Tentamen

TSEA22 Digitalteknik
23 mars, 2020, kl. 8.00-12.00
Lisaminlämningen stänger kl. 12.10

Tillåtna hjälpmedel: Inga.

Ingen kommunikation med annan person än kursansvarig får ske under skrivtiden.

Mobil, surfplatta och/eller dator får användas för att hämta tentan och skicka in lösningarna via kurshemsidan på Lisam. Skicka in lösningarna som en pdf-fil. Döp filen med ditt personnummer och namn på formatet: ÅÅMMDD-XXXX_Efternamn_Förnamn.pdf

Om inlämning via lisam inte fungerar, skicka filen till:
mattias.krysander@liu.se med titel TENTA

Eventuell information som berör alla skrivande kommer att skickas ut via e-post till studentadressen.

Ansvarig lärare: Mattias Krysander, 073 - 270 18 25

Totalt 50 poäng.
Preliminära betygsgränser:
Betyg 3: 21 poäng
Betyg 4: 31 poäng
Betyg 5: 41 poäng

Uppgift 1. Talkoder och Boolesk algebra.

a) Skriv det decimala talet 2020 hexadecimalt. (1 poäng)

b) Omvandla det decimala talet 0.5625 till binärform. (1 poäng)

c) Förenkla uttrycket $be + bc'e + a'bd + a'de'$. Ange vilka räknelagar du använder. (2 poäng)

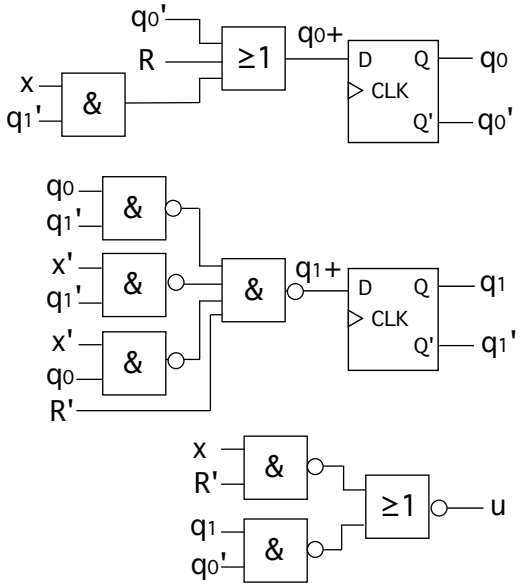
Lösning.

a) $2020_{10} = 7E4_{16}$

b) $0.5625_{10} = 0.1001_2$

c) $be + bc'e + a'bd + a'de' = [\text{absorption}] = be + a'bd + a'de' = [\text{consensus}] = be + a'de'$

Uppgift 2. Den okända sekvenskretsen. Figuren nedan visar en sekvenskrets.



a) Skriv uttryck för q_1^+ och q_0^+. Förenkla så långt som möjligt. (2 poäng)

b) Skriv upp tillståndtabell och rita tillståndsdiagram för kretsen. (4 poäng)

Lösning.

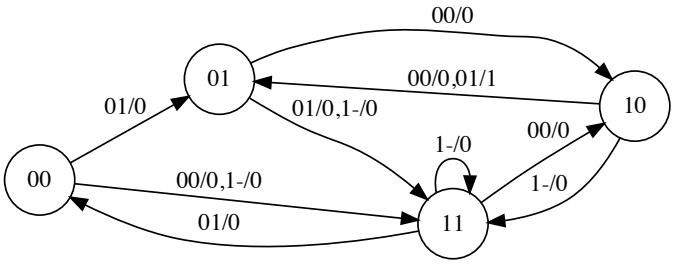
a)

$$q_0^+ = q_0' + q_1'x + R$$
$$q_1^+ = ((q_1'q_0)'(q_1'x')'(q_0x')'R')' = q_1'q_0 + q_1'x' + q_0x' + R$$
$$u = ((xR')' + (q_1q_0')')' = q_1q_0'R'x$$

b) Tillståndstabell:

	$q_1^+q_0^+/u$			
q_1q_0	$Rx = 00$	$Rx = 01$	$Rx = 11$	$Rx = 10$
00	11/0	01/0	11/0	11/0
01	10/0	11/0	11/0	11/0
11	10/0	00/0	11/0	11/0
10	01/0	01/1	11/0	11/0

Tillståndsgrafen blir där nodmarkeringarna anger q_1q_0 och bågmarkeringarna Rx/u:



Uppgift 3. Kombinationskrets: åter till djurparken: Tigerdetektorn. Ett digitalt system i en djurpark skall klassificera olika typer av djur som passerar en rad sensorer x_3, x_2, x_1 och x_0. Om de ger mönster med varannan etta och nolla så skall svaret TIGER ges. Om det däremot är exakt två ettor och dom är i rad så är svaret LITET DJUR. Till slut, om det är exakt tre ettor och dom är i rad så är svaret STORT DJUR. Enbart nollor skall ge svaret INGET DJUR. Systemet är utformat så att vi kan bortse från transienter i övergångar. Fall som inte är specificerade ovan anses inte kunna hända och ignoreras. Några exempel:

- $(x_3, x_2, x_1, x_0) = (1, 1, 0, 1) \Rightarrow$ don't care
- $(x_3, x_2, x_1, x_0) = (1, 0, 0, 0) \Rightarrow$ don't care
- $(x_3, x_2, x_1, x_0) = (0, 1, 0, 1) \Rightarrow$ TIGER

Kodning av utsignaler:

- INGET DJUR: $(u_1, u_0) = (0, 0)$
- LITET DJUR: $(u_1, u_0) = (0, 1)$
- TIGER: $(u_1, u_0) = (1, 0)$
- STORT DJUR: $(u_1, u_0) = (1, 1)$

Konstruera två kretsar, en lösning som enbart använder NAND-grindar och en som enbart använder NOR-grindar. Båda lösningarna skall ha grinddjup 2 och använda så få grindar som möjligt. I båda fallen får inverterare användas för att invertera insignaler. Inverterarna räknas inte in i grinddjupet. För full poäng krävs funktionstabell, Karnaughdiagram, minimala uttryck och uppritade kretsar.
(10 poäng)

Lösning.

x_3	x_2	x_1	x_0	u_1	u_0
0	0	0	0	0	0
0	0	0	1	-	-
0	0	1	0	-	-
0	0	1	1	0	1
0	1	0	0	-	-
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	-	-
1	0	0	1	-	-
1	0	1	0	1	0
1	0	1	1	-	-
1	1	0	0	0	1
1	1	0	1	-	-
1	1	1	0	1	1
1	1	1	1	-	-

NAND-realisering:

u_1

x_3, x_2 \ x_1, x_0	00	01	11	10
00	0	-	0	-
01	-	1	1	0
11	0	-	-	1
10	-	-	-	1

u_0

x_3, x_2 \ x_1, x_0	00	01	11	10
00	0	-	1	-
01	-	0	1	1
11	1	-	-	1
10	-	-	-	0

$u_1 = (x_2x_0 + x_3x_1)'' = ((x_2x_0)'(x_3x_1)')'$ $\quad$ $u_0 = (x_3'x_1 + x_3x_2)'' = ((x_3'x_1)'(x_3x_2)')'$

Det krävs en inverterare och 6 NAND-grindar.

NOR-realisering:

u_1

x_3,x_2 \ x_1,x_0	00	01	11	10
00	0	-	0	-
01	-	1	1	0
11	0	-	-	1
10	-	-	-	1

u_0

x_3,x_2 \ x_1,x_0	00	01	11	10
00	0	-	1	-
01	-	0	1	1
11	1	-	-	1
10	-	-	-	0

$$u_1 = ((x_3'x_2')'' + (x_1'x_0')'' + (x_3'x_0')'')' = \\ = ((x_3 + x_2)' + (x_1 + x_0)' + (x_3 + x_0)')'$$

$$u_0 = ((x_3'x_1')'' + (x_2'x_0')'')' = \\ = ((x_3 + x_1)' + (x_2 + x_0)')'$$

Det krävs 7 NOR-grindar.

Uppgift 4. Sekvenskrets. Konstruera en enkel 3-bitars aritmetisk logisk enhet (ALU) som lagrar ett nuvarande värde $q = (q_2, q_1, q_0)$ där q tolkas som ett binärt tal med q_2 som mest signifikant bit. Utsignalerna från ALUn är det lagrade värdet, dvs $q = (q_2, q_1, q_0)$. ALUn har två insignaler x och y som styr hur värdet q ska uppdateras.

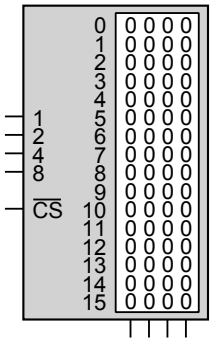
ALUn ska ha följande funktion:

- Insignal $(x, y) = (0, 0)$: Nollställ.
- Insignal $(x, y) = (0, 1)$: Räkna upp med 1.
- Insignal $(x, y) = (1, 0)$: Halvera talet.
- Insignal $(x, y) = (1, 1)$: Dubblera talet.

Den skall inte räkna wrap-around, utan stanna på $q = 7$ om någon operation överskrider det. Om halvering ger rest så ignoreras denna. En exempelsekvens ses nedan:

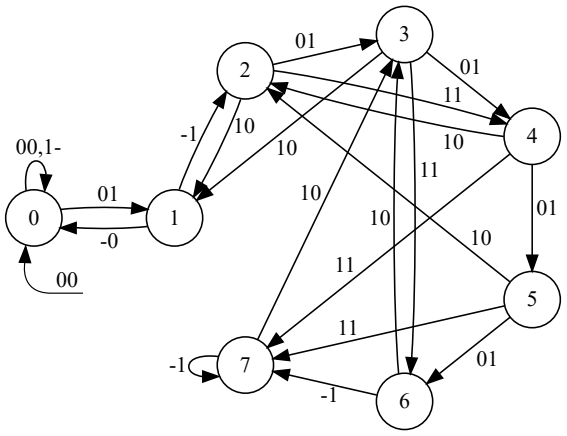
x:	0	0	0	1	1	1	1	0	1	1	1	1	1	0	1	1	0	0	0	0
y:	1	1	1	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0
q:	0	1	2	3	1	2	4	7	7	3	1	0	0	0	1	2	4	5	6	7	7	0

Till er konstruktion har ni D-vippor och två ROM av den typen som ni använde på laborationerna, dvs



Insignaler behöver inte synkroniseras. För full poäng krävs tillståndsdiagram, tillståndstabell och uppritad krets. Asynkrona lösningar ger kraftiga poängavdrag. (10 poäng)

Lösning. Tillståndsdiagram med bågmarkeringar xy och nodmarkeringar q.



Tillståndstabellen blir

Nollställ:

xy	q	q^+
00	0	0
00	1	0
00	2	0
00	3	0
00	4	0
00	5	0
00	6	0
00	7	0

Räkna upp:

xy	q	q^+
01	0	1
01	1	2
01	2	3
01	3	4
01	4	5
01	5	6
01	6	7
01	7	7

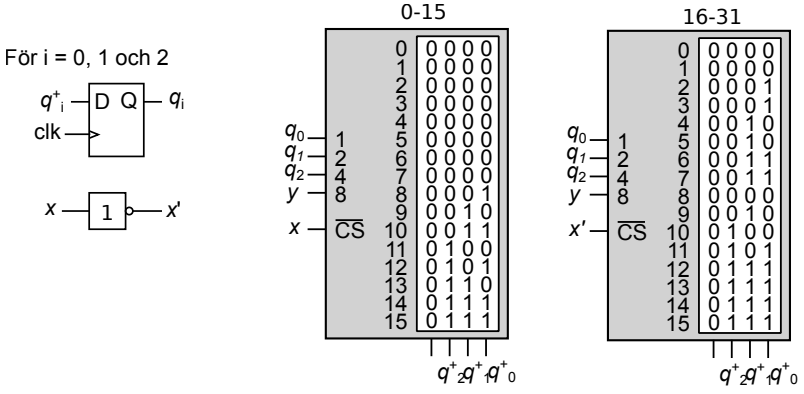
Halvera:

xy	q	q^+
10	0	0
10	1	0
10	2	1
10	3	1
10	4	2
10	5	2
10	6	3
10	7	3

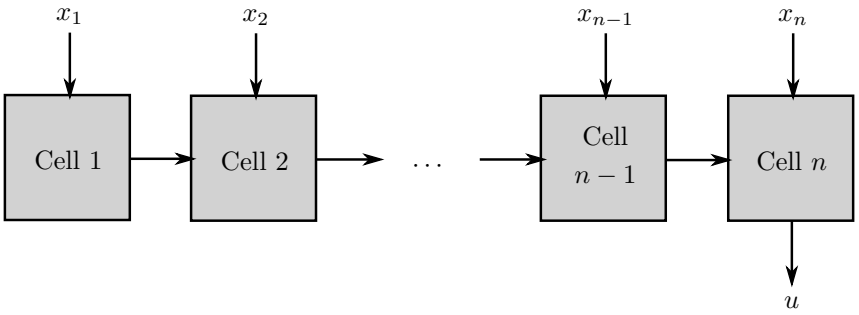
Dubblera:

xy	q	q^+
11	0	0
11	1	2
11	2	4
11	3	6
11	4	7
11	5	7
11	6	7
11	7	7

Kretsen blir:



Uppgift 5. Iterativ kombinationskrets. Konstruera en iterativ kombinatorisk krets med följande struktur:



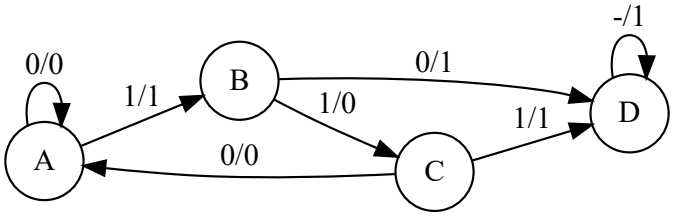
Kretsen ska detektera om det finns någon puls med längd $\neq 2$ klockcykler enligt följande exempel där n har valts till 10:

Ex 1:	$x = (x_1, x_2, \ldots, x_{10}) = (0000000000) \Rightarrow u = 0$
Ex 2:	$x = (0110000011) \Rightarrow u = 0$
Ex 3:	$x = (0000000001) \Rightarrow u = 1$
Ex 4:	$x = (1010011011) \Rightarrow u = 1$
Ex 5:	$x = (1111011011) \Rightarrow u = 1$

Exempel 1 visar insignaler utan pulser, dvs ingen puls med längd $\neq 2$ finns, alltså ska utsignalen vara $u = 0$. Exempel 2 visar en insignal med två pulser båda med längd 2 alltså ska utsignalen vara $u = 0$. Exempel 3 innehåller en puls med längd 1 och ska därför larma, dvs $u = 1$. Exempel 4 visar att det kan komma många pulser men det räcker med en puls av felaktig längd för att larma. Längden på felaktiga pulser kan anta vilken längd som helst och i exempel 5 illustreras detta med en puls av längd 4.

Ni har tillgång till AND-, OR-grindar och inverterare och kan anta att $n \geq 4$. För full poäng krävs tillståndsdiagram med minimalt antal tillstånd, tillståndstabell, minimerade uttryck för alla celler och kretsschema med minimerade celler. (10 poäng)

Lösning. Tillståndsdiagram med bågmarkeringar x/u.



Kodning alternativ 1: Binär
Starttillståndet är $q = 00$.

	$q_1^+ q_0^+ / u$	
$q_1 q_0$	$x = 0$	$x = 1$
00	00/0	01/1
01	11/1	10/0
11	11/1	11/1
10	00/0	11/1

Cell 1: $(q_1, q_0) = (0, 0)$

$$q_1^+ = 0$$
$$q_0^+ = x_1$$

Cell 2: $(q_1, q_0) \in \{(0, 0), (0, 1)\}$

$$q_1^+ = q_0$$
$$q_0^+ = q_0' x_2 + q_0 x_2'$$

Cell $k \in \{3, \ldots, n-1\}$:

$$q_1^+ = q_0 + \underline{q_1 x_k}$$
$$q_0^+ = q_0' x_k + q_0 x_k' + \underline{q_1 x_k}$$

Cell n:

$$u = q_0' x_n + q_0 x_n' + q_1 x_n$$

Det är möjligt att grinddela de understrukna termerna i den generella cellen. Detta ger $2n - 2$ inverterare, $2n - 4$ OR-grindar och $3n - 4$ AND-grindar.

Kodning alternativ 2: Gray
Starttillståndet är $q = 00$.

	$q_1^+ q_0^+ / u$	
$q_1 q_0$	$x = 0$	$x = 1$
00	00/0	01/1
01	10/1	11/0
11	00/0	10/1
10	10/1	10/1

Cell 1: $(q_1, q_0) = (0, 0)$

$$q_1^+ = 0$$
$$q_0^+ = x_1$$

Cell 2: $(q_1, q_0) \in \{(0, 0), (0, 1)\}$

$$q_1^+ = q_0$$
$$q_0^+ = x_2$$

Cell $k \in \{3, \ldots, n-1\}$:

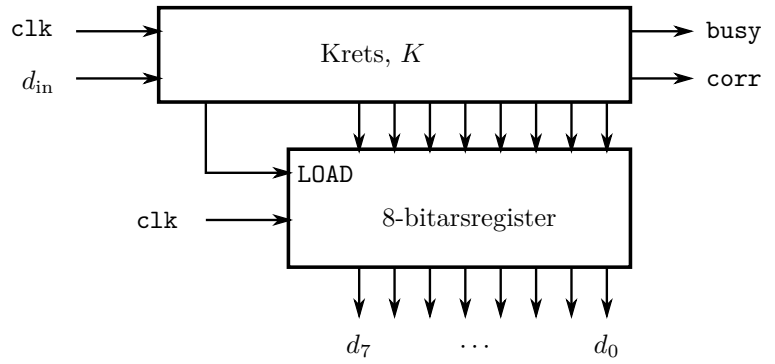
$$q_1^+ = q_1' q_0 + q_1 q_0' + q_1 x_k \text{ (eller } q_0 x_k)$$
$$q_0^+ = q_1' x_k$$

Cell n:

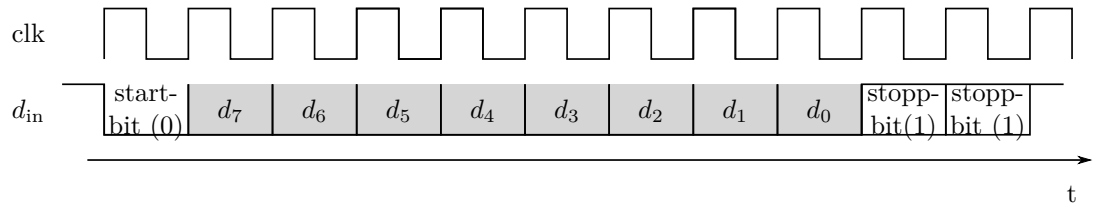
$$u = q_1' q_0 x_n' + q_1 q_0' + q_1 x_n + q_0' x_n$$

Detta ger $2n - 3$ inverterare, $n - 2$ OR-grindar och $4n - 8$ AND-grindar.

Uppgift 6. Seriell till parallell datakonvertering. Konstruera den *synkrona* sekvenskretsen, K i figuren nedan som läser in ett 8-bitarsord seriellt på ingången d_{in} och sparar det i ett 8-bitarsregister.

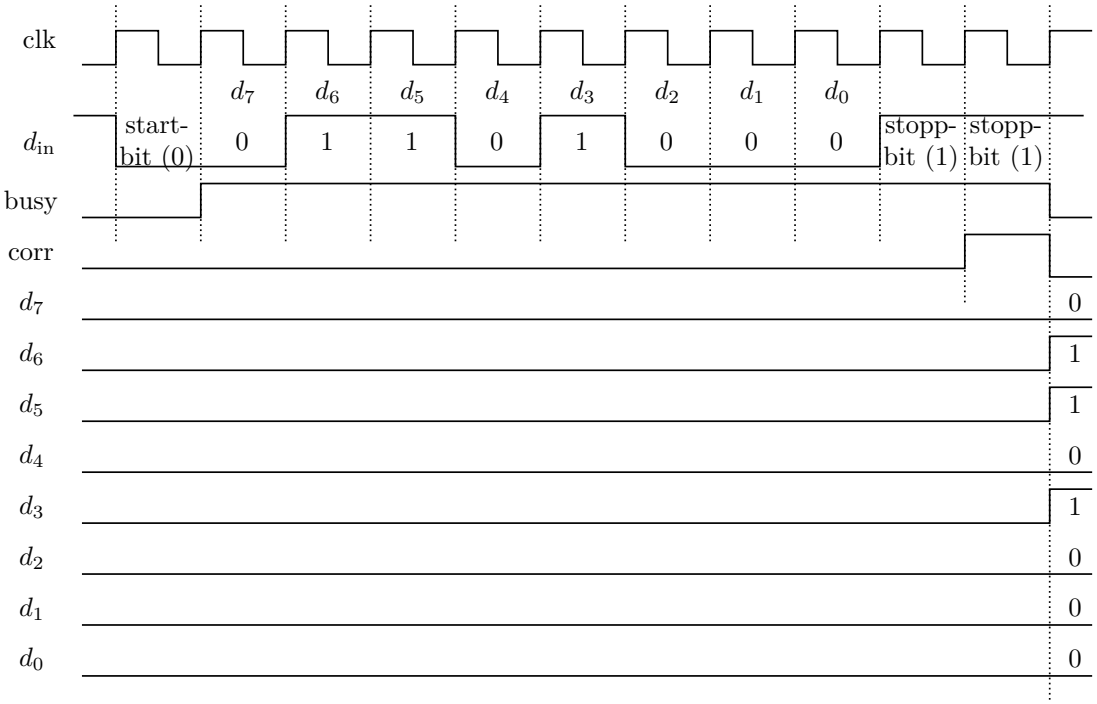


Formatet på meddelandet ser ut som följer:



Signalen d_{in} är normalt hög när inget meddelande mottas. Ett meddelande inleds med en låg startbit. Därefter kommer ordet 8 bitar d_7 först och d_0 sist. Meddelandet avslutas med 2 stoppbitar som båda är höga. Er uppgift är att konstruera krets, K i figuren som tar in data seriellt på d_{in} och sparar mottaget ord i 8-bitarsregistret. Utsignaler från kretsen K är `LOAD`-signalen och datasignalerna till registret samt en `busy`-signal som signalerar att ett meddelande mottas samt en `corr`-signal som indikerar att meddelandet är korrekt mottaget och kan sparas i registret.

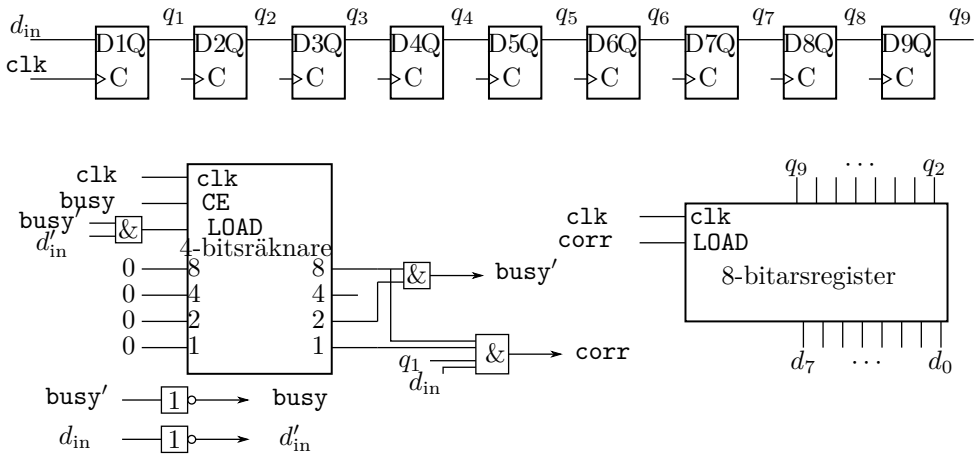
Ett tidsdiagram visar ett exempel på in- och ut-signaler.



Signalen `busy` är hög under pågående mottagning. Då kan inte en ny mottagning påbörjas. En mottagning är korrekt om stoppbitarna är höga och då ska signalen `corr` sättas hög och det mottagna ordet sparas i registret enligt tidsdiagrammet. Om inte båda stoppbitarna är höga ska signalen `corr` vara låg och registrets utdata ska bli oförändrat, dvs utdata ska ligga kvar ända tills dess ett nytt korrekt mottaget ord har registrerats.

Till er konstruktion har ni en 4-bitarsräknare med funktionerna count enable och load, D-vippor, grindar och inverterare. Insignalerna får antas vara synkroniserade med klockan. Meddelandena kommer inte regelbundet utan initieras av att insignalen går låg. För full poäng krävs ett kretsschema med en beskrivning hur kretsen ska fungera. Initialvillkor på räknare och eventuella vippor ska anges. Onödigt komplicerade lösningar och asynkrona lösningar ger poängavdrag. (10 poäng)

Lösning. Det finns många olika lösningar. Här presenteras en variant.



Kedjan med D-vipporna (skiftregister) sparar de senaste 9 bitarna på d_{in}. Initiera räknaren 10_{10}. Då kommer `busy` $= 0$, `CE` $= 0$ och `LOAD` $= 0$ tills d_{in} blir låg, dvs räknaren kommer vara kvar på 10 till dess startbiten på ett meddelande kommer, då kommer räknaren ladda in 0. När räknaren blir 0 blir `busy` $= 1$ och `CE` $= 1$, dvs räknaren börjar räkna upp. När `busy` $= 1$ kan inte räknaren laddas, dvs avbrytas under mottagande av ord. När räknaren kommer till 9 ligger först stoppbiten lagrad i q_1 och den andra finns på d_{in}, dvs `corr` $= 1$ om räknaren är 9 och $q_1 = 1$ och $d_{\text{in}} = 1$. Korrektsignalen kan direkt användas på load i registret, dvs `LOAD` $=$ `corr`. Vid denna tidpunkt ligger det mottagna ordet sparat i D-vipporna så att $d_i = q_{i+2}$ för $i \in \{0, 1, \ldots, 7\}$. Mottagningen slutar med att räknaren åter kommer till 10 vilket leder till att `busy` $= 0$ och räknaren stannar beredd att ta emot ett nytt meddelande.