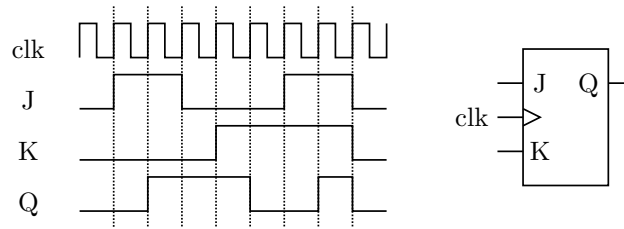


Försättsblad till skriftlig tentamen vid Linköpings universitet



Datum för tentamen	2018-03-13
Sal (5)	G32(35) G33(3) G34(2) <u>TER4(104)</u> TERF(1)
Tid	8-12
Kurskod	TSEA22
Provkod	TEN1
Kursnamn/benämning Provnamn/benämning	Digitalteknik Skriftlig tentamen
Institution	ISY
Antal uppgifter som ingår i tentamen	6
Jour/Kursansvarig Ange vem som besöker salen	Mattias Krysander
Telefon under skrivtiden	073-2701825
Besöker salen ca klockan	9 och 11
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Mattias Krysander 073-2701825 mattias.krysander@liu.se
Tillåtna hjälpmedel	Inga
Övrigt	Totalt: 50 poäng Preliminära betygsgränser: Betyg 3: 21 poäng Betyg 4: 31 poäng Betyg 5: 41 poäng Visning 10.00-11.00 den 10/3 i Datortekniks bibliotek.
Antal exemplar i påsen	

Uppgift 1. Pelle har hittat en gammal krets i labbet som innehåller en JK-vippa. Han vill bygga om kretsen med D-vippor och grindar. För att undersöka hur JK-vippan fungerar mäter han insignalerna J och K och utsignalen/tillståndet Q och erhåller följande tidsdiagram:



- a) Beskriv för de fyra insignalkombinationerna, dvs $(J,K) \in \{(0,0), (0,1), (1,0), (1,1)\}$, vilken funktion vippan har, dvs hur nästa tillstånd Q^+ blir som funktion av tillståndet Q ? (2 poäng)
- b) Konstruera en krets som har samma funktion som en JK-vippa men med en D-vippa samt valfria grindar. Visa tillståndsdigram, tillståndstabell och kretsschema. Onödigt komplicerade lösningar ger poängavdrag. (3 poäng)

Lösning.

- a) – $(J,K) = (0,0)$: $q^+ = q$, dvs tillståndet förändras ej.
 – $(J,K) = (0,1)$: $q^+ = 0$, dvs tillståndet nollställs.
 – $(J,K) = (1,0)$: $q^+ = 1$, dvs tillståndet sätts till 1.
 – $(J,K) = (1,1)$: $q^+ = q'$, dvs tillståndet byts.

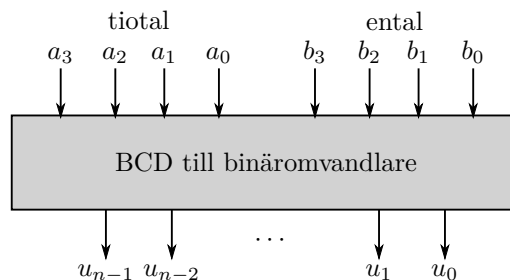
b) Tillståndstabell

JK	$q = 0$	$q = 1$
00	0	1
01	0	0
11	1	0
10	1	1

Detta ger följande uttryck:

$$q^+ = Jq' + K'q$$

Uppgift 2. Konstruera en kombinationskrets som gör om ett BCD-kodat tal 0-99 till ett binärkodat tal med följande in- och ut-signaler:



Högst index svarar mot mest signifikant bit.

- a) Hur många bitar n behövs för att i binärkod kunna representera det decimala talet 99? (1 poäng)

- b) Med det n som fastställdes i a)-uppgiften konstruera kretsen. Till ert förfogande har ni endast heladderare (FA) och halvadderare (HA). Använd bara heladderare om det är nödvändigt eftersom halvadderare är komponentsnålare.

Använd följande uttryck för u vid konstruktion av kretsen

$$u = 10a + b = (8 + 2)a + b = 8a + 2a + b \quad (1)$$

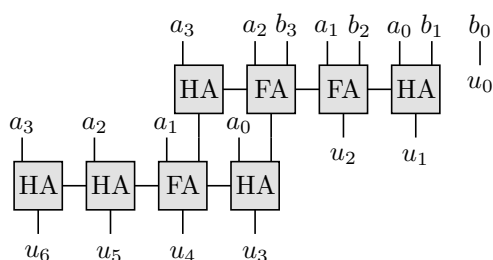
där $a = (a_3, a_2, a_1, a_0)$, $b = (b_3, b_2, b_1, b_0)$ och $u = (u_{n-1}, u_{n-2}, \dots, u_1, u_0)$.

Lösning.

- a) $n = 7$ bitar behövs eftersom

$$2^7 - 1 = 127 > 99 \geq 2^6 = 64$$

- b) Multiplikation med 2 är detsamma som att vänsterskifta binärkodningen 1 bit. Detta ger att det givna uttrycket kan realiseras som



I den första nivån beräknas $x = 2a + b$ och i den undre $u = 8a + x$. Om bara två signaler ska adderas används HA. Utsignalerna från de vänstraste HA är alltid noll eftersom

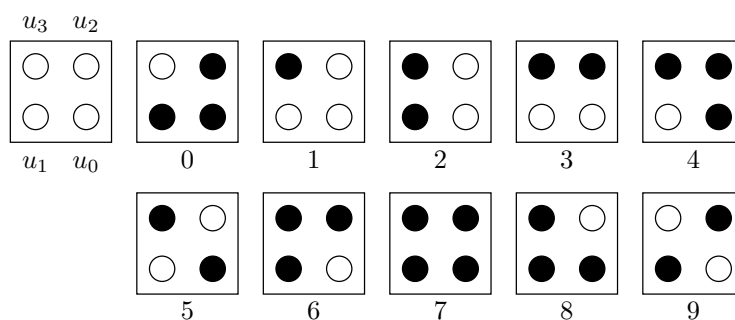
$$2a + b \leq 27 < 2^5 = 32 \quad \text{kräver bara 5 bitar}$$

samt

$$8a + 2a + b \leq 99 < 2^7 = 128 \quad \text{kräver bara 7 bitar}$$

och behöver därför inte kopplas in.

Uppgift 3. Konstruera en krets som har omvandlar talen 0-9 till punktskrift. Punktskrift definieras av



Insignalerna till kretsen ges i BCD-kod av $x = (x_3, x_2, x_1, x_0)$ där x_3 är mest signifikant bit och svart prick motsvarar $u_i = 1$. Ni behöver bara konstruera kretsarna som skapar utsignalerna u_2 , u_1 och u_0 , dvs ej kretsen för u_3 . Till er konstruktion får ni endast använda NAND-grindar. Inverterade insignalers antas vara tillgängliga. För full poäng krävs en korrekt funktionstabell, Karnaughdiagram, minimala Booleska uttryck och ett kretsschema. (10 poäng)

Lösning. Funktionstabell:

x	u
0	0111
1	1000
2	1010
3	1100
4	1101
5	1001
6	1110
7	1111
8	1011
9	0110
f.ö.	—

u_2

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

$$u_2 = x'_3 x'_1 x'_0 + x_1 x_0 + x_3 x_0 + x_2 x_1 = ((x'_3 x'_1 x'_0)' (x_1 x_0)' (x_3 x_0)' (x_2 x_1)')'$$

u_1

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

$$u_1 = x_3 + x'_2 x'_0 + x_2 x_1 = (x'_3 (x'_2 x'_0)' (x_2 x_1)')'$$

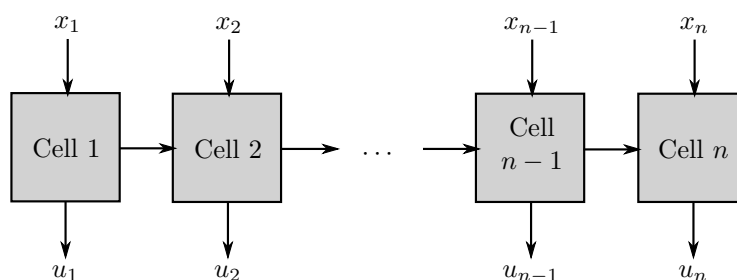
u_0

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

$$u_0 = x'_1 x'_0 + x_2 x_0 = ((x'_1 x'_0)' (x_2 x_0)')'$$

Totalt krävs 10 NAND-grindar. Den understrukna termen grinddelas.

Uppgift 4. Konstruera ett iterativt kombinatoriskt nät med struktur enligt:

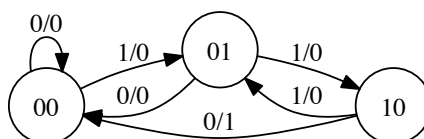


Utsignal u_i ska vara 1 om och endast om $x_i = 0$ och insignalerna till cellerna direkt till vänster om cell i består av en grupp av ett jämnt antal ettor. Ett exempel för $n = 20$ är

x_i : 0 1 1 0 0 1 1 1 0 1 1 1 1 0 0 0 0 1 0 0
 u_i : 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

Ni har tillgång till AND-, OR-grindar och inverterare. För full poäng krävs tillståndsdigram med minimalt antal tillstånd, tillståndstabell, Booleska minimerade uttryck för alla celler och kretsschema med minimerade celler. (10 poäng)

Lösning. Tillståndsdigram med nodmarkeringar q_1q_0 och bågmarkeringar x_i/u_i .



Starttillståndet är $q = 00$.

q_1q_0	$q_1^+q_0^+/u_i$	
	$x_i = 0$	$x_i = 1$
00	00/0	01/0
01	00/0	10/0
11	-/-	-/-
10	00/1	01/0

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

$$q_0^+ = x_1$$

$$u_1 = 0$$

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

$$q_1^+ = q_0x_2$$

$$q_0^+ = q_0'x_2$$

$$u_2 = 0$$

Cell $k \in \{3, \dots, n-1\}$: $(q_1, q_0) \in \{(0, 0), (0, 1), (1, 0)\}$

$$q_1^+ = q_0x_k$$

$$q_0^+ = q_0'x_k$$

$$u_k = q_1x_k'$$

Cell $n - 1$:

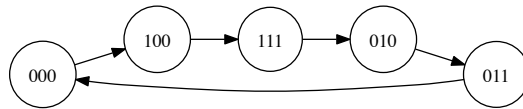
$$q_1^+ = q_0 x_{n-1}$$

$$u_{n-1} = q_1 x'_{n-1}$$

Cell n

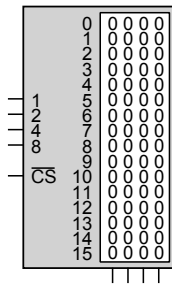
$$u_n = q_1 x'_n$$

Uppgift 5. Konstruera en synkron 3-bitarsräknare med count enable CE, aktivt hög. När CE = 1 ska följande sekvens genomlöpas för tillståndet $q = (q_2, q_1, q_0)$:



Notera att nästa tillstånd för 001, 101 och 110 är ospecificerat.

Till ert förfogande har ni D-vippor och PROM av typen

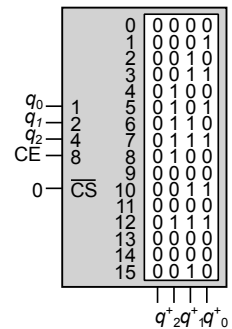
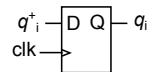


där chip select (\overline{CS}) är aktivt låg. För full poäng krävs korrekt tillståndsdigram, tillståndstabell och kretsschema. Insignalen behöver inte synkroniseras. (10 poäng)

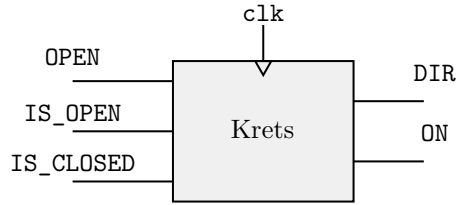
Lösning.

CE	$q_2 q_1 q_0$	$q_2^+ q_1^+ q_0^+$
0	$q_2 q_1 q_0$	$q_2 q_1 q_0$
1	000	100
	001	—
	010	011
	011	000
	100	111
	101	—
	110	—
	111	010

För $i = 0, 1$ och 2



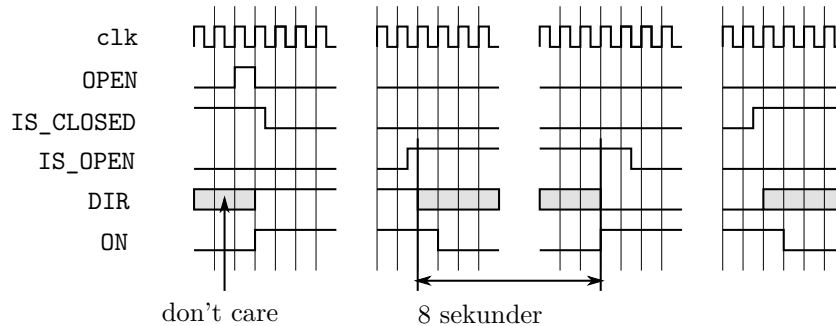
Uppgift 6. En synkron sekvenskrets för styrning av en hissdörr ska konstrueras.



Insignalerna till kretsen är en styrsignal OPEN som om den är 1 innebär att hissdörren ska öppnas, IS_CLOSED en sensorsignal som om den är 1 indikerar att dörren är stängd samt IS_OPEN en sensorsignal som om den är 1 indikerar att dörren är öppen. Kretsens utsignaler är ON som om den är 1 kör motorn som öppnar och stänger hissdörren samt DIR som anger om hissdörren öppnas eller stängs (direction).

Kretsen ska styra dörren så att om OPEN = 1 så ska dörren öppnas, vara i öppet läge i 8 sekunder och därefter stängas. Öppningssignalen är en synkroniserad och enpulsad signal. Sensorsignalerna är asynkrona.

Ett exempel på korrekt funktion ses i följande tidsdiagram:

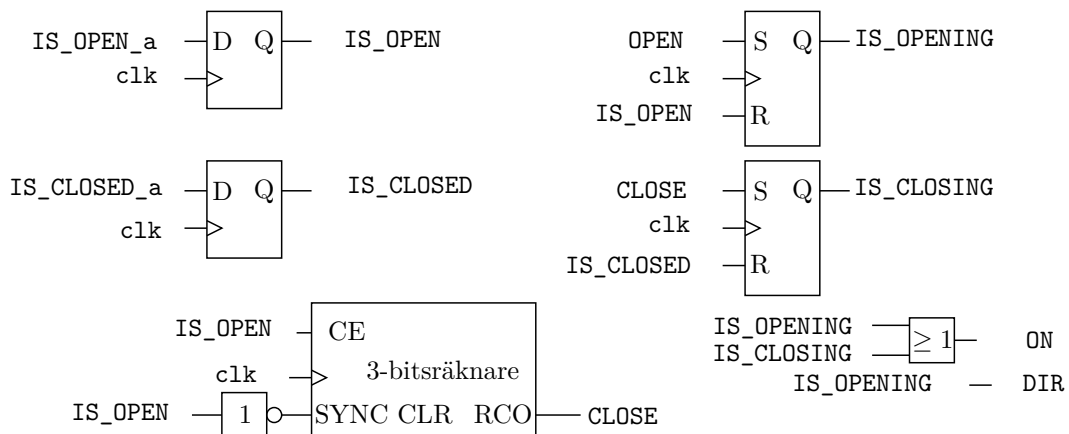


Observera att när dörren ska öppnas kan det dröja några klockpulser från det att motorn startas till det att sensorn för stängd dörr IS_CLOSED blir 0. Det samma gäller när dörren ska stängas och IS_OPEN blir 0. Det gör inget om motorn stängs av någon klockpuls efter att dörren har detekterats i ett ändläge. När motorn inte är på spelar det inte någon roll vilket värde DIR har. Detta är indikerat i grått. Ni får anta att OPEN bara sätts till ett när dörren är stängd.

Till er konstruktion har ni valfria vippor, grindar, räknare och en klocka på 1 Hz. (10 poäng)

Lösning. Det finns många olika lösningar. Här presenteras två varianter.

Lösningsförslag 1



Två synkroniseringsvippor för asynkrona insignaler. En SR-vippa håller ordning på om dörren håller

på att öppnas $IS_OPENING = 1$, en SR-vippa om dörren håller på att stängas $IS_CLOSING = 1$ och räknaren ser till att dörren är öppen specificerad tid.

Öppningsfasen startas genom att $OPEN$ sätt till 1 och avslutas med att $IS_OPEN = 1$ vilket därmed definierar insignalerna till SR-vippan med tillstånd $IS_OPENING$.

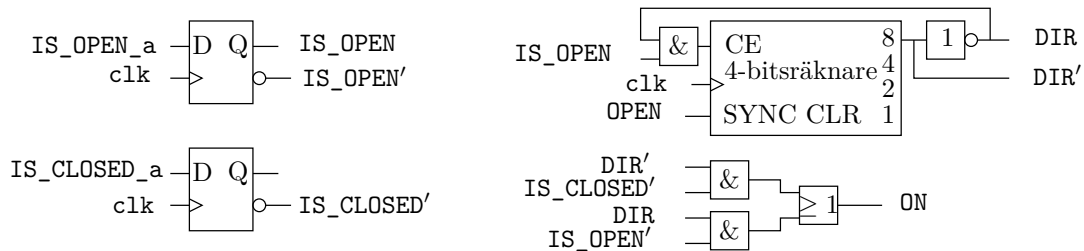
Innan dörren är öppen $IS_OPEN = 0$ så nollställs räknaren. När IS_OPEN blir 1 räknar räknaren upp till 7 då $CLOSE$ på RCO-utgången blir 1. Detta aktiverar dörrstängningen. Räknaren kommer räkna vidare till dess $IS_OPEN = 0$ men inte påverka övriga kretsen på något sätt.

När $CLOSE = 1$ aktiveras stängningsfasen genom att SR-vippan med tillstånd $IS_CLOSING$ sätt till 1. Stängningen avslutas då $IS_CLOSED = 1$.

Motorn ska köras $ON = 1$ under öppning och stänging. Detta sköts med OR-grinden.

Riktningen kan beräknas som $DIR = IS_OPENING$ som är 1 precis då hissdörren öppnas.

Lösningsförslag 2



I startläget när dörren är stängd står räknaren på 8 och $DIR = 0$. När $OPEN = 1$ nollställs räknaren och $DIR = 1$ vilket indikerar att dörren ska öppnas.

Den undre AND-grinden i AND-OR-kretsen håller motorn påslagen $ON = 1$ från och med att DIR blir 1 till dess att dörren är öppen, dvs då IS_OPEN blir 1.

När IS_OPEN blir 1 startar räknaren att räkna upp från 0 till 8 då DIR blir 0 och dörren ska stängas.

När DIR blir 0 kommer räknaren att stanna på 8 vilket gör att DIR förblir 0 tills räkaren nollställs vid nästa dörröppning.

Den övre AND-grinden i AND-OR nätet håller motorn påslagen till dess att dörren är stängd, dvs då IS_CLOSED blir 1.

Notera att AND-OR-kretsen realiserar en 2-1 multiplexer som väljer mellan IS_CLOSED' and IS_OPEN' då DIR är 0 respektive 1.