

Digital design

Labbrappo

Dana Ghafour
Högskoleingenjör i Data teknik
danagf@kth.se
19961024-3959

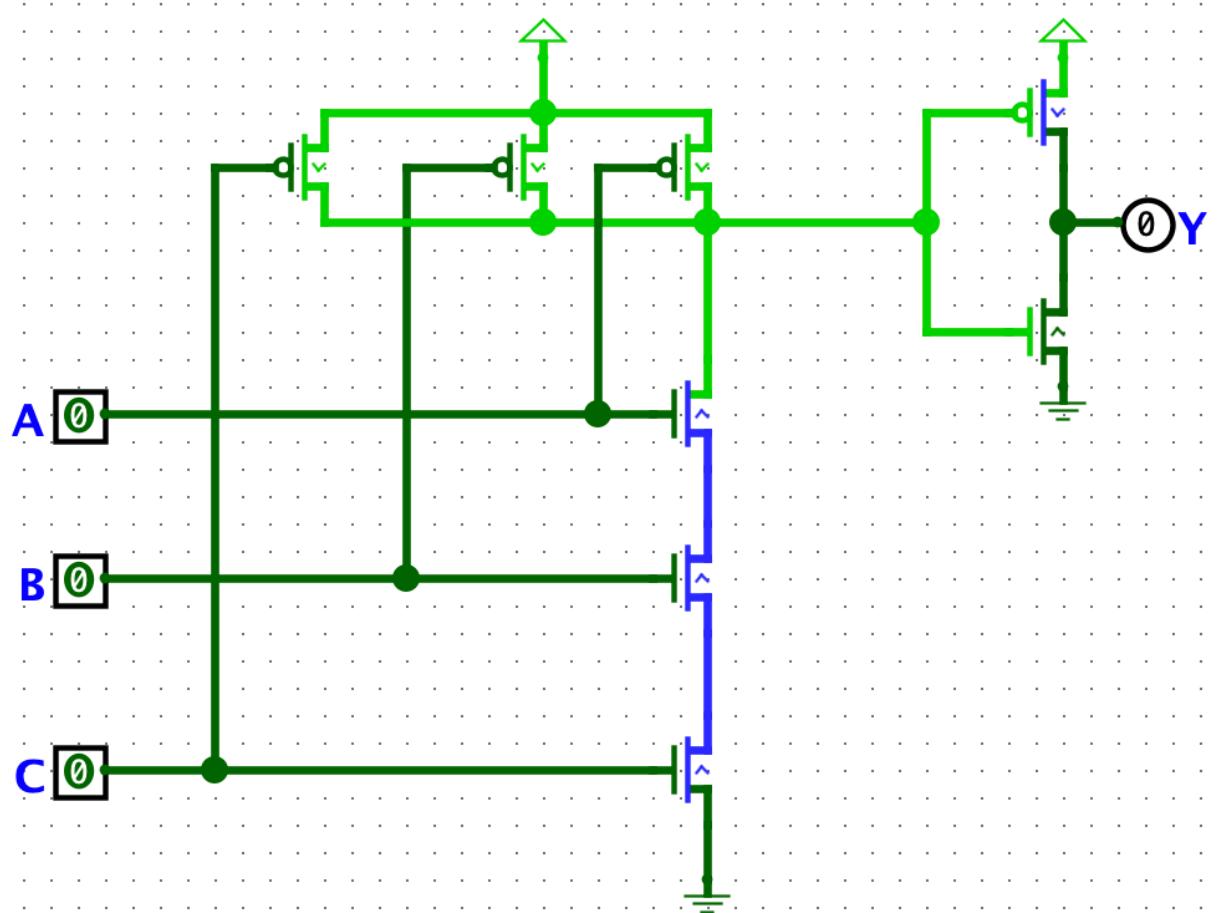
Module 1 task

Den här delen av labbrapporten kommer att beskriva de steg jag tog för att konstruera en CMOS OCH-grind med tre ingångar och en utgång. Kopplingsschemat för min CMOS-grind med tre ingångar byggdes med hjälp av MOSFET:er som är fälteffektransistorer, och kopplade in det på mitt kopplingsdäck. Sanningstabellen kunde jag därefter verifiera med hjälp av tre taktila omkopplare, med lysdioder kopplade till dessa, och en lysdiod på utgången.

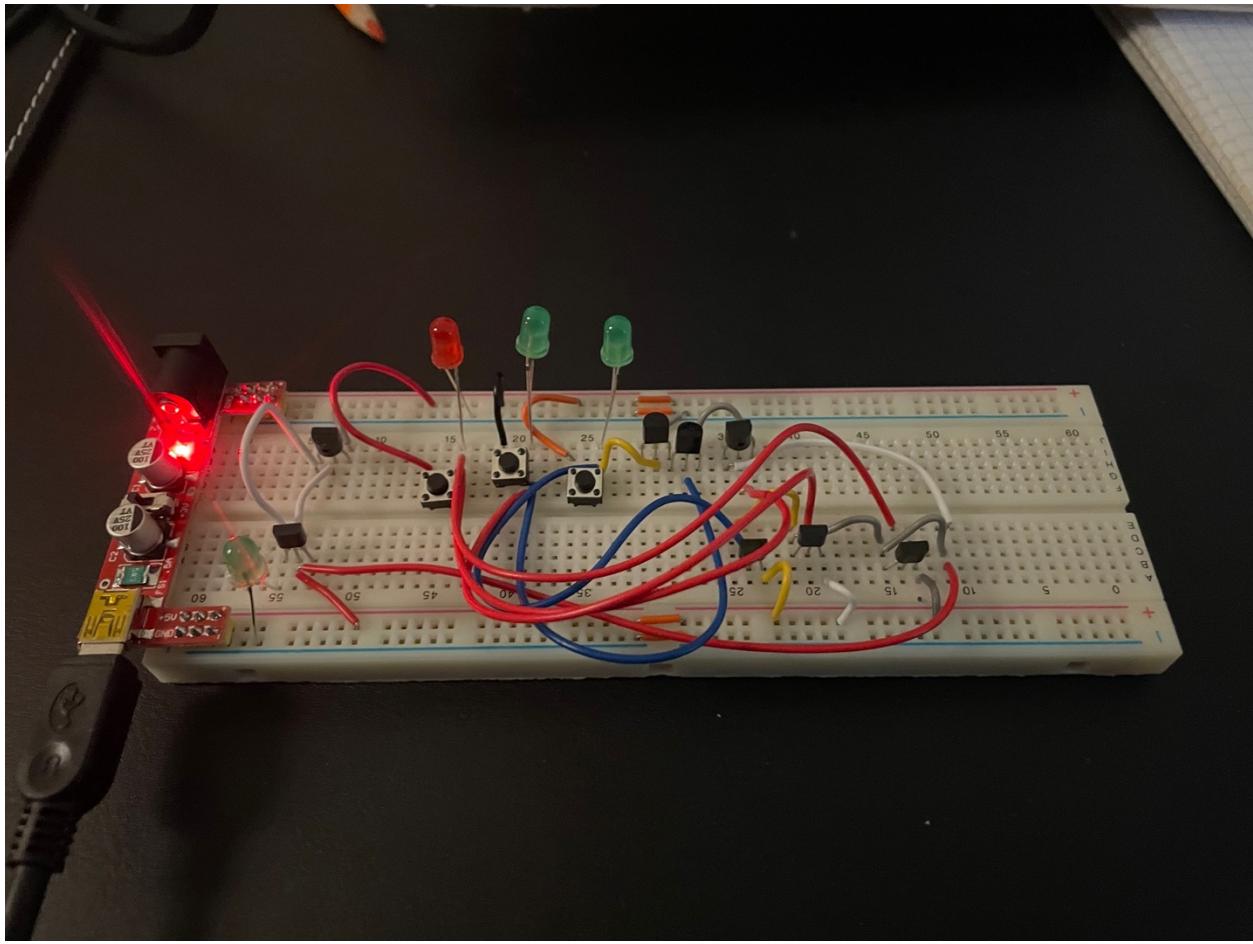
Kretsen simulerades med hjälp av Logisim, där en bild togs på kretsen efter att den fungerande kretsen demonstrerades för en Lab Assistant.

Allting gick inte så smidigt med labben som jag hade önskat mig. En av de största svårigheterna var att förstå hur man skulle ansluta alla komponenter på ett korrekt sätt för att få till en korrekt kretsschema. Inte nog med det så hade jag också svårt med användningen av sanningstabellen för att kunna verifiera att kretsschemat stämde, eftersom det var ganska nytt för mig. Men som tur är med användningen av Logisim gick det lite smidigare.

En annan problem var också att jag inte visste vilket ben på lysdioderna som var katoden, och vilken som var anoden. Men det var för det mesta mitt fel att jag inte var uppmärksam på det som stod i texten i canvas, även om det var mycket text och länkar som man behövde navigera sig igenom. Men som tur är, trots utmaningarna, lyckades jag äntligen få min kopplingsschema att fungera på kopplingsdäcket när den testades med en lampa.



A	B	C		Y
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		0
1	0	0		0
1	0	1		0
1	1	0		0
1	1	1		1



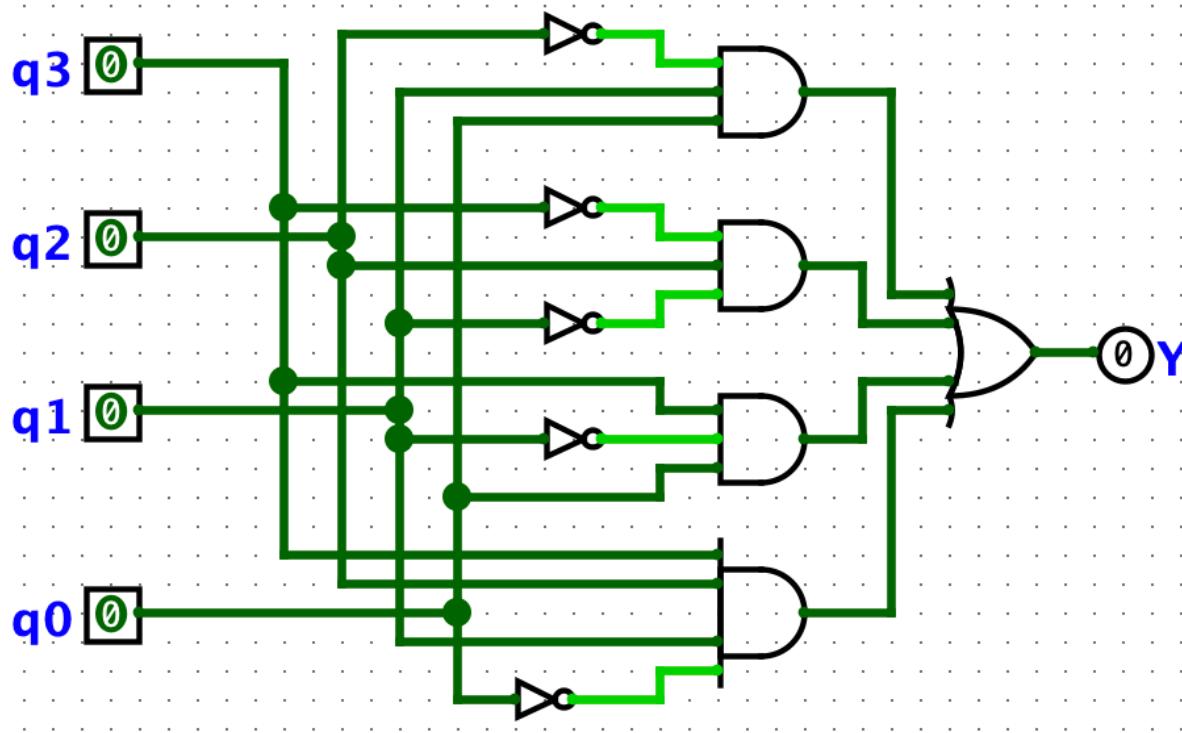
Utgången ligger i nedre-vänstra hörnet av kopplingsdäcket.

Module 2 task

Den här delen av labbrapporten kommer att beskriva de steg jag tog för att få utdatabitarna för en sanningstabell baserat på mitt födelsedatum. En K-diagram ritades sedan i syftet av att minimera det booleska uttrycket.

Jag använde Logisim för att verifiera designen senare, och kopplingsschemat byggdes på mitt kopplingsräck, och fyra taktala omkopplare samt en lysdiod användes för att kontrollera sanningstabellen. Sanningstabellen kontrollerades för att säkerställa att kretsen fungerade som jag hade i avsikt, och att jag fick de utgångsbitar som jag förväntade mig. När sanningstabellen hade kontrollerats på Logisim kunde jag också se om det förekom någon fel i logiken av mitt kopplingsschema. Till slut tog jag en bild på den fungerande kretsen och demonstrerade den för en Lab Assistant.

I denna labb hade jag svårighet med att veta var kablarna skulle gå på mitt kopplingsräck då det var första gången jag fick arbeta med integrerade kretsar. Till det var en annan svårighet att förstå diagrammen som fanns på instruktionerna i canvas, och om q0, q1, q2, och q3 representerade taktala omkopplare på Logisim. En annan svårighet var att rita kopplingsschemat på Logisim, även om det inte var lika svårt som i modul 1. Av labbassistenterna fick jag hjälp med att förstå att jag behövde en 4ingångars-OR grind, och att det fanns flera grindar inom en IC. Jag hade också problem med att förstå vilken sida på ICen som var GND eller VCC.



Output: Y ▾

Format: Sum of products ▾

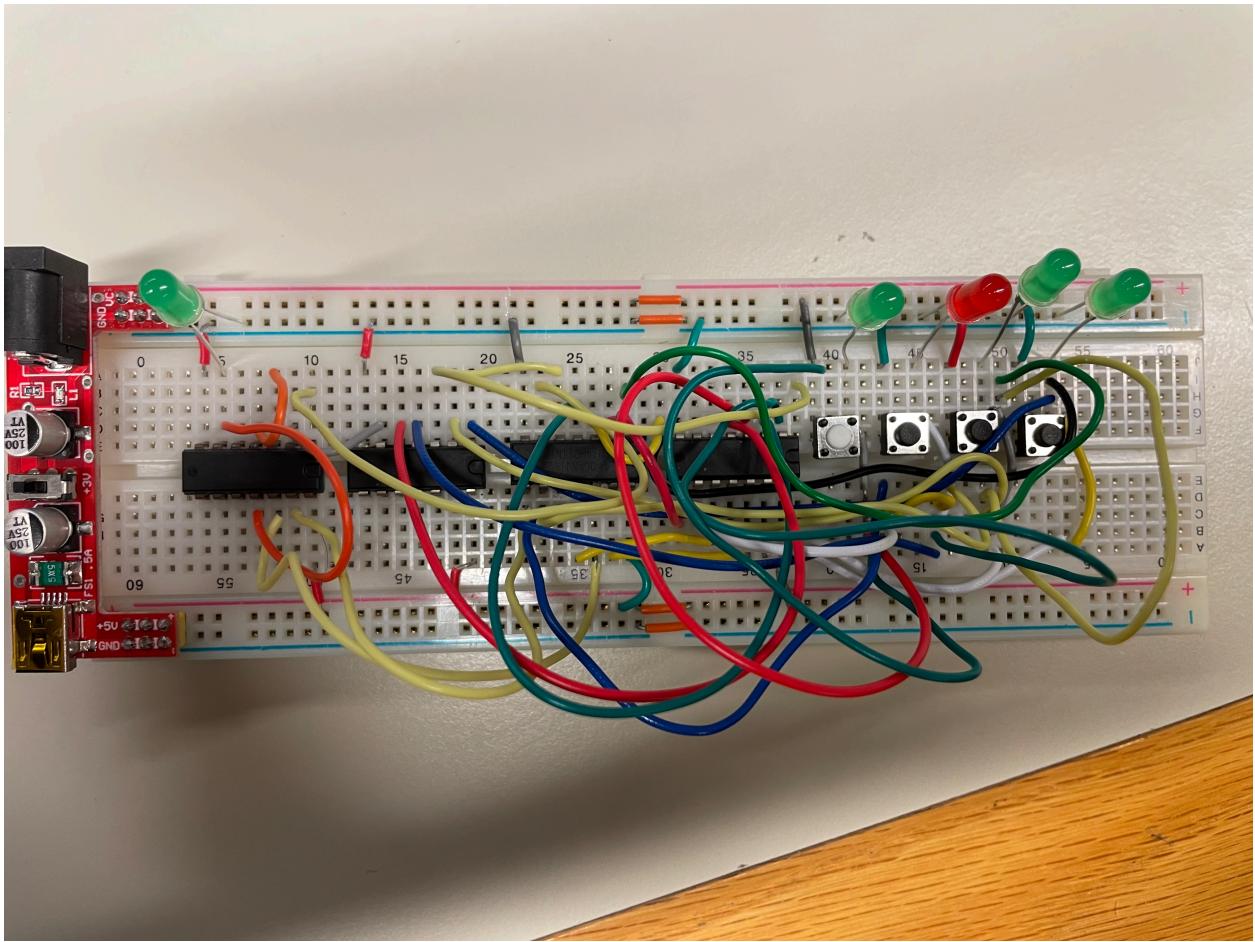
Style: Numbered ▾

Notation: Mathematical ▾

		q1, q0	00	01	11	10
		q3, q2	00	01	11	10
		00	0	0	1	0
		01	1	1	0	0
		11	0	1	0	1
		10	0	1	1	0

No group selected.

$$\overline{q_2} \cdot q_1 \cdot q_0 + \overline{q_3} \cdot q_2 \cdot \overline{q_1} + q_3 \cdot q_2 \cdot q_1 \cdot \overline{q_0} + q_3 \cdot \overline{q_1} \cdot q_0$$



Utgången ligger i utkant, övre-vänstra hörnet.

q3	q2	q1	q0	 	Y
0	0	0	0		0
0	0	0	1		0
0	0	1	0		0
0	0	1	1		1
0	1	0	0		1
0	1	0	1		1
0	1	1	0		0
0	1	1	1		0
1	0	0	0		0
1	0	0	1		1
1	0	1	0		0
1	0	1	1		1
1	1	0	0		0
1	1	0	1		1
1	1	1	0		1
1	1	1	1		0

Module 3 task

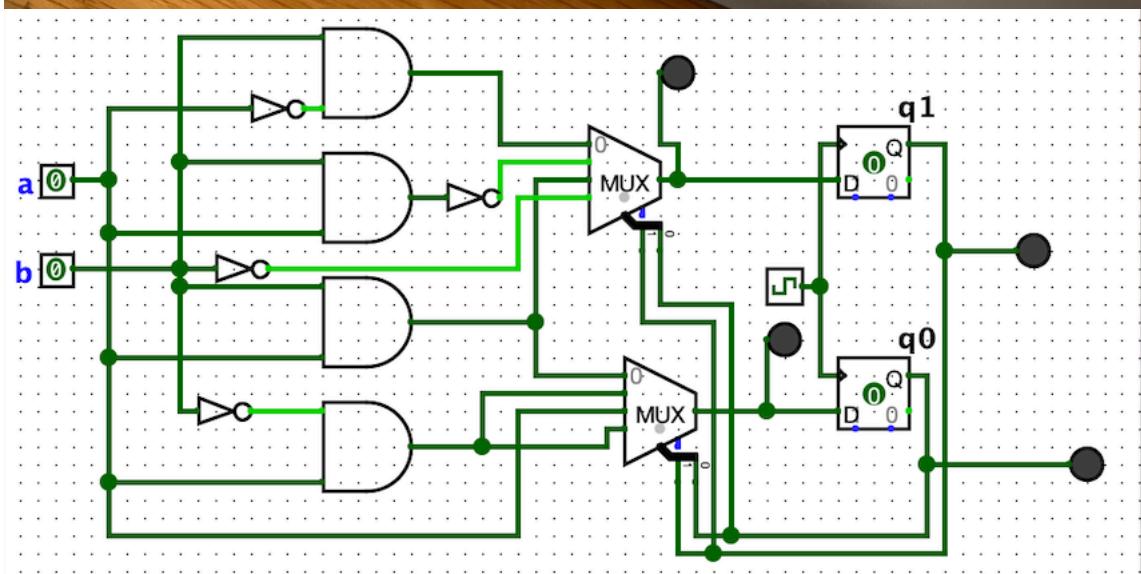
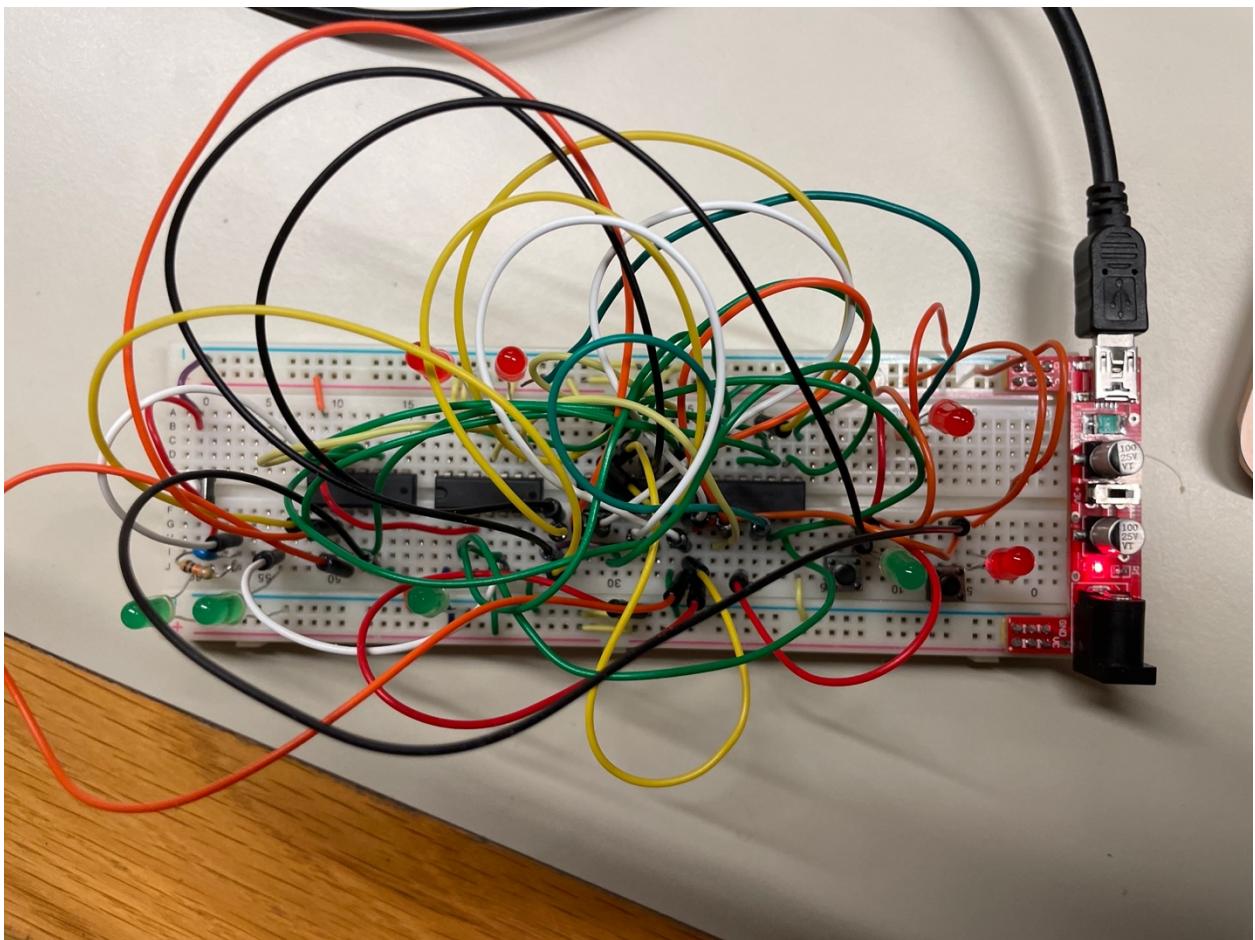
I den här delen av labrapporten kommer jag att redogöra för hur jag skapade en tillståndsmaskin baserad på min månads- och födelsedatum. Först hittade jag tillståndsdiagrammet som motsvarade min månads- och födelsedatum i FSM Calendar 2022, och det noterade numret för diagrammet var 22. Av denna anledning behövde jag inte lägga till någon ytterligare krets för att nå alla tillstånden. Ifrån min FSM förde jag över tillstånden till en K-diagram, för att därefter få ut en förenklad booleska uttryck för q1 och q0.

Jag kontrollerade sedan mina resultat från K-diagrammet med hjälp av Logisim. Sedan byggde jag tillståndsmaskinen med hjälp av två D-vippor (74HC74), MUX, och andra integrerade kretsar. Jag kopplade också lysdioder till q1 och q0, som motsvarade nuvarande tillstånd i tillståndsdiagrammet, och lysdioder till q1+ och q0+ för felsökning, som var nästa tillstånd.

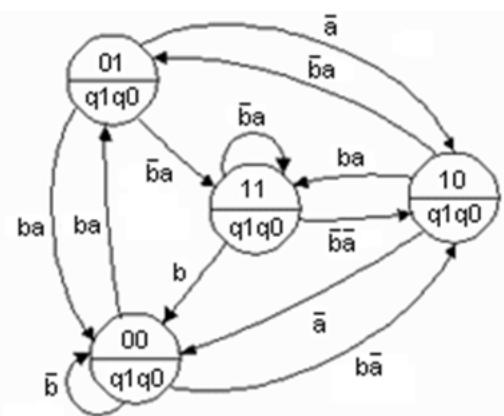
Sen användes en pulsgenerator-klocka för att testa tillståndsmaskinens funktion som skapades med hjälp av min Schmitt trigger.

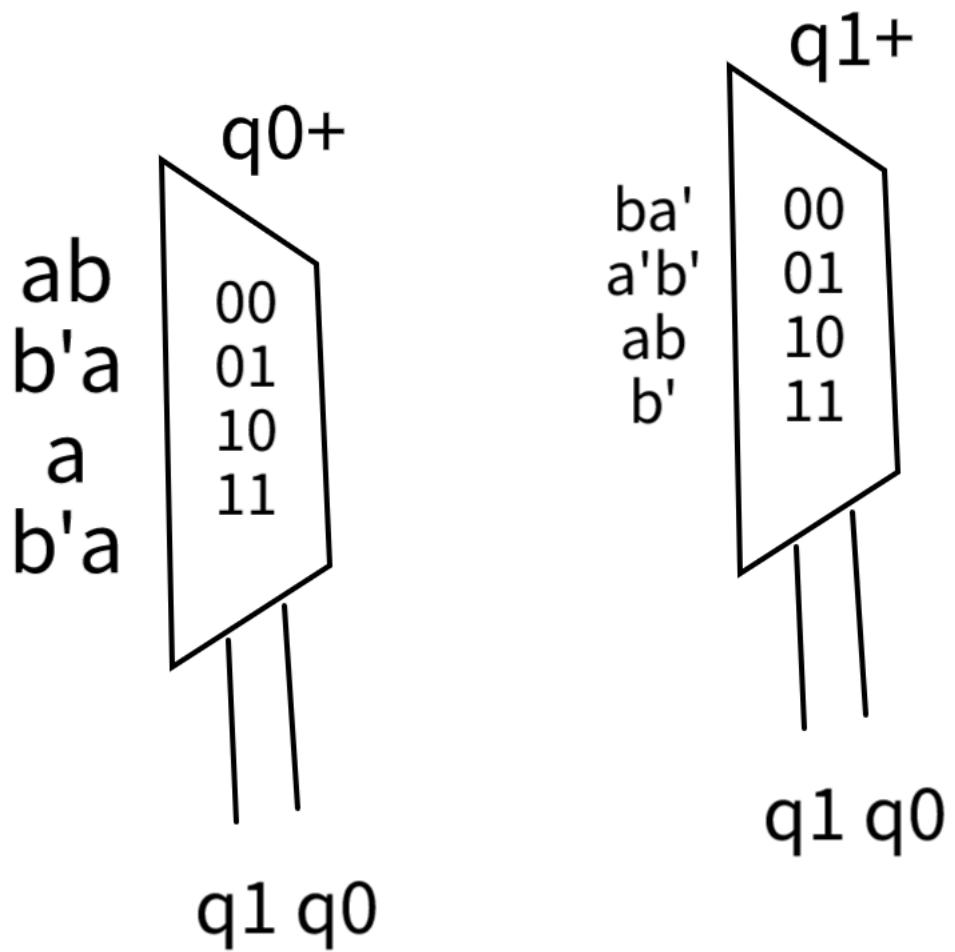
Till en början hade jag svårt att förstå hur man skulle bygga en Schmitt trigger klocka och vad som menas med att den skulle generera definierade positiva och negativa "edges". Något annat jag körde fast med var att veta hur jag skulle felsöka problemen med min krets, och anledningen till detta var att den inte fungerade enligt min FSM. Något annat som var svårt var att rita kopplingsschemat på Logisim med MUX, och vad som menades med att det skulle ersätta SOP och POS.

Efter att ha fått hjälp av min labbassistent så kom jag lite fram med mitt arbete. Jag försökte trycka på taktala omkopplaren som var kopplat till min d-vippa för att se vilka förändringar det skulle resultera i, och efter att ha experimenterat med att trycka på "b"- och "a"-knappen verkade det inte ge mig resultat jag ville. Men sen insåg jag att lysdioderna på q1+ och q0+ var endast i felsökningssyfte, och att de inte är relevanta för min FSM.



October 16-31
Version 22 ↓





Truth table for state $q0+$:

		00	01	11	10
		0	0	1	0
		0	1	0	0
		0	1	0	0
		0	1	1	0

Inputs: b, a

Outputs: $q1, q0$

Truth table for state $q1+$:

		00	01	11	10
		0	0	0	1
		1	1	0	1
		1	1	0	0
		0	0	1	0

Inputs: b, a

Outputs: $q1+ q0$

Module 4 task

Innan labbet kopplade jag en 7-bitars display till en CD4543B 7-segmentsavkodare. Segmenten "a" kopplades till "a", "b" till "b" osv. Sedan anslutade jag pin tre och pin åtta på displayen till jord. Pin 1 "Latch" behövde vara kopplad högt till VCC.

Jag anslöt sedan diod-ROM minnet med min födelsedatum som dess innehåll, 19961024.

Y0 = 0001

Y1 = 1001

Y2 = 1001

Y3 = 0110

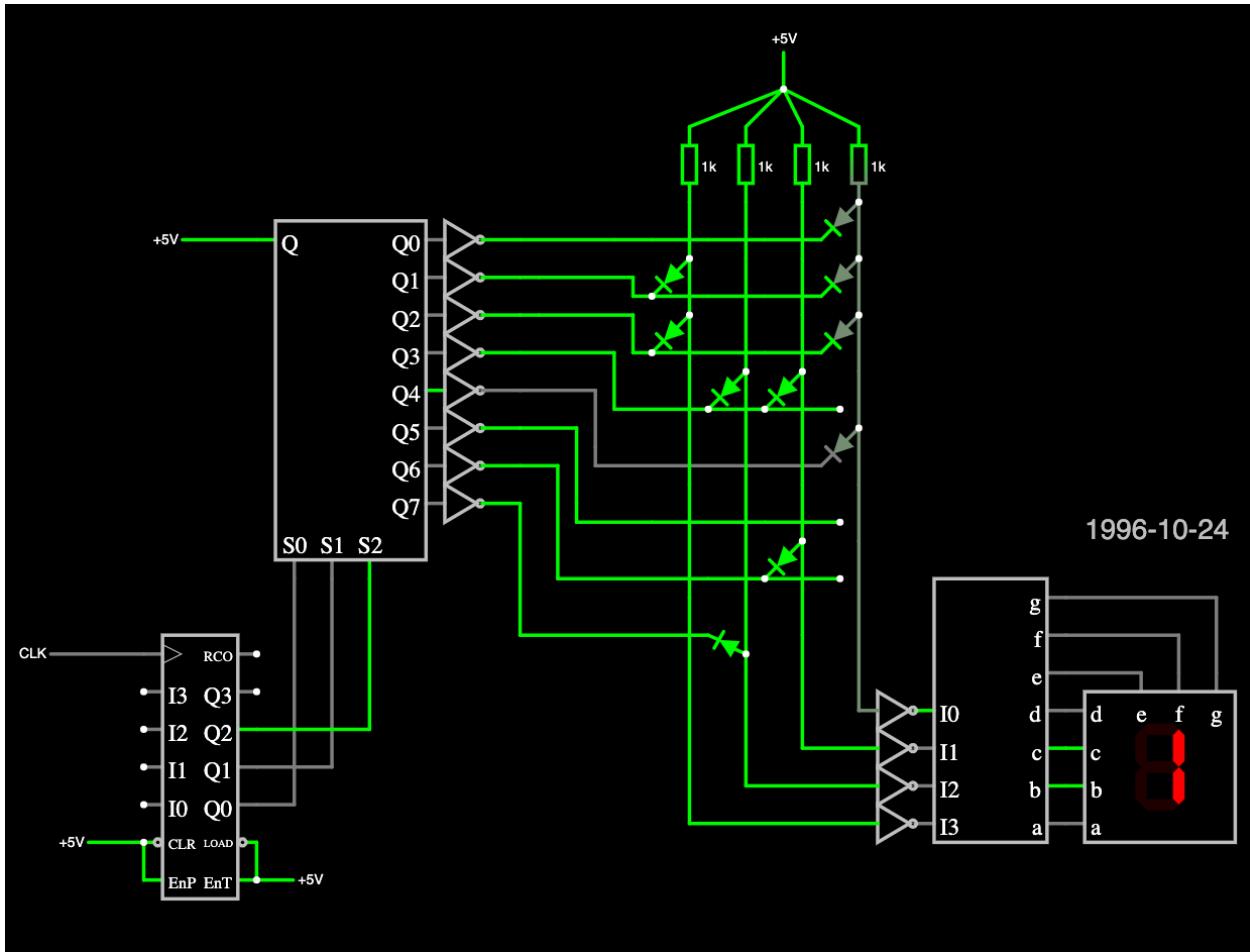
Y4 = 0001

Y5 = 0000

Y6 = 0010

Y7 = 0100

Utgång	Bits
Y0	0001
Y1	1001
Y2	1001
Y3	0110
Y4	0001
Y5	0000
Y6	0010
Y7	0100



I början undrade jag om jag skulle designa kopplingsschemat på Logisim, eftersom det underlättade om jag kunde simulera för att se om allting såg korrekt ut.

Ett annat problem var också att få alla dioder att få plats på kopplingsdäcket inom en liten yta. Lösningen var då att skapa lite mellanrum mellan resistorerna på däcket, och det underlättade kopplingen för mig och minimerade fel. I början hade jag svårigheter att få alla dioder att fungera som de skulle, därav löste jag det genom att sätta resistorerna med lite avstånd. Dessutom hade jag problem med att få counter att fungera som det skulle, och försökte lösa det genom att koppla om kablarna.

Att simulera på Falstad underlättade arbetet för mig för att jag skulle kunna få en överblick över arbetet. Logisim hade tyvärr inte de nödvändiga komponenter som behövdes för simuleringen.

