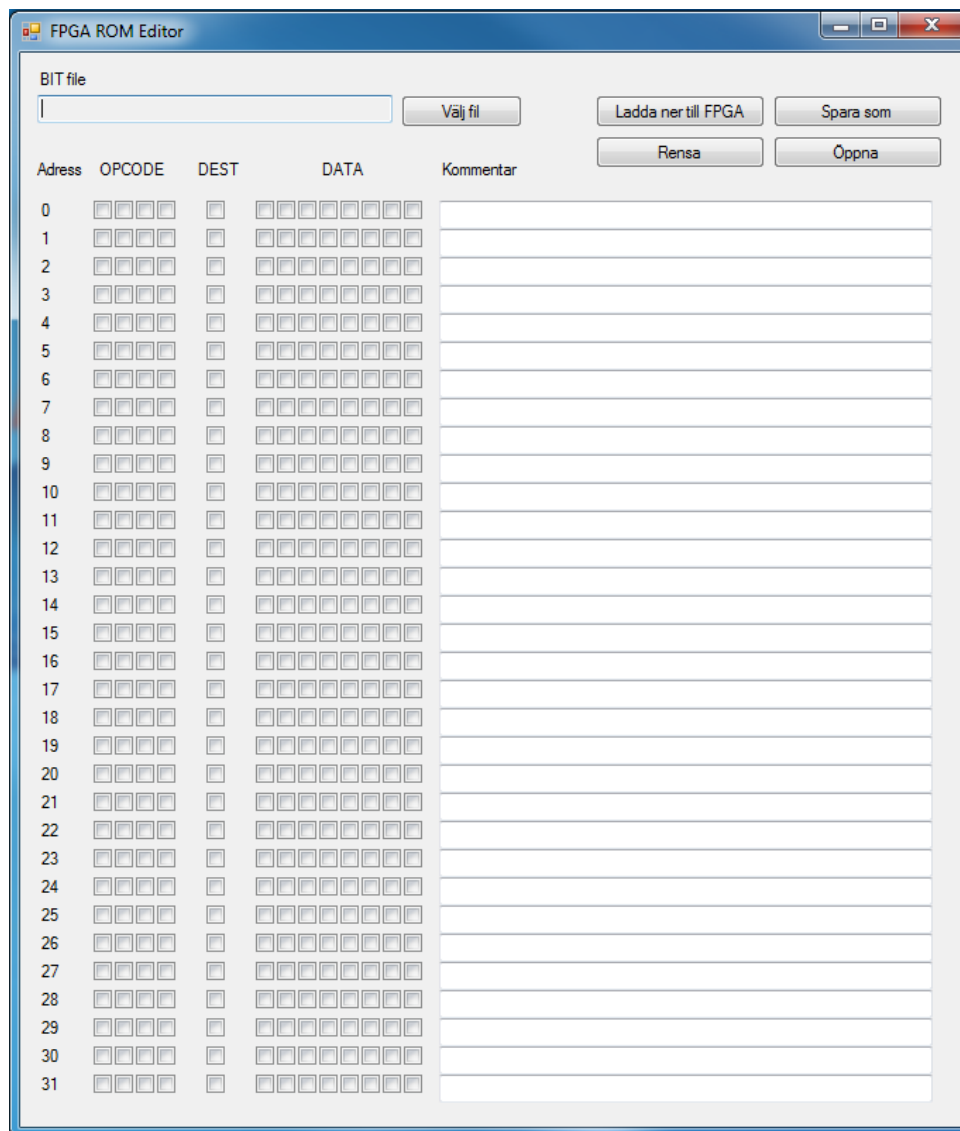
I denna laboration ska MCU:n som konstruerades i laboration 5 programmeras . Instruktionerna som vi har tillgång till är de som implementerades i instruktionsavkodaren i föregående labb.

Två uppgifter ska lösas, en trafikkorsning och en lejonbur med flera lejon (max 9) med hjälp av mikroprogrammering.

Mjukvara - Programmering av MCU

Med en väl fungerande MCU återstår nu bara att skriva programmen som löser de två problemen trafik Korsningen och lejonburen.

Minnesmodulen i MCU:n är avsedd att innehålla programmet som styr enhetens beteende. Efter att bitströmmen generats i Vivado fylls hela minnesarean med nollor. Bitströmmen måste därför modifieras så att ert program placeras i minnet. Modifieringen av bitströmmen görs med programmet **FPGA_ROM_Editor.exe** som finns i mappen `S:\Cources\eit\EITA15\laborationer\lab6` tillsammans med ett antal andra filer.



Figur 6.1: Skärmbild av FPGA_ROM_Editor's gränssnitt. Programmet används för att kopiera kod till MCU:s minne innan designen laddas ner till FPGA-kortet.

Figur 6.1 visar hur programmet ser ut då det startats. Antalet programrader (instruktioner) är 64 och varje instruktion består av en fyrabitars operationskod, en destinationsbit och åtta bitar data (MSB LSB). En bock i en ruta markerar att motsvarande bit är satt till en etta, annars noll. Efter varje instruktionsrad finns det en ruta där ni kan skriva kommentarer. De 5 knapparna i programmet har följande funktion

Välj fil	Välj bitström som ska modifieras.
Ladda ner till FPGA	Modifierar bitströmmen så att ert program läggs i minnet. Därefter laddas designen ner till FPGA-kortet via USB-kabeln.
Spara som	Sparar programet och kommentarer i en .hex fil
Öppna	Hämta program och kommentarer från .hex fil
Rensa	Nollställer alla instruktioner och kommentarer.

Då programmet laddas ner kommer även configurationen för hårdvaran att laddas över till FPGA-kortet.

Observera att första instruktionen kan inte börja med operanden IN!!

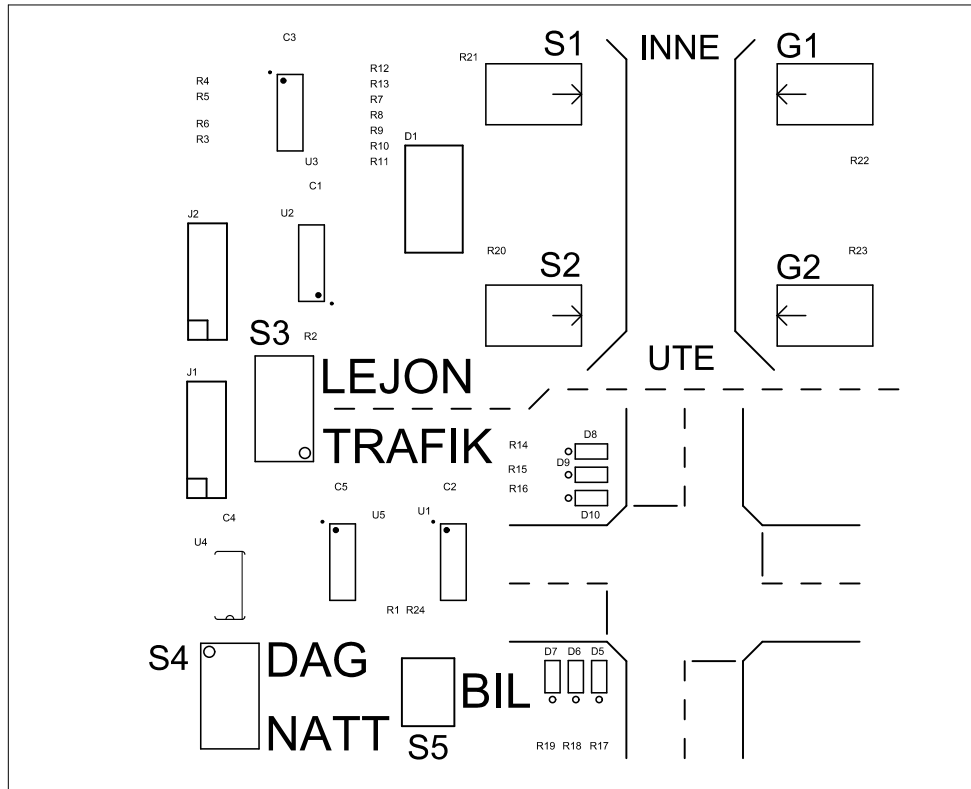
Labbkortet som är anslutet till Nexys4 består av en trafik Korsning och en lejonbur, se figur 6.2. Med strömställaren S3 väljer man om kortet skall fungera som en trafik Korsning (TRAFIK) eller lejonbur (LEJON). Det finns två kontakter på labbkortet märkta JA och JB. Dessa är anslutna till motsvarande kontakter på Nexys4.

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



Figur 6.2: Labbkort

Uppgift 6.1. Programmering av trafikorsning

I denna uppgift gäller det att programmera styrningen av trafikljus i en korsning, ett trafikljus på en huvudgata och ett på korsande sidogata. Strömställaren S4 fungerar som omkopplare mellan dag och natt. Knappen S5 används för att simulera att en bil kommer på sidogatan, se figur 6.2.

Under dagtid då $S4 = 0$ ska trafikljusena fungera enligt följande sekvens som upprepas

Huvudgata	Sidogata
grönt	rött
gult	rött
rött	rött
rött	rött + gult
rött	grönt
rött	gult
rött	rött
rött + gult	rött

I läget natt då $S4 = 1$ ska trafikljuset visa grönt på huvudgatan och rött på sidogatan. Då det kommer en bil på sidogatan ($S5 = 1$ då knappen Bil trycks ner) ska ljuset skall växla

en gång så att det blir grönt på sidogatan och bilen kan passera. Därefter ska det återgå till att vara grönt på huvudgatan.

Observera att det ska gå att se de olika ljussekvenserna då den automatiska klockan på 1 KHz används. Valet mellan 1 KHz och manuell klocka görs med strömställaren SW15. Om du valt manuell klocka tryck på knappen BTNC för att stega fram programmet.

Om programmet och kretsen ska nollställas och den manuella klockan är vald, måste man hålla knappen CPU RESET nere och klocka 2 gånger med BTNC för att nollställningen ska utföras.

Filerna ni behöver finns i katalogen S:\Cources\eit\EITA15\laborationer\lab6. Ladda ner dessa fyra filer till C:\users\StilID\program\lab6 (skapa lab6 om den saknas) starta programmet **FPGA_ROM_Editor.exe** och fyll i mikroprogrammet för trafik-korsningen (se fig 6.1)

BIT file är den från S: hämtade mcu.bit!

Se till att Lejon/Trafik är kopplad och att Nexys 4 kortet har spänning.

Testa!

Kalla på handledaren och förklara när ni är nöjda!

Slut på uppgift 6.1

Uppgift 6.2. Programmering av lejonbur

I denna uppgift skall en kontrollenhet programmeras som håller reda på hur många lejon som är ute. I princip är denna uppgift identisk med labbuppgift 4 men nu ska den mikroprogrammeras. Titta på föregående uppgift för att programmera.

Testa!

Kalla på handledaren och förklara när ni är nöjda!

Slut på uppgift 6.2