

Linköpings tekniska högskola, ISY, Datorteknik

Laborationskompendium

TSEA22 Digitalteknik



Linköping 2022

Introduktion

Detta häfte innehåller laborationsuppgifter i digitalteknik och används i kurserna TSEA22, TSEA51, TSEA52 och TDDC75. Läs igenom dokumentet “Allmänna anvisningar för laborationer i digitalteknik” som finns länkat från kurshemsidan innan ni börjar att lösa uppgifterna. “Allmänna anvisningar för laborationer i digitalteknik” ger råd och stöd för laborationernas genomförande. Vid konstruktion, följ det tillvägagångssätt som beskrivs i avsnitten 3.1 (Konstruktion) och 3.2 (Uppkoppling). Läs och begrunda även avsnitt 3.3 (Felsökning) **inför varje laboration**. (Bäst förståelse erhålls sedan man bekantat sig med utrustningen).

De fel som är svårast att hitta vid laborationerna är de rent mekaniska. **Dessa uppstår p.g.a. att materielen inte behandlas tillräckligt varsamt**. Sträcks sladdarna för hårt finns risk att det blir glapp i kontaktstiften och i sladdkontakten. När konstruktionerna kopplas ner är det av största vikt att varje sladd lossas genom att **dra i kontakten** (inte i sladden) och **rakt upp**. Lösa stift, avbrott och dåliga kontakter ger upphov till s.k. intermittenta fel. Dessa är mycket svårifunna, eftersom de har ett slumpmässigt beteende, och drabbar efterföljande laboranter.

Till laborationerna ska du inte bara medföra ett klart och tydligt kopplingschema för varje uppgift utan också de fullständiga lösningarna. Har du inte förberett uppgifterna till aktuell laboration så går det inte att klara uppgifterna på utsatt tid. Får du problem med förberedelsen av någon uppgift kan du få tips av lektionsassistenten.

Laborationerna ger studiepoäng och utgör därför ett examinationsmoment på samma sätt som en tentamen. Uppgifterna ska redovisas för och godkännas av laborationsassistenten. Godkända hela laborationer, men inte ströuppgifter, tillgodoräknas från ett läsår till nästa. Vid det första labtillfället kommer ni få ut en lapp där labassistenterna signerar avklarade uppgifter. Det är ert ansvar att ta med lappen till laborationerna. När en uppgift är godkänd ska labassistenten markera detta genom att signera er lapp.

Laboration 1

Kombinationskretsar

I denna laboration skall du undersöka, förenkla och konstruera olika kombinatoriska nät. Laborationen ger dig också praktisk erfarenhet om komplexiteten vid uppkoppling av näten (= realisering av konstruktionen). Du kommer också märka att ett gediget och noggrant förarbete innan labben är av godo. Laborationen är uppdelad i två 2-timmarspass. Till det första 2-timmarspasset bör uppgift 1.1–1.4 förberedas.

Efter genomförd laborations ska ni:

- Känna till, förstå funktionen av samt använda kombinatoriska byggblock för design av digitala system.
- Kunna konstruera kombinationskretsar med hjälp av TTL-kretsar.
- Systematiskt kunna felsöka kretsar.

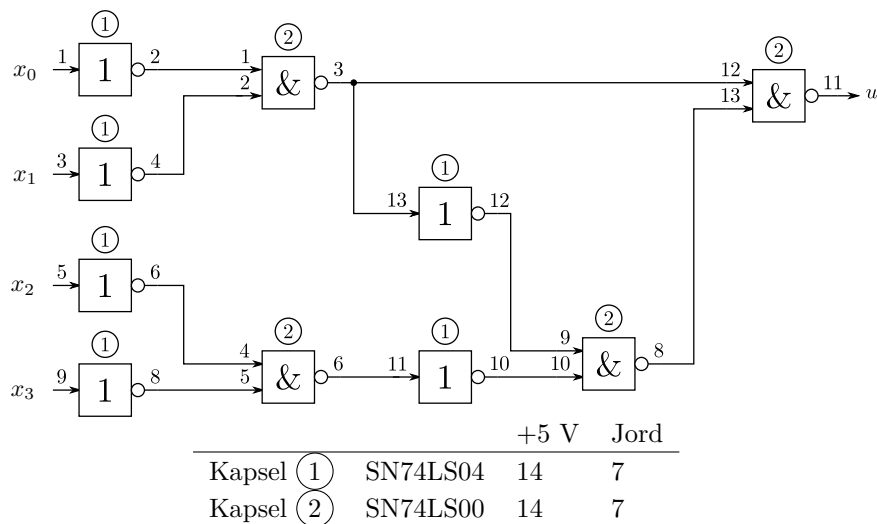
1.1 Förberedelser

Varje uppgift är indelad i inledande förberedelseuppgifter och labuppgifter. Förberedelseuppgifterna ska vara klara innan laborationen och laborationsuppgifterna ska genomföras på laborationstid. Både förberedelseuppgifter och laborationsuppgifter examineras under laborationerna. Kretsscheman skall också inkludera **numrerade kapslar och ben** för att underlätta uppkoppling.

1.2 Uppgifter

Uppgift 1.1. Undersökning och förenkling av en given kombinationskrets. Vi vill förenkla kretsen i figuren. Följ instruktionerna steg för steg.

Figuren visar ett kopplingschema där siffror i cirklar anger grindar som får plats i samma kapsel, kapsel ① är av typ 74LS04 och kapsel ② av typ 74LS00. Siffror vid ingångar och utgångar anger vilken pinne på respektive modul som ska användas.



Förberedelser:

(a) Fyll i kolumnen "u, beräknad" i funktionstabellen nedan.

x_3	x_2	x_1	x_0	u, beräknad	u, uppmätt (original)	u, uppmätt (förenklad)
0	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	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

- (b) Fyll i Karnaughdiagrammet med hjälp av funktionstabellen

	x_1x_0			
	00	01	11	10
x_3x_2	00			
	01			
	11			
	10			

- (c) Skriv ett minimalt uttryck baserat på NAND-grindar och inverterare.

- (d) Rita ett logiskt kopplingsschema med kapsel- och bennumrering på samma sätt som originalkretsen är numrerad. Använd endast NAND-grindar och inverterare.

Laborationsuppgifter:

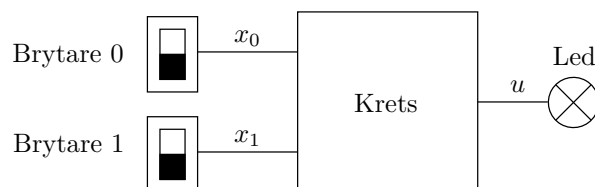
- (e) Koppla upp originalkretsen enligt figuren. Ingångsvariablerna x_3 , x_2 , x_1 , och x_0 hämtas från skjutomkopplare. Utsignalen registreras med hjälp av en lysdiod. Variera insignalerna enligt funktionstabellen och komplettera “ u , uppmätt (original)” med det erhållna resultatet. Tänd lysdiod motsvarar $u = 1$.

Gå inte vidare förrän de beräknade och uppmätta värdena stämmer överens.

- (f) Koppla upp den förenklade kretsen enligt förberedelserna i deluppgift (d). Verifiera kretsens funktion och fyll i “ u , uppmätt (förenklad)” i funktionstabellen. Ett vanligt mått att uppskatta komplexiteten hos en digital krets är att summera ingångarna till kretsens alla grindar och inverterare. Hur mycket förenklades kretsen genom minimeringen?

Svar:

Uppgift 1.2. I ett trapphus ska en LED-lampa tändas och släckas med två lysknappar här implementerade som skjutomkopplare. Oberoende av vilken skjutomkopplare som ändrar läge så ska lampan tändas om den var släckt eller släckas om den var tänd. Konstruera kretsen med valfria grindar och inverterare i labsatsen. I databladet står vilka grindar som finns att tillgå i labsatsen.



Förberedelse: Rita funktionstabell, Booleskt uttryck och kopplingsschema:

Laborationsuppgift: Koppla upp och verifiera kretsens funktion. Glöm inte att även koppla in matningsspänning (+5 V och GND) till logikkretsarna.

Uppgift 1.3. Uppgiften syftar till att förstå hur en Boolesk funktion kan realiseras med hjälp av olika typer av grindar, dels NAND-grindar och dels NOR-grindar. Kretsens funktion ska vara

$$u(x_2, x_1, x_0) = x_2(x_1 + x_0).$$

I båda deluppgifterna är det tillåtet att använda inverterare för att skapa inverterade insignaler som t ex x'_0 .

Förberedelser:

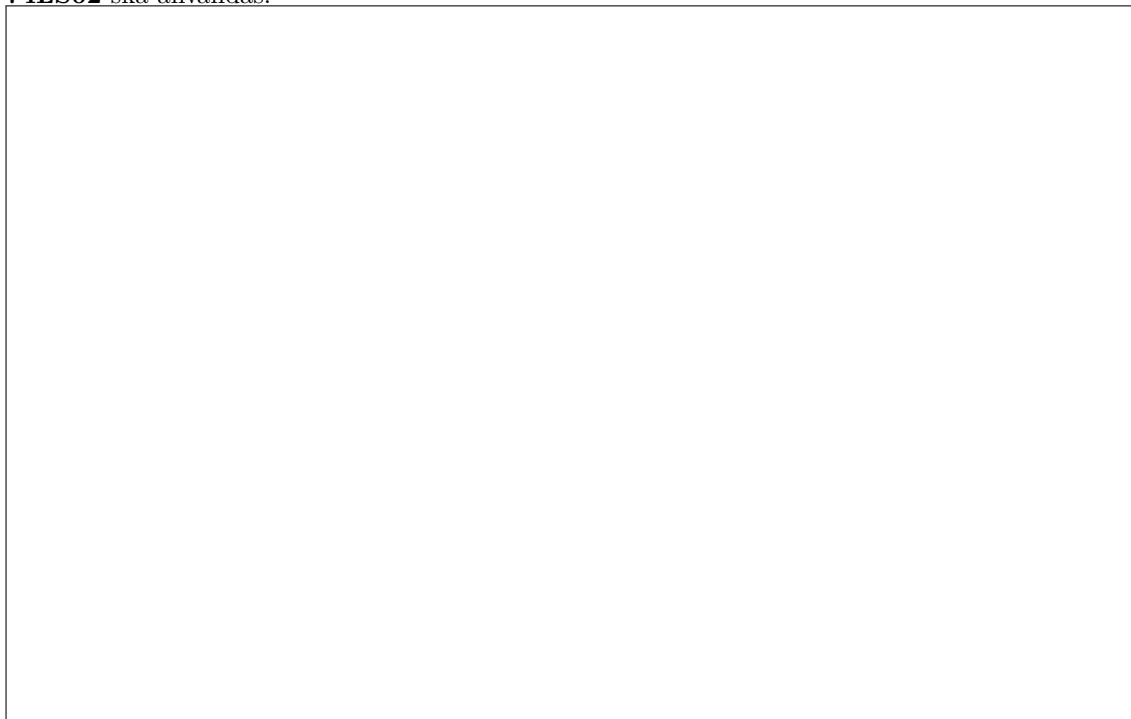
(a) Fyll i kolumnen “ u , beräknad” i funktionstabellen.

x_2	x_1	x_0	u , beräknad	u , NAND-kretsen uppmätt	u , NOR-kretsen uppmätt
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

- (b) Skriv upp det minimala uttrycket och kopplingsschema för u då NAND-grindar av typen **74LS00** ska användas. Utnyttja gärna Karnaughdiagram för att visa att uttrycket är minimalt.



- (c) Skriv upp det minimala uttrycket och kopplingsschema för u då NOR-grindar av typen **74LS02** ska användas.



Laborationsuppgift:

- (d) Koppla upp kretsarna förberedda i deluppgift (b) och (c) och verifiera kretsarnas funktion genom att fylla i funktionstabellen i (a)-uppgiften.

Uppgift 1.4. En kombinationskrets som realiserar funktionen

$$u(x_3, x_2, x_1, x_0) = \sum (0, 1, 2, 3, 6, 7, 9, 13, 14, 15),$$

där x_3 är mest signifikant bit ska konstrueras med minimalt antal NAND-grindar och inverterare.

Förberedelser:

(a) Karnaughdiagram och Booleskt uttryck:

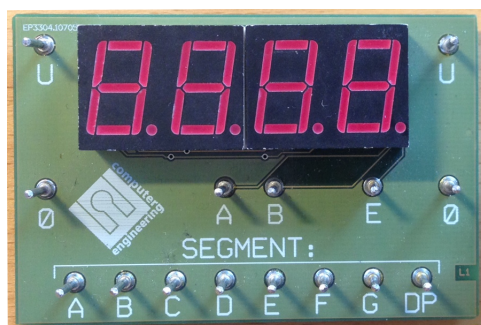
(b) Kopplingsschema:

Laborationsuppgift:

(c) Koppla upp kretsen och verifiera funktionen genom att fylla i funktionstabellen nedan.

x_3	x_2	x_1	x_0	u , uppmätt	x_3	x_2	x_1	x_0	u , uppmätt
0	0	0	0		1	0	0	0	
0	0	0	1		1	0	0	1	
0	0	1	0		1	0	1	0	
0	0	1	1		1	0	1	1	
0	1	0	0		1	1	0	0	
0	1	0	1		1	1	0	1	
0	1	1	0		1	1	1	0	
0	1	1	1		1	1	1	1	

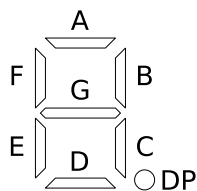
Uppgift 1.5. Konstruera en krets som visar ett binärt tal $x = (x_3, x_2, x_1, x_0)$ på en sju-segmentsdisplay hexadecimalt. Insignalerna hämtas från skjutomkopplare och kretsen konstrueras med PROM, ett eller flera, samt modulen med sju-segmentsdisplayer som visas på bilden.



Följande tal ska visas på en av de fyra displayerna för motsvarande binärtal:



Segmentnamnen är enligt:



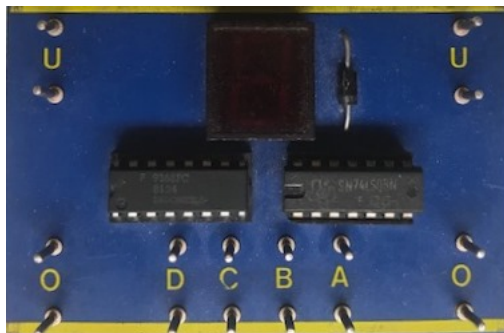
Utöver segmenten så finns det även tre andra ingångar. A och B väljer vilken sju-segmentsdisplay av de fyra som siffran visas på. Om $A = B = 0$ används displayen längst till höger, om $A = 0, B = 1$ nästa och så vidare. E är en enable-signal och måste sättas till 1 för att något skall visas.

Förberedelse: Rita logiskt kopplingschema och minnesinnehåll i PROM:en:

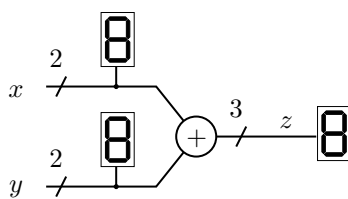
Laborationsuppgift: Koppla upp kretsen, programmera PROM:en och kontrollera funktionen genom att se vad displayen visar för samtliga insignalkombinationer. När uppgiften är redovisad och godkänd ska PROM:en nollställas genom att i PROG-mode samtidigt trycka på de tre röda knapparna.

Uppgift 1.6. Denna uppgift finns i två versioner den vanliga som beskrivs först eller en svårare variant som beskrivs sist. Välj en variant och implementera kretsen.

Konstruera en 2-bitsadderare enligt figuren nedan där $x = (x_1, x_0)$, $y = (y_1, y_0)$ och $z = (z_2, z_1, z_0)$. Använd skjutomkopplare för att välja insignaler och de avkodade sju-segmentsdisplayerna



för att visa talen x och y som ska adderas samt resultatet z . Bygg adderaren genom att implementera en heladderare i vardera PROM och koppla ihop de två PROM:en till en tvåbitsadderare. Observera att det går att bygga tvåbitsadderaren med 1 PROM men eftersom den iterativa strukturen går förlorad är denna lösning ej tillåten.



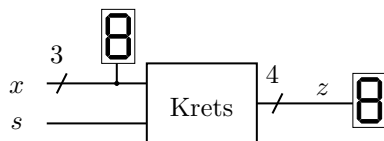
Tips: Glöm inte att koppla in 0:or på de mest signifikanta bitarna till displayerna.

Alternativ svårare uppgift: Bygg en adderare där x har 4 bitar och y 3 bitar och z 5 bitar i figuren ovan. Två PROM får användas och programmeras fritt. Använd alla 6 skjutomkopplare och en tryckomkopplare för att generera de 7 insignalerna samt led-lampor för att visa utsignalerna z .

Förberedelse: Rita kopplingsschema och minnesinnehållet i PROM:en:

Laborationsuppgift: Koppla upp kretsen och verifiera kretsens funktion för samtliga insignalvärden.

Uppgift 1.7. Konstruera en krets som kan multiplicera ett tal med 2.



Insignaler till kretsen är ett trebitstal $x = (x_2, x_1, x_0)$ och en styrsignal s . Utsignal är ett hexadecimalt fyrbitstal $z = (z_3, z_2, z_1, z_0)$. Kretsens funktion ska vara

$$z = \begin{cases} x & \text{om } s = 0, \\ 2x & \text{om } s = 1. \end{cases}$$

Visa talen x och z på avkodade sju-segmentsdisplayer. Använd skjutomkopplare för att välja värde på x och s . Till er konstruktion får **endast** multiplexrar användas, dvs 74LS153 och 74LS157 i labsatsen.

Förberedelse: Rita kopplingsschema:

Laborationsuppgift: Koppla upp kretsen och verifiera kretsens funktion för samtliga insignalvärden.

Tips: Alla insignaler till displayer och multiplexrar måste kopplas in. Se speciellt insignalen STROBE på multiplexrarna.

Laboration 2

Sekvenskretsar

Syftet med laborationen är att få förståelse för och färdighet att kunna konstruera synkrona sekvenskretsar på ett systematiskt vis. Metoderna omfattar förutom de som övades i laboration 1 av tidsdiagram, tillståndsdigram och tillståndstabeller.

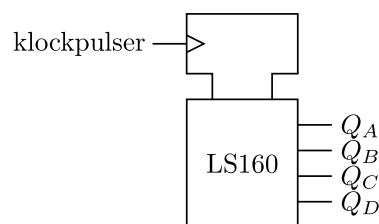
Efter genomförd laboration ska ni:

- Med systematiska metoder kunna konstruera synkrona sekvenskretsar.
- Känna till, förstå funktionen av samt använda D-vippor och räknare.
- Förstå skillnaden mellan asynkrona och synkrona insignaler och kunna synkronisera asynkrona insignaler.
- Kunna felsöka sekvenskretsar genom att med mätning och manuell klockning stega igenom sekvenser av tillståndsövergångar.

Först ges en introduktion av räknare i avsnitt 2.1 därefter följer uppgifterna i avsnitt 2.2.

2.1 Introduktion av räknare

TTL-familjen innehåller ett flertal olika räknare. En räknare är ett sekvensnät som i takt med en klockpulssignal växlar från ett räkneläge till ett annat.



Du ska undersöka den synkrona dekadräknaren 74LS160. Som framgår av namnet har denna räknare 10 räknelägen.

Antal inräknade klockpulser	Räkneläge			
	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0
11	0	0	0	1

Räkneläget ändras vid positiv flank på klockpulsen.

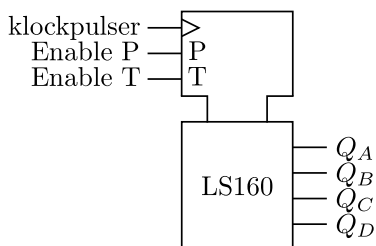
Klockpulser:



Förutom klockingången och utgångarna $Q_D Q_C Q_B Q_A$ med räknetillståndet har räknaren ett flertal andra in- och utgångar. Jämför nedanstående beskrivning med databladen i "Allmänna anvisningar ...".

Enable-ingångar

För att räknaren överhuvud taget ska kunna räkna krävs att Enable-ingångarna P och T aktiveras.

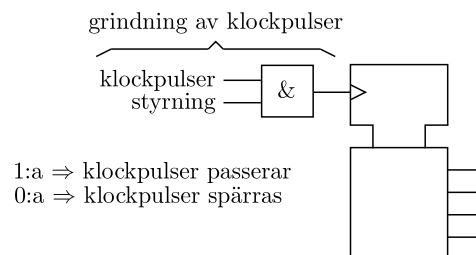


Enable P och T aktiveras med en 1:a.

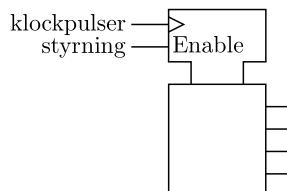
Enable		Räknarens funktion
P	T	
0	0	Ingen ändring av räknarläget
0	1	Ingen ändring av räknarläget
1	0	Ingen ändring av räknarläget
1	1	Räknaren ändrar läge vid positiv flank

Enable-ingångarna används för att starta och stoppa räknaren. **Start och stopp av räknaren får aldrig ske genom grindning av klockpulser.**

FEL:

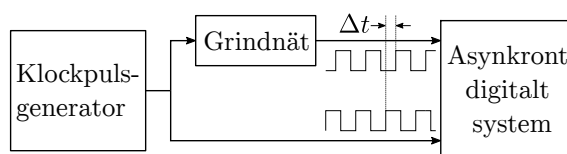


RÄTT:



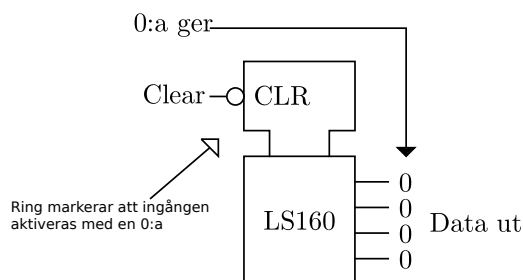
Grindning av klockpulser leder i bästa fall till att ditt digitala system blir asynkront. Systemet kan dock p.g.a. styrningen ta emot ofullständiga klockpulser och kan då hamna i vilket läge som helst eller, i värsta fall, t.o.m. erhålla utgångsvärden i det förbjudna området mellan logiskt noll och logiskt ett.

Även om systemet inte spårar ur blir det ändå asynkront, eftersom klockpulserna är tidsförskjutna. Alla dina teoretiska kunskaper i digitalteknik gäller synkrona nät.



Clear-ingången

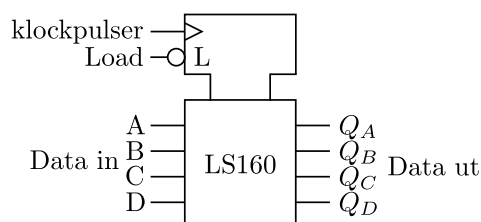
När Clear-ingången aktiveras nollställs räknaren oberoende av klockpulser och Enable-ingångar. Clear-ingången aktiveras med en nolla illustrerat genom att det finns en ring (motsvarande invertering) på ingången. Eftersom Clear-ingången verkar utan klockpulser sägs Clear-funktionen vara asynkron.



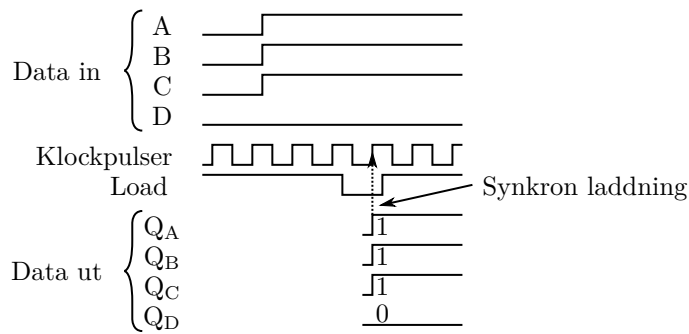
Clear-ingången används t.ex. för manuell nollställning av räknaren.

Load-ingången

Med hjälp av Load-ingången kan räknaren laddas med data som anges av de fyra dataingångarna. Load-ingången aktiveras liksom Clear-ingången med en nolla och laddning sker vid klockpulsens positiva flank.

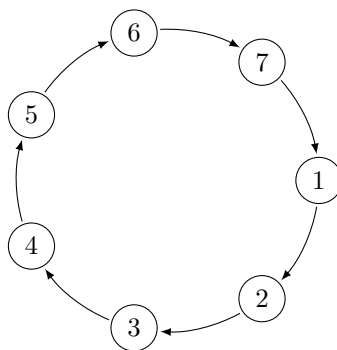


Exempel:

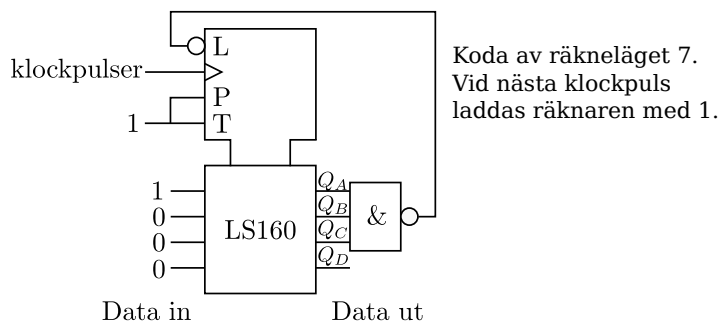


Laddningen sker synkront och **oberoende av Enable-ingångarna.**

Exempel 2.1 Räkaren 74LS160 ska räkna runt i sekvensen

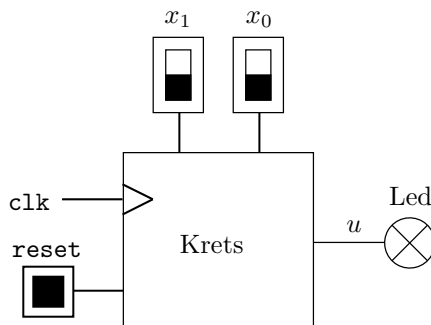


Detta kan åstadkommas genom att ladda räkaren med 1 vid räkneläge 7 enligt:



2.2 Uppgifter

Uppgift 2.1. Ett kombinationslås ska konstrueras som en synkron sekvenskrets enligt Moore illustrerad i följande figur:



Låset har två osynkroniserade insignaler x_1 och x_0 samt en utsignal u . Insignalerna hämtas från de två studs fria skjutomkopplarna (det övre läget ger logiskt ett och det nedre logiskt noll) och utsignalen avläses på en lysdiod.

Låset öppnas om skjutomkopplarna manövreras i sekvens enligt 1–3

1. Båda i nedre läget
2. Vänster i nedre läget, höger i övre läget
3. Vänster i övre läget, höger i övre läget

vilket markeras av att lysdioden tänds. Låset ska förbli öppet ända tills båda omkopplarna förs till nedre läget varefter nya öppningsförsök ska kunna göras. Det ska även finnas en asynkron resethknapp som återställer systemet till startläge, dvs låset är stängt och hela sekvensen 1–3 måste genomlöpas för att öppna låset igen. Detta ska ske oberoende av hur skjutomkopplarna står när resethknappen trycks ned.

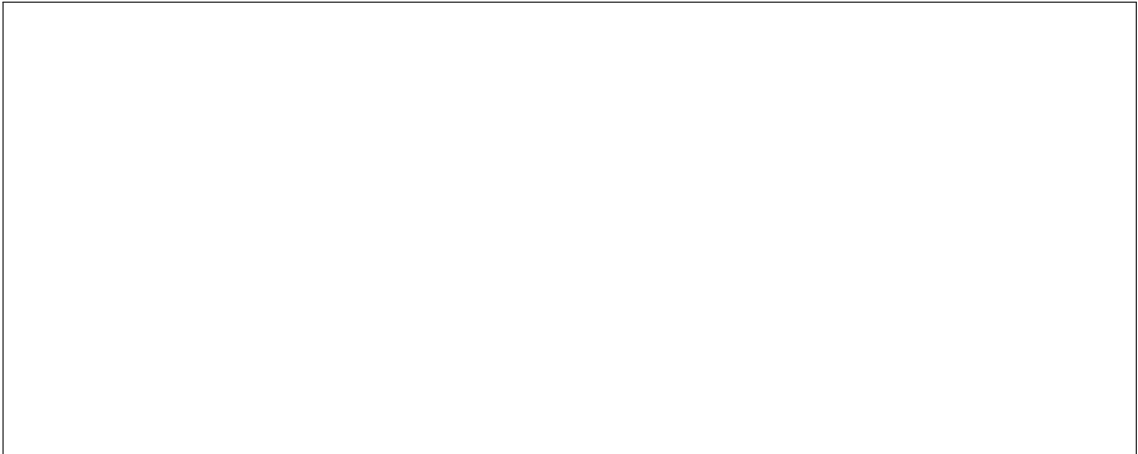
Använd olika moduler för synkroniseringsvipporna respektive tillståndsvipporna så att bara tillståndsvipporna nollställs när resethknappen trycks ned. Synkroniseringsvipporna ska aldrig nollställas. Notera att D-vipporna i labbet bara kan nollställas asynkront vilket begränsar möjliga kodningar av tillstånden.

Konstruera sekvenskretsen med D-vippor och PROM. Insignalerna x_1 och x_0 måste synkroniseras.

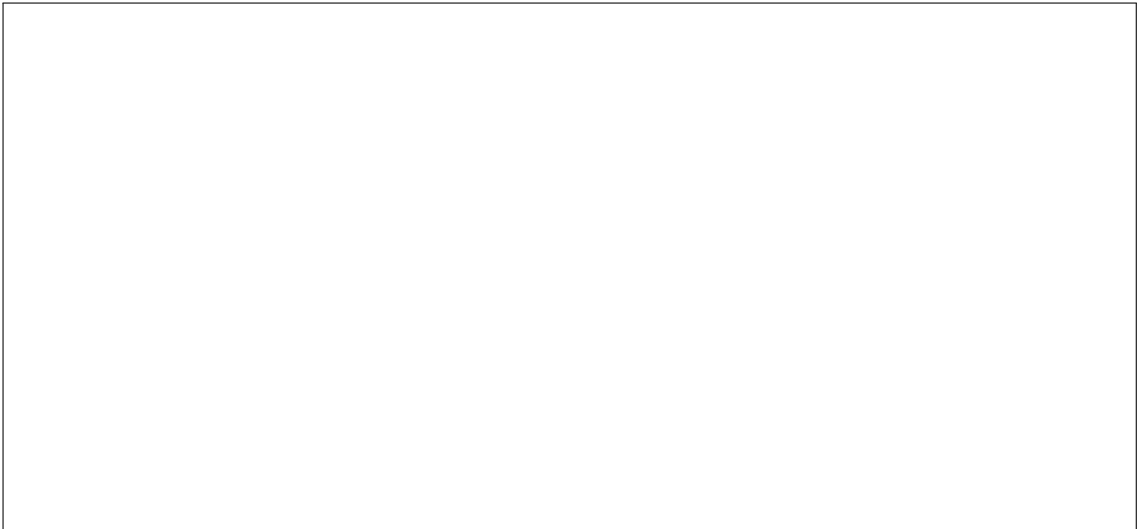
Tips: Spara anteckningarna från konstruktionen då dessa kommer till användning i laboration 3.

Förberedelseuppgifter:

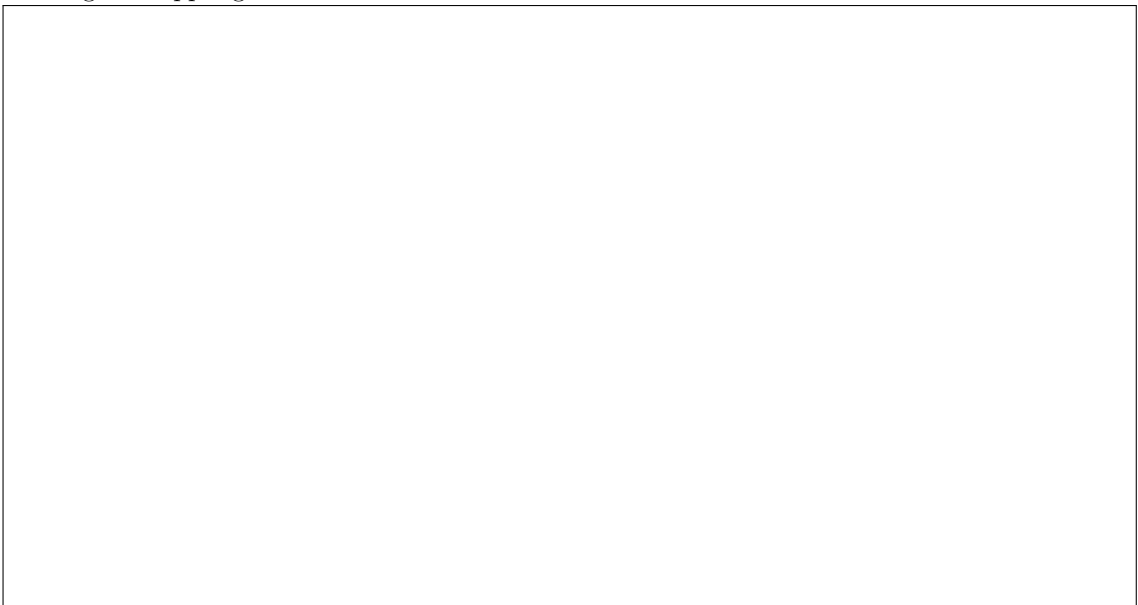
(a) Rita tillståndsdigram:

A large, empty rectangular box intended for drawing a state diagram.

(b) Rita tillståndstabell:

A large, empty rectangular box intended for drawing a state table.

(c) Rita logiskt kopplingsschema och minnesinnehåll i PROM:en:

A large, empty rectangular box intended for drawing a logic connection diagram and PROM memory content.

Laborationsuppgifter:

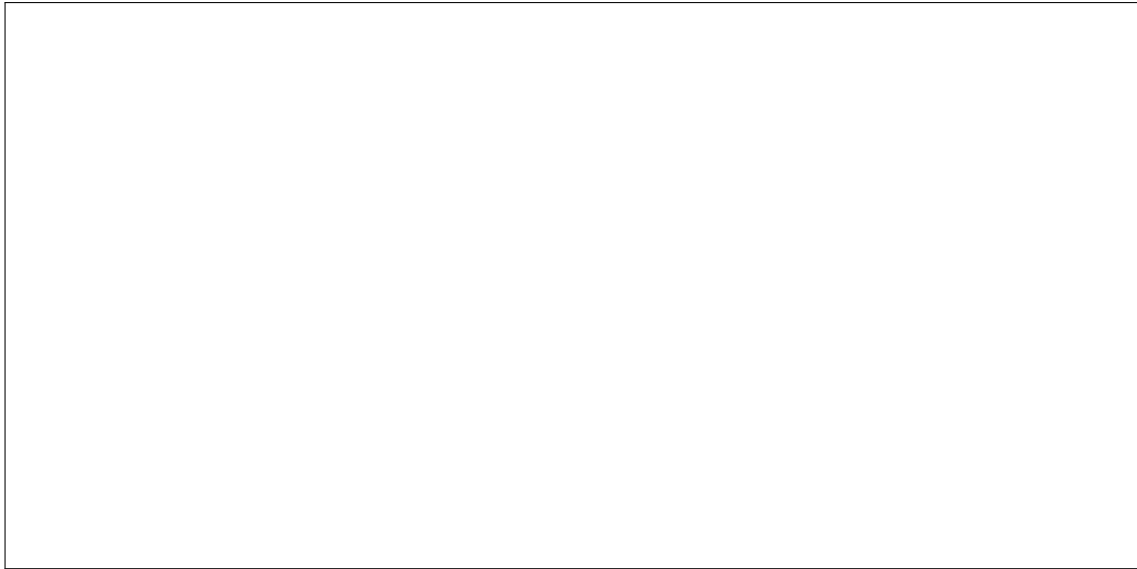
- (d) Koppla den förberedda sekvenskretsen. Kontrollera först funktionen genom manuell klockning från en studsfri tryckomkopplare. Anslut därefter kristalloscillatorn (på spänningsskenan) inställd på 8 MHz. För att kunna se hur nätet beter sig ska tillståndvariablerna anslutas till var sin lysdiod, alternativt till en logikprob.

När uppgiften är redovisad och godkänd ska PROM:en nollställas genom att i PROG-mode samtidigt trycka på de tre röda knapparna.

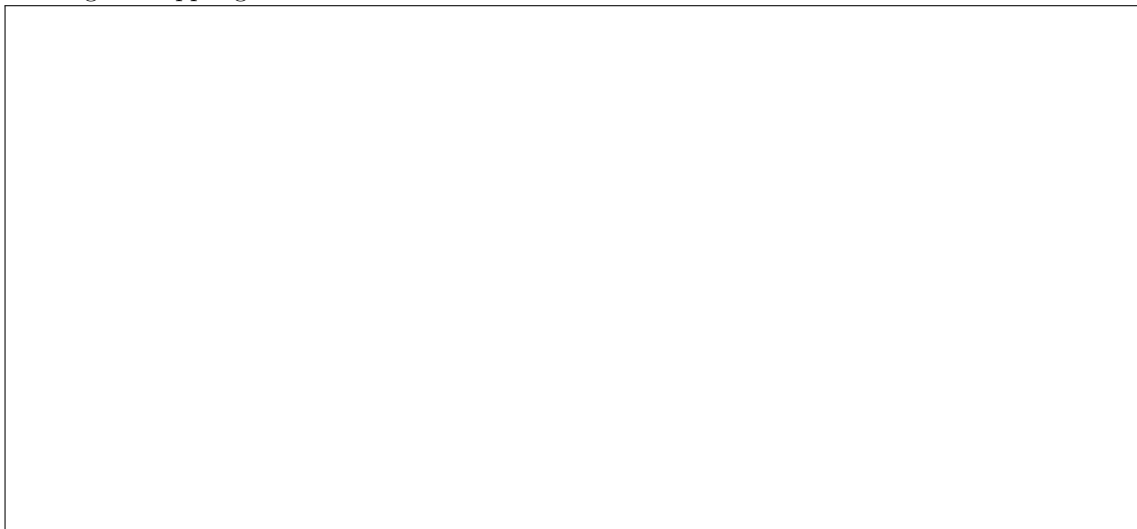
Uppgift 2.2. Lös uppgift 2.1 men nu med hjälp av D-vippor, NAND-grindar och inverterare. Lösningen ska även nu vara enligt Moore vilket gör att tillståndstabellen i förberedelseuppgift 2.1(b) kan återanvändas.

Förberedelseuppgift:

- (a) Rita Karnaughdiagram och beräkna minimala uttryck:



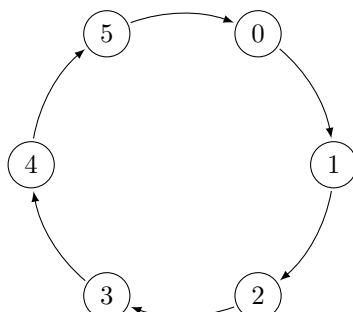
- (b) Rita logiskt kopplingsschema:

**Laborationsuppgift:**

- (c) Koppla upp kretsen och verifiera dess funktion.

Uppgift 2.3. Du ska undersöka den synkrona dekadräknaren 74LS160 i denna uppgift. Till din hjälp finns beskrivningen i avsnitt 2.1 samt databladet för räknaren.

- (a) Koppla in dekadräknaren 74LS160 så att den räknar autonomt med sex räknelägen enligt

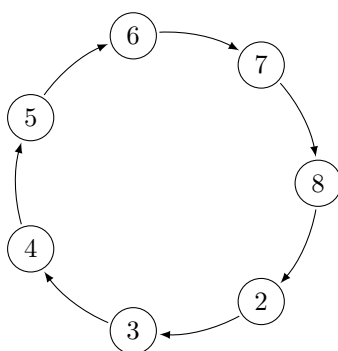


För att generera insignaler till räknaren får valfria grindar och inverterare användas. För övergången $5 \rightarrow 0$ får clear-ingången inte användas eftersom denna ingång är asynkron. Räknaren klockas manuellt från den studs fria tryckomkopplaren. Modulerna med endast en sju-segmentsdisplay är avkodade, dvs de visar den hexadecimala siffra som svarar mot det binära 4-bitarstal som finns på ingångarna. Använd en av dessa för att visa räkneläget.

Förberedelseuppgift: Rita logiskt kopplingsschema:

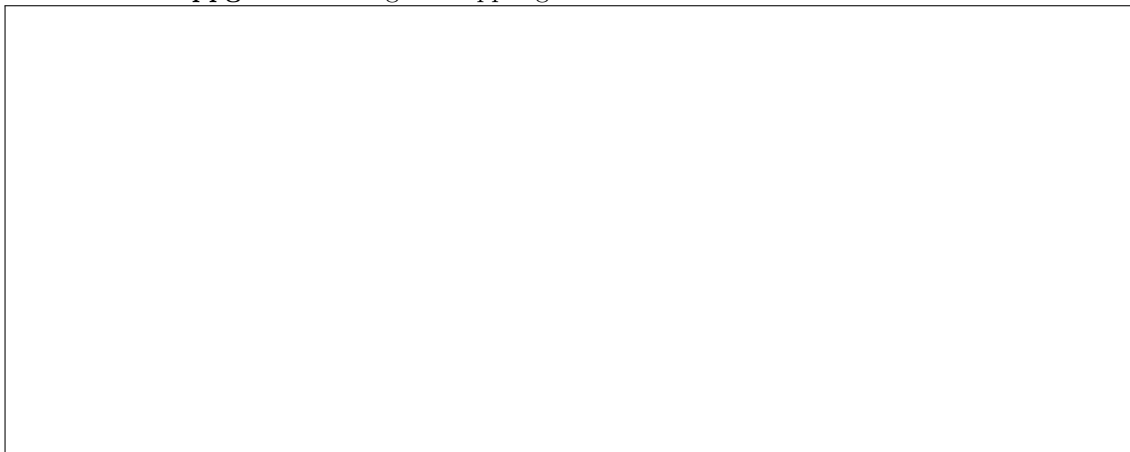
Laborationsuppgift: Koppla upp och verifiera kretsens funktion.

- (b) Koppla upp en räknare med sju räknelägen enligt



Använd så få NAND-grindar och inverterare som möjligt. Räknaren klockas manuellt.

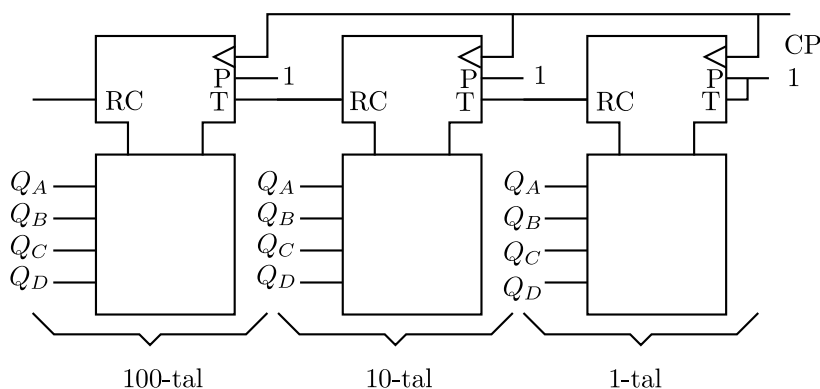
Förberedelseuppgift: Rita logiskt kopplingsschema:



Laborationsuppgift: Koppla upp och verifiera kretsens funktion.

- (c) Dekadräknaren 74LS160 kan seriekopplas (kaskadkopplas) så att en räknare med flera deka-der erhålls. Konstruera en krets som räknar decimalt från 0 till 999 för att sedan börja om på 0 och räkna upp gång på gång genom att kaskadkoppla tre dekadräknare. Vid kaskadkoppling används Enable-ingångarna, P och T, samt utgången Ripple Carry.

Laborationsuppgift: Koppla upp en räknare med tre deka-der enligt följande kopplings-schemat och visa resultatet på de tre avkodade sju-segmentsdisplayerna:

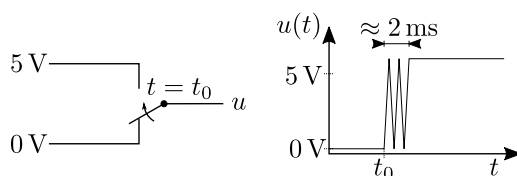


Kontrollera funktionen. Klockpulser tas från den speciella klockgeneratorn.

Tips: Dekadräknaren kan användas i uppgift 2.4.

Uppgift 2.4. Valfri. Uppmätning av effekter från oavstudsade skjutomkopplare.

När en mekanisk omkopplare sluts eller bryts uppstår alltid kontaktstudsar. När omkopplaren skjuts från logiskt noll till logiskt ett erhålles slutning till +5 V enligt:



När skjutomkopplaren förs från logiskt ett till logiskt noll erhålls slutning till jord.

Du ska utföra mätningar på de fyra skjutomkopplarna som finns monterade på en modul. Signalen från skjutomkopplaren ska användas som klockpuls till den dekadräknare (3 dekader) som du kopplade upp i uppgift 2.3(c). Komplettera dekadräknaren med manuell nollställning.

Gör fem mätningar på varje skjutomkopplare. Nollställ räknaren efter varje mätning. Fyll i tabellen.

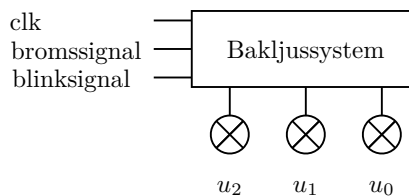
Skjutomkopplare							
1		2		3		4	
0 → 1	1 → 0	0 → 1	1 → 0	0 → 1	1 → 0	0 → 1	1 → 0

Gör om mätningen med de studs fria skjutomkopplarna.

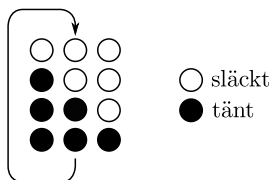
Slutning 0 → 1. Dekadräknaren visar:

Slutning 1 → 0. Dekadräknaren visar:

Uppgift 2.5. Konstruera en synkron sekvenskrets som styr de tre högra baklamporna på en Ferrari F50.



Systemet har två asynkrona insignaler en bromssignal och en blinksignal båda aktivt höga och tre utsignaler (u_2, u_1, u_0) kopplade till lamporna så att hög spänning tänds motsvarande lampan. Bakbelysningen har 3 driftlägen, blinker, broms eller släckt. Blinkern ska genomlöpa följande sekvens

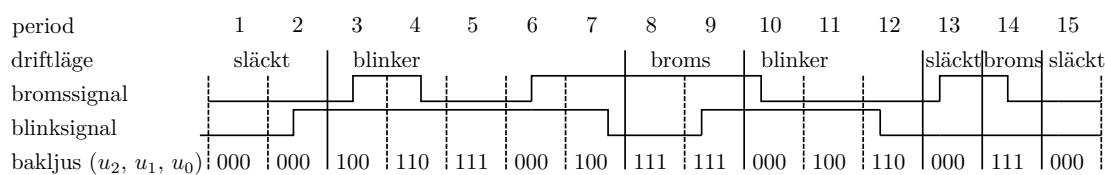


I bromsläget ska alla lampor vara tända och i det släckta läget ska som namnet indikerar alla lampor vara släckta. Lägesvalet styrs av insignalerna på följande sätt

$$(\text{bromssignal}, \text{blinksignal}) = \begin{cases} (0, 0) & \text{släckt} \\ (1, 0) & \text{broms} \\ (-, 1) & \text{blinker} \end{cases}$$

Notera att om både broms- och blink-signalen är aktiva samtidigt så ska blinkern vara aktiverad, bromsljuset kommer då att visas på bilens vänstra sida som ni inte behöver konstrueras.

Systemets exakta beteende vid bland annat övergång mellan lägena ses i följande tidsdiagram



där period numrerar klockintervallen, driftläge indikerar vilket läge som är aktivt, broms- och blink-signal är de två asynkrona insignalerna till systemet och bakljus de tre utsignalerna som visar vilka bakljus som är tända. Period 1, 13 och 15 visar exempel på att alla lampor är släckta när varken broms eller blinker är aktiverad. Period 3 visar hur blinksekvensen startas från släckta ljus och 10 hur blinksekvensen startas under inbromsning. Period 4 visar att blinksekvensen inte påverkas om bromsen samtidigt är aktiv. Period 7 visar att blinksekvensen upprepas så länge som blinkern är aktiverad. Period 8, 9 och 14 visar att alla ljus är tända när bara bromssignalen är aktiverad.

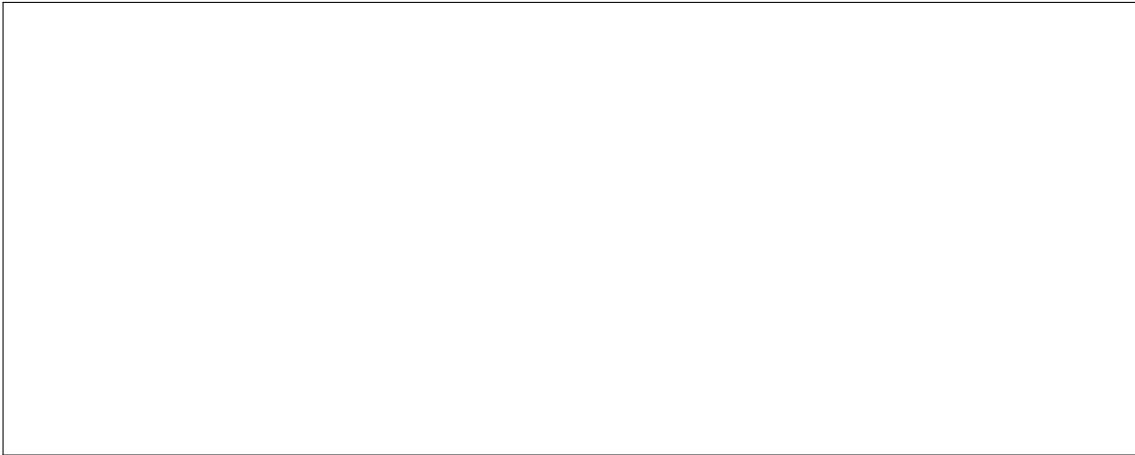
Använd valfria kretsar ur labsatsen, men det är rekommenderat att träna på den generella designmetoden inför laxen och då räcker det med PROM och D-vippor. Det går faktiskt att lösa uppgiften med bara ett PROM. Använd lysdioder för att simulera de tre lamporna. Insignalerna styrs med avstudsade skjutomkopplare. Insignalerna **behöver inte synkroniseras** och kretsen ska vara av Moore-typ. Klockfrekvensen skall vara cirka 1 Hz vilket förstås ger upphov till maximalt en sekunds fördröjning på t ex bromsljuset, men det kan ni bortse från. Vid verifiering och examination ska en tryckomkopplare användas för att manuellt klocka kretsen. Testa själva att er krets fungerar exakt som i tidsdiagrammet innan uppgiften examineras.

Förberedelseuppgift:

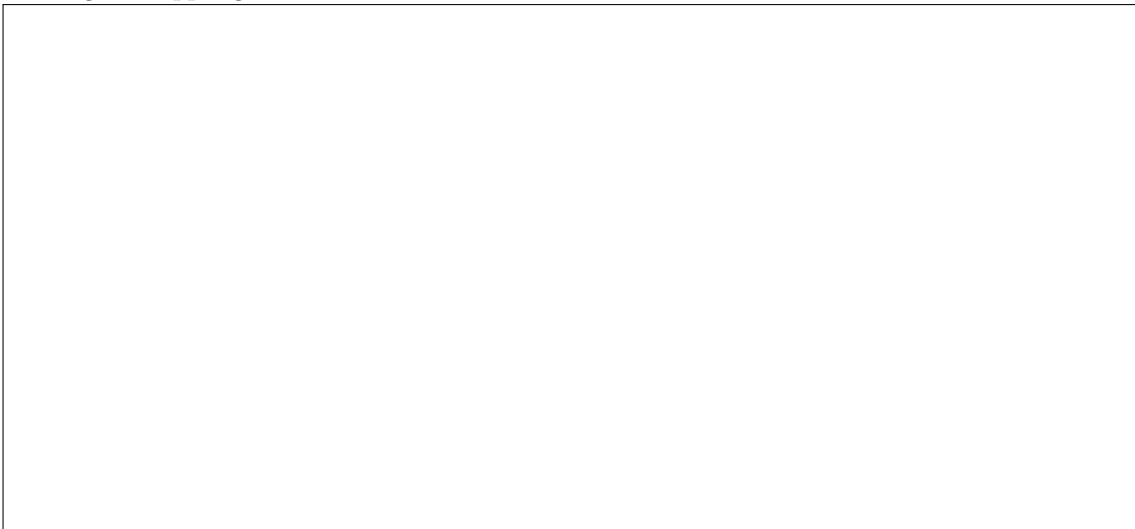
(a) Rita tillståndsdigram:



(b) Rita tillståndstabell:

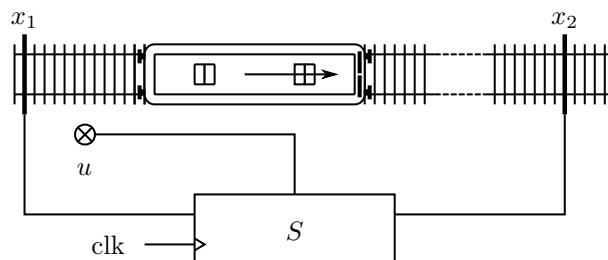


(c) Rita logiskt kopplingsschema:

**Laborationsuppgift:**

(d) Koppla upp och verifiera kretsens funktion.

Uppgift 2.6. Valfri. På en enkelriktad järnvägssträcka (tågen rör sig från vänster till höger) vill man ha ett system, i form av en synkron sekvenskrets S , som varnar om två tåg kör för nära varandra. Man har därför i banan placerat två givare på ett inbördes avstånd som är lika med säkerhetsavståndet. Givarna lämnar signalerna x_1 och x_2 . Befinner sig ett tåg över givaren är $x_i = 1$, annars är $x_i = 0$.



Om någon del av ett tåg befinner sig mellan de två givarna och ett nytt tåg når givare x_1 ska en tidigare släckt ($u = 0$) stopplampa tändas ($u = 1$). Det bakre tåget förutsätts då omedelbart tvärnita och stanna. Ett tåg är så långt, kör så sakta och stannar så snabbt att det fortfarande står över givare x_1 när det stannat. Först sedan det främre tåget helt passerat givare x_2 släcks stopplampan ($u = 0$) och det bakre tåget kan fortsätta sin färd. **(Att tända stopplampan så fort någon del av ett tåg befinner sig mellan givarna ger inte korrekt funktion).**

Ersätt givarna med två studsria skjutomkopplare. Anslut u till en lysdiod. För felsökningsändamål är det lämpligt att också koppla tillståndsvariablerna till lysdioder. Klockgenerators frekvens ska kunna varieras från 1 Hz till 1 kHz. Insignalerna måste synkroniseras.

Observera att ett tåg kan vara såväl längre som kortare än avståndet mellan givarna. Fallet att ett tåg är exakt lika långt som avståndet mellan givarna behöver inte beaktas.

Logiskt kopplingschema:

