



KTH Informations- och kommunikationsteknik

Tentamen med lösningar i IE1204/5 Digital Design Torsdag 29/10 2015 9.00-13.00

Allmän information

Examinator: Ingo Sander.

Ansvarig lärare: William Sandqvist tel 08-7904487

Tentamensuppgifterna behöver inte återlämnas när du lämnar in din skrivning.

Hjälpmedel: Inga hjälpmedel är tillåtna!

Tentamen består av tre delar med sammanlagt 14 uppgifter, och totalt 30 poäng:

Del A1 (Analys) innehåller åtta korta uppgifter. Rätt besvarad uppgift ger en poäng. Felaktig besvarad ger 0 poäng. Det totala antalet poäng i del A1 är **10 poäng**. För **godkänt på del A1 krävs minst 6p, är det färre poäng rättar vi inte vidare.**

Del A2 (Konstruktionsmetodik) innehåller två metodikuppgifter om totalt **10 poäng**. För att bli **godkänd på tentamen** krävs **minst 11 poäng** från A1+A2, *är det färre poäng rättar vi inte vidare.*

Del B (Designproblem) innehåller två friare designuppgifter om totalt **10 poäng**.

OBS! I slutet av tentamenshäftet finns ett inlämningsblad för del A1, som ska avskiljas för att lämnas in tillsammans med lösningarna för del A2 och del B.

För ett godkänt betyg (**E**) krävs **minst 11 poäng på hela tentamen.**

Betyg ges enligt följande:

0 –	11 –	16 –	19 –	22 –	25
F	E	D	C	B	A

Resultatet beräknas meddelas före torsdagen den 19/11 2015.

Del A1: Analysuppgifter

Endast svar krävs på uppgifterna i del A1. Lämna svaren på inlämningsbladet för del A1 som du hittar på sista sidan av tentahäftet.

1. 1p/0p

En funktion $f(x, y, z)$ beskrivs med hjälp av uttrycket:

$$f(x, y, z) = \overline{x \cdot y \cdot z} + x \cdot \overline{y \cdot z} + (y + z)$$

Ange funktionens maxtermer, dvs ange funktionen som produkt av summor.

$$f(x, y, z) = \{PoS\} = ?$$

1. Lösningförslag

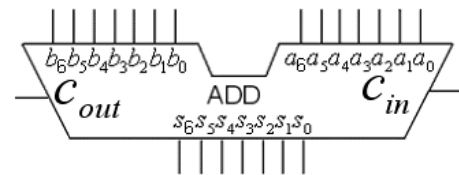
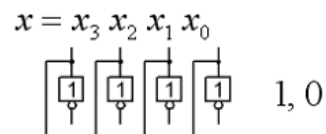
$$\begin{aligned} f(x, y, z) &= \overline{x \cdot y \cdot z} + x \cdot \overline{y \cdot z} + (y + z) = (\overline{x + y + z}) + x\overline{y}z + y\overline{z} = \overline{x + y + z} + \overline{y}z(x + 1) = \\ &= \overline{x + z} + y(1 + \overline{z}) = (\overline{x + y + z}) \quad \text{only one maxterm is needed} \end{aligned}$$

2. 1p/0p

Ett fyrabitars teckenlöst tal x ($x_3x_2x_1x_0$) ska multipliceras med konstanten 7.

Detta sker genom att talet x ansluts till en sju bitars adderare som konfigurerats för att utföra operationen $y = 7 \cdot x = (8 \cdot x - 1 \cdot x)$

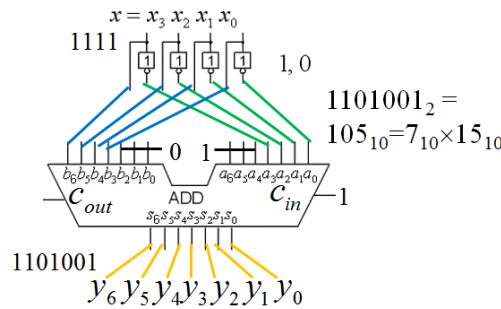
Rita hur adderaren ska konfigureras. Förutom de fyra bitarna i talet x så finns även konstanterna med värdet 0 eller 1 tillgängliga. En kopia av figuren finns även på svarsblanketten.



$$y_6 y_5 y_4 y_3 y_2 y_1 y_0$$

2. Lösningförslag

$$y = 7 \cdot x = (8 \cdot x - 1 \cdot x)$$



3. 1p/0p

Två binära 6 bitars tvåkomplement tal adderas. Vad blir resultatet uttryckt som ett decimaltal med tecken?

$$\begin{array}{r} 001011 \\ + 101110 \\ \hline \end{array} = \text{signed decimal } \pm??_{10}$$

3. Lösningförslag

$$001011 + 101110 = 111001 \quad -2^5 + 11001_2 = -32 + 25 = -7$$

$$\begin{array}{r} 111 \\ 001011 \\ + 101110 \\ \hline 111001 = -000111 = -7_{10} \end{array}$$

4. 1p/0p

Ett Karnaughdiagram för en funktion av fyra variabler $y = f(x_3, x_2, x_1, x_0)$ ges nedan.

Ange funktionen minimerad y_{\min} som en summa av produkter, på **SoP** form.

”-” i diagramet står för ”don't care”.

		y								
		x_1x_0	00	01	11	10				
x_3	x_2	0	0	1	0	3	0	2	1	
	0	-	0	1	0	1	0	1		
0	1	4	1	-	5	1	7	1	6	0
	1	12	1	1	13	1	15	0	14	0
1	1	8	1	9	0	10	11	0	10	-
	0	1	0	0	1	0	1	0	1	0

4. Lösningsförslag

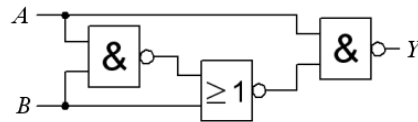
$$y = \bar{x}_2 \bar{x}_0 + x_2 \bar{x}_1 + x_3 x_2 x_0$$

		y								
		x_1x_0	00	01	11	10				
x_3	x_2	0	0	1	0	3	0	2	1	
	0	-	0	1	0	1	0	1		
0	1	4	1	-	5	1	7	1	6	0
	1	12	1	1	13	1	15	0	14	0
1	1	8	1	9	0	10	11	0	10	-
	0	1	0	0	1	0	1	0	1	0

5. 1p/0p

Figuren nedan visar ett grindnät med två NAND grindar och en NOR-grind. Förenkla funktionen

$Y = f(A, B)$ så långt som möjligt.



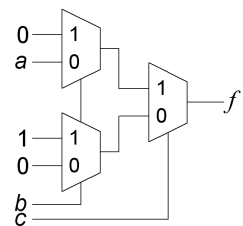
5. Lösningsförslag

$$Y = \overline{A \cdot A \cdot B} + A = \bar{A} + A \cdot B + A = 1$$

6. 1p/0p

En logisk funktion av tre variabler $c b a$ är realiserad med multiplexorer. Ange funktionen på minimerad **Pos** form (som produkt av summor).

$$f(c,b,a) = \{PoS\}_{\min} = ?$$



6. Lösningsförslag

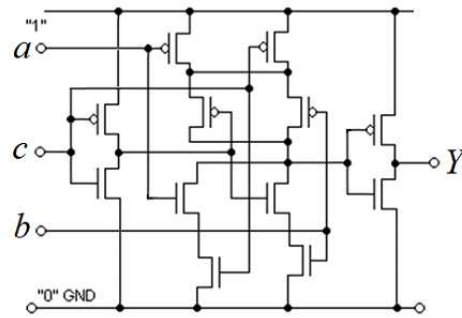
$$f(c,b,a) = \{SoP\} = 0 \cdot bc + a \cdot \bar{b}c + 1 \cdot b\bar{c} + 0 \cdot \bar{b}\bar{c} = \bar{a}bc + b\bar{c} = \bar{a}bc + (a + \bar{a})b\bar{c} = \bar{a}bc + a\bar{b}\bar{c} + \bar{a}b\bar{c}$$

		bc					
		00	01	11	10		
a	0	0	1	0	1	2	1
	1	4	5	1	7	6	1

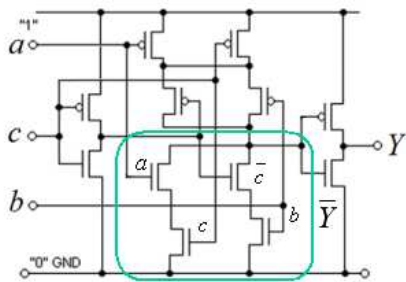
$$f(c,b,a) = \{PoS\}_{\min} = (\bar{b} + \bar{c})(b + c)(a + \bar{c}) \quad \text{or} \quad = (\bar{b} + \bar{c})(b + c)(a + b)$$

7. 1p/0p

Ange den logiska funktion som realiseras av CMOS kretsen i figuren nedan. $Y = f(a,b,c) = ?$



7. Lösningförslag

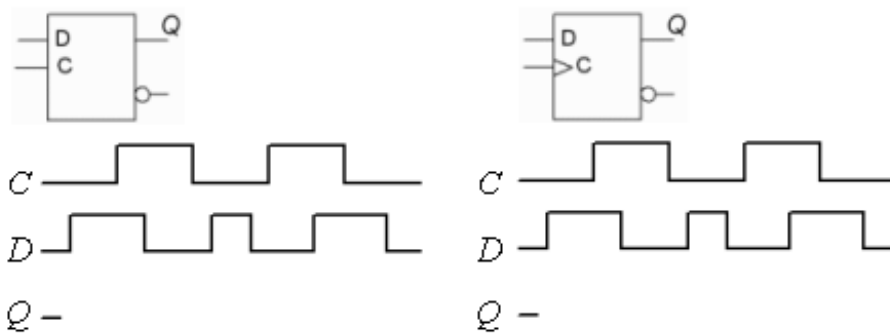


$$PDN: \bar{Y} = ac + b\bar{c}$$

$$\Rightarrow \bar{\bar{Y}} = Y = ac + b\bar{c}$$

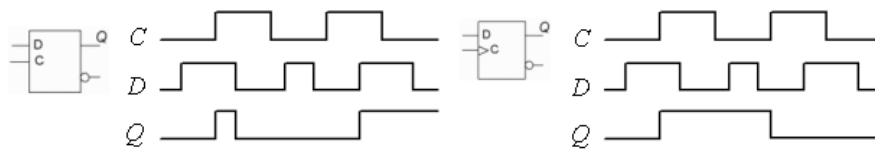
CMOS Multiplexor

8. 1p/0p

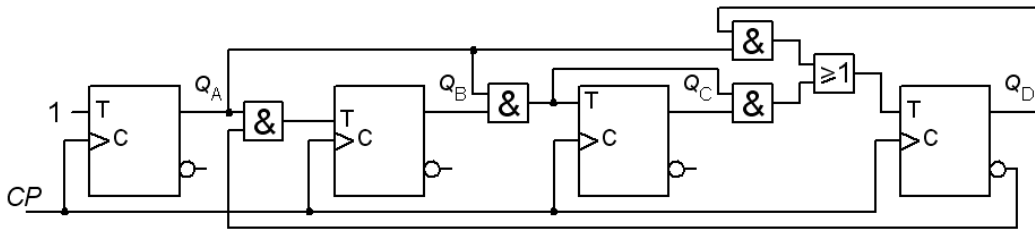


Komplettera tidsdiagrammen för D-latch och D-flipflop genom att rita signalen Q för båda fallen. Rita tydlig figur så att det framgår vad det är som orsakar förändringar i Q !

7. Lösningförslag

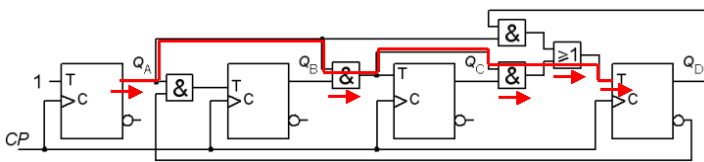


9. 1p/0p



Figuren visar en synkron dekadräknare ($Q_D Q_C Q_B Q_A$ 0...9). Markera (= rita i figuren på svarsblanketten) den kritiska signalvägen som avgör hur snabbt räknaren kan räkna (the "critical path"). Beräkna den minsta tid T [ns] mellan klockpulserna som fortfarande ger säker funktion.

Grindar: $t_{pdOR} = 4, t_{pdAND} = 5$ [ns] Vippor: $t_{su} = 3, t_h = 1, t_{pdQ} = 2$ [ns]



$$T = t_{pdQ} + t_{psAND} + t_{psAND} + t_{psOR} + t_{su} = 2 + 5 + 5 + 4 + 3 = 19 \text{ ns}$$

10. 1p/0p

Nedan är VHDL-koden för en 2:1 Multiplexor. Multiplexorns Karnaughdiagram visas till höger. Komplettera koden så att det blir en **Hazardfri** MUX. Kodraden finns också på svarsblanketten.

		Y			
		00	01	11	10
c	b a	0 0	1 1	3 1	2 0
	0	0	1	1	0
1	4	0	5	0	6
		0	0	1	1

```
-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;
```

```
-- this is the entity
entity MUX is
  port (
    a : in std_logic;
    b : in std_logic;
    c : in std_logic;
    Y : out std_logic);
end entity MUX;
```

```
-- this is the architecture
architecture gates of MUX is
begin
  Y <= (b AND c) OR (a AND NOT c) ;
end architecture gates;
```

10. Lösningsförslag

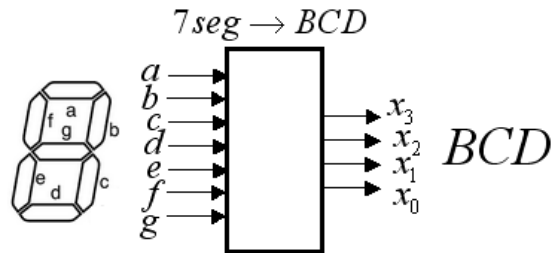
		Y			
		00	01	11	10
c	b a	0 0	1 1	3 1	2 0
	0	0	1	1	0
1	4	0	5	0	6
		0	0	1	1

```
Y <= (b AND c) OR (a AND NOT c) OR (a AND b);
```

Del A2: Konstruktionsmetodik

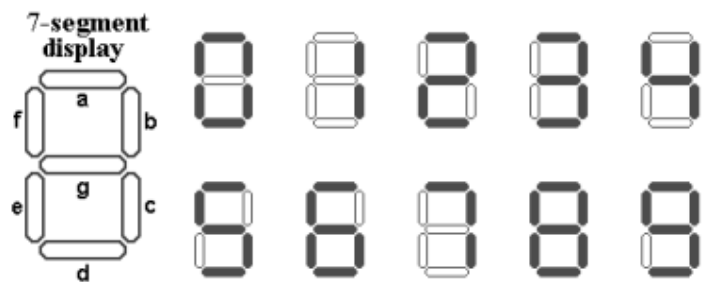
Observera! Del A2 rättas endast om Du är godkänd på del A1

11. 5p Ett äldre mätinstrument har en sjusegmentdisplay med sju glödlampor, men saknar ett uttag för anslutning till dator. Man vill tillverka ett kombinatoriskt nät som ansluts till glödlamporna och som sedan omvandlar 7-segmentkoden till den vanliga BCD-koden (normalt binärkodade siffror 0 till 9) som används av en mängd andra utrustningar.



a) (1p) Ställ upp **sanningstabellen** för de tio BCD-siffrorna. Lysande segment är "1". $(x_3x_2x_1x_0)_{BCD} = f(abcdefg)$

b) (1p) Inspektera **sanningstabellen**. Man kan då upptäcka att även om upp till två av segmenten **utesluts** som signaler förblir sambandet mellan segmentbild och BCD-siffra entydigt. Finn **ett/två** segment som man kan klara sig utan? Ställ upp den nya **sanningstabellen utan** ett eller två segment.



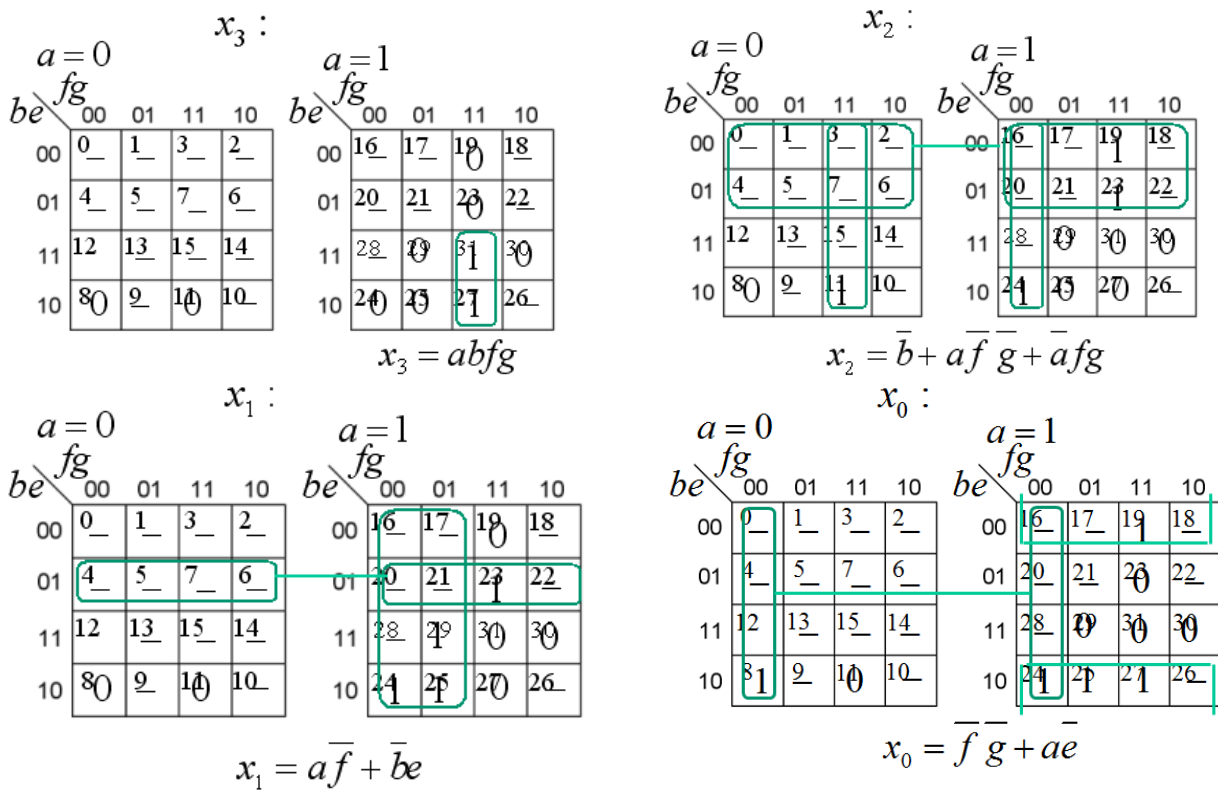
c) (2p) Ställ upp karnaughdiagram för de fyra bitarna i BCD-koden och ta fram de minimerade uttrycken för $x_3 x_2 x_1 x_0$ på SoP-form. Segmentkombinationer som aldrig uppträder **ska** utnyttjas som don't care. (Med ett eller två uteslutna segment i sanningstabellen blir antalet variabler hanterliga 5 eller 6).

d) (1p) Välj själv ut ett av uttrycken för $x_3 x_2 x_1 x_0$ och realisera det med enbart 2 ingångars **NAND**-grindar. (inga inverterade variabler finns tillgängliga)

11. Lösningförslag

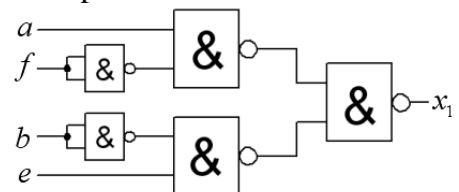
$abcdefg$	$x_3x_2x_1x_0$	7-segment display	$abefg$	$x_3x_2x_1x_0$
126 1111110	0 0000		30 11110	0 0000
48 0110000	1 0001		8 01000	1 0001
109 1101101	2 0010		29 11101	2 0010
121 1111001	3 0011		25 11001	3 0011
51 0110011	4 0100		11 01011	4 0100
91 1011011	5 0101		19 10011	5 0101
95 1011111	6 0110		23 10111	6 0110
112 1110000	7 0111		24 11000	7 0111
127 1111111	8 1000		31 11111	8 1000
123 1111011	9 1001		27 11011	9 1001

Segment *a* är nödvändigt för att kunna skilja mellan "1" och "7".
Segment *e* behövs för att skilja "8" från "9" och "5" från "6".
Däremot kan tex. segment *c* och *d* uteslutas utan att segmentkombinationerna blir mångtydiga.
Se figuren. Detta kan användas för att förenkla problemet ner till 5 variabler.

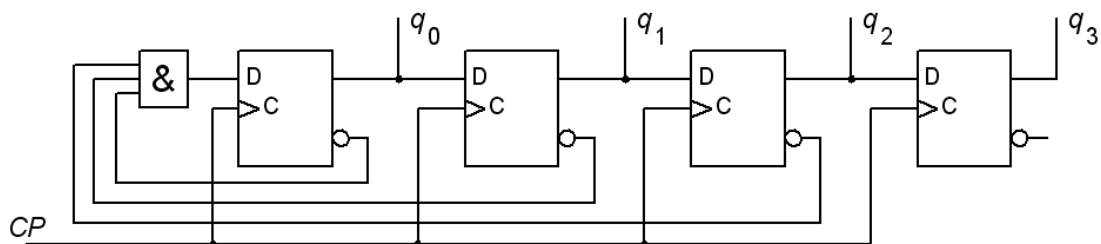


$x_3 = abfg$
 $x_2 = \bar{b} + a\bar{f}\bar{g} + afg$
 $x_1 = a\bar{f} + \bar{b}e$
 $x_0 = \bar{f}\bar{g} + a\bar{e}$

Exempel. Bit x_1 :

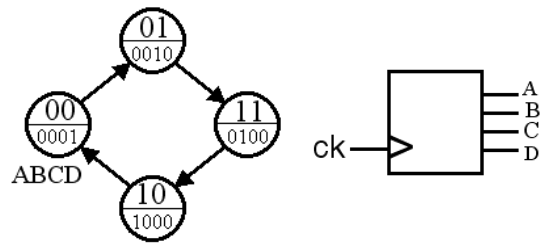


10. 5p Figuren visar ett sekvensnät, en "självkorrigerande ringräknare" som räknar "one hot"-sekvensen $q_3q_2q_1q_0$ 0001, 0010, 0100, 1000.



a) (2p) Analysera sekvensnätet i figuren och rita det **fullständiga tillståndsdigrammet** och den **fullständiga tillståndstabellen**. Om räknaren skulle starta i något *annat* tillstånd än något av de fyra önskade "one hot" tillstånden, hur många klockpulser krävs det i *värsta fall* innan räknaren "korrigerat" detta och hamnar i den rätta sekvensen?

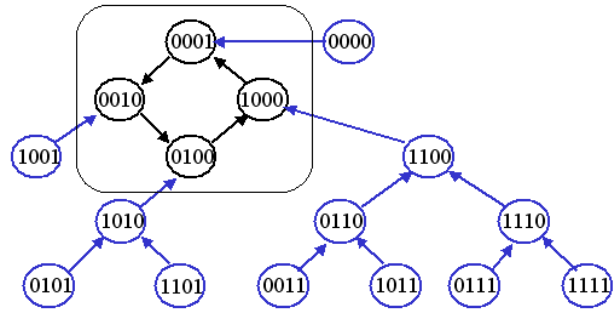
b) (3p) Man kan även få samma "one hot"-sekvens från en Moore-automat med fyra tillstånd, se tillståndsdia-grammet till höger. Konstruera detta sekvensnät med D-vippor och valfria grindar. Använd figurens tillståndskodning. Rita kretsschema.



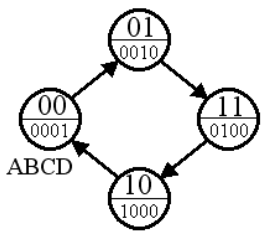
12. Lösningförslag

$q_3q_2q_1q_0$	$q_3^+q_2^+q_1^+q_0^+$	$q_3q_2q_1q_0$	$q_3^+q_2^+q_1^+q_0^+$
0000	0001	1000	0001
0001	0010	1001	0010
0010	0100	1010	0100
0011	0110	1011	0110
0100	1000	1100	1000
0101	1010	1101	1010
0110	1100	1110	1100
0111	1110	1111	1110

$$q_3^+q_2^+q_1^+q_0^+ = q_2q_1q_0d_0 \quad d_0 = \bar{q}_2 \cdot \bar{q}_1 \cdot \bar{q}_0$$



Efter högst **tre** klockpulser kommer man till "one hot" sekvensen!



Q_1Q_0	$Q_1^+Q_0^+$
00	01
01	11
11	10
10	00

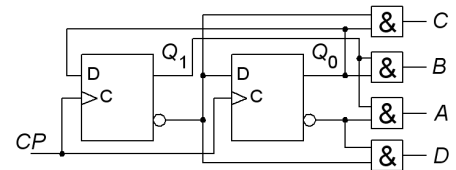
Inspektion av tabellen ger:

$$Q_1^+ = Q_0 \quad Q_0^+ = \bar{Q}_1$$

Avkodning:

$$A = Q_1\bar{Q}_0 \quad B = Q_1Q_0$$

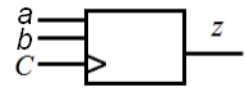
$$C = \bar{Q}_1Q_0 \quad D = \bar{Q}_1\bar{Q}_0$$



Del B. Designproblem

Observera! Del B rättas endast om Du har mer än 11p på del A1+A2.

13. 4p Sekvensdetektor. Olika tre i rad.



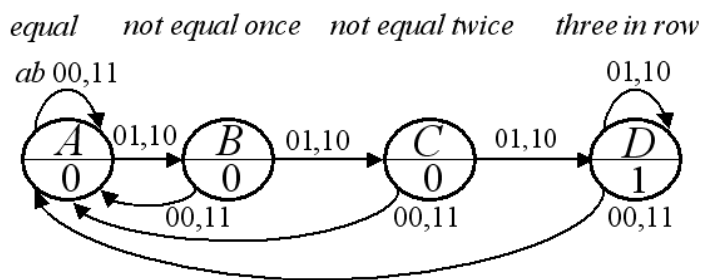
Du ska konstruera en synkron sekvenskrets, i form av en positivt flanktriggad Moore-automat med D-vippor. Insignalerna a och b är synkroniserad med klockpulserna C . Utsignalen z ska bli **1** när a och b varit **olika** i minst tre på varandra följande klockpulssintervall. För övriga sekvenser ska z vara lika med 0.

a) (2p) Ställ upp kretsens **tillståndstabell** och rita **tillståndsdiagram**.

b) (1p) Använd Graykod för att koda tillstånden och ställ upp den **kodade tillståndstabellen**. Tag fram de minimerade uttrycken **för nästa tillstånd** och för **utgångsvärdet**.

c) (1p) Rita nästa tillståndsavkodarens **grundnät**, det finns bara tillgång till AND, OR, och XOR grindar.

13. Lösningsförslag



q_1q_0	ab 00	01	11	10
A: 00	A: 00	B: 01	A: 00	B: 01
B: 01	A: 00	C: 11	A: 00	C: 11
C: 11	A: 00	D: 10	A: 00	D: 10
D: 10	A: 00	D: 00	A: 00	D: 00

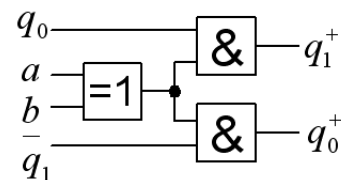
q_1q_0	ab 00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	0	1	0	1
10	0	0	0	0

$$q_1^+ = q_0(a\bar{b} + \bar{a}b) = q_0(a \oplus b)$$

q_1q_0	ab 00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

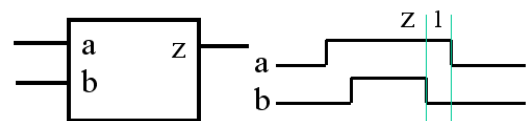
$$q_1^+ = q_1(a\bar{b} + \bar{a}b) = \bar{q}_1(a \oplus b)$$

$$z = q_1^+ \bar{q}_0^+$$



14. 6p Inside pulse detector.

Ett asynkront sekvensnät ”jämför” pulser som inkommer på två ingångar a och b . Pulsen på b ingången är alltid lite kortare än pulsen på a ingången, och det kommer högst en b -puls under intervallet a . b -pulsen kommer slumpvis i förhållande till a -pulsen. (Det förekommer inga *exakt* simultana händelser).



Sekvensnätet ska indikera det fallet när **b** har startat (blivit 1) *efter* det att **a** startat (blivit 1), och **b** har tagit slut (blivit 0) *innan* **a** har slutat (blivit 0). Utsignalen **z** ska då hållas =1 från **b**:s bakkant till **a**:s bakkant. **z** ska vara 0 för alla andra fall. Se figurens tidsdiagram som visar detta fall.

- Ställ först upp en korrekt **flödestabell** för sekvensnätet. Du behöver nu från början inte bry dig om att minimera antalet tillstånd. Alla positioner i tabellen som *inte* kan förekomma enligt beskrivningen ovan ska vara markerade som don't care.
- Förenkla tillståndsdiagrammet genom att slå ihop kompatibla tillstånd. (Olika lösningar är möjliga, det finns bla en lösning med fyra tillstånd).
- Gör en lämplig **tillståndstilldelning** med en **excitations-tabell** som ger nät som är **fria från kritisk kapplöpning**. (Olika lösningar är möjliga, en lösning med två tillståndsvariabler finns som utnyttjar icke stabila övergångstillstånd och okritisk kapplöpning). Du skall även ta fram de **hasardfria uttrycken** för nästa tillstånd samt ett **uttryck för utgångsvärdet**, och rita **grindnäten** med valfria grindar.

13. Lösningförslag – derive state chart step by step

• inside:
ab: 00, 10, 11, 10(1), 00

• partially before:
ab: 00, 01, 11, 10, 00

• partially after:
ab: 00, 10, 11, 01, 00

• before:
ab: 00, 01, 00, 10, 00

• after:
ab: 00, 10, 00, 01, 00

ab: 00 01 11 10 z

A	A	B	0
B	C	B	0
C	C	D	0
D	A	D	1
E			
F			
G			
H			

ab: 00 01 11 10 z

A	A	E	B	0
B	A	C	B	0
C	C	C	D	0
D	A	D	D	1
E	E	F		0
F	F	F	G	0
G	A	G		0
H	A	H		0

ab: 00 01 11 10 z

A	A	E	B	0
B	A	C	B	0
C	H	C	D	0
D	A	D	D	1
E	E	F		0
F	F	F	G	0
G	A	G		0
H	A	H		0

ab: 00 01 11 10 z

A	A	E	B	0
B	A	C	B	0
C	H	C	D	0
D	A	D	D	1
E	A	E	F	0
F	F	F	G	0
G	A	G		0
H	A	H		0

ab: 00 01 11 10 z

A	A	E	C	A	0
B	A	C	B		0
C	E	C	D		0
D	A		D		1
E	A	E	E	E	0

ab: 00 01 11 10 z

A	A	E	-	B	0
B	A	-	C	B	0
C	-	H	C	D	0
D	A	-	*	D	1
E	A	E	F	-	0
F	-	*	F	G	0
G	A	-	*	G	0
H	A	H	*	-	0

ab: 00 01 11 10 z

A	A	E	-	B	0
B	A	-	C	B	0
C	-	H	C	D	0
D	A	-	-	D	1
E	A	E	F	-	0
F	-	-	F	G	0
G	A	-	-	G	0
H	A	H	-	-	0

(EFGH)=E

ab: 00 01 11 10 z

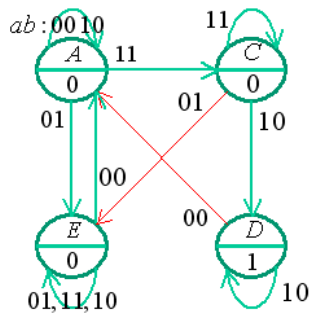
A	A	E	-	B	0
B	A	-	C	B	0
C	-	E	C	D	0
D	A	-	-	D	1
E	A	E	E	E	0

(AB)=A

ab: 00 01 11 10 z

A	A	E	C	A	0
B	A	-	C	B	0
C	-	E	C	D	0
D	A	-	-	D	1
E	A	E	E	E	0

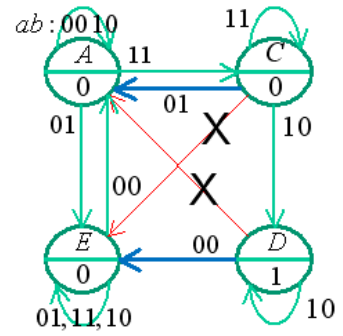
- can not occur, double change in input
* has not occurred in any of the possible sequences, then will not occur



ab: 00 01 11 10 z

A	A	E	C	A	0
C	-	E	C	D	0
D	A	E	-	D	1
E	A	E	E	E	0

Remove intersecting lines by inserting transition states to avoid hamming distance 2.



A:00 C:01 D:11 E:10

ab: 00 01 11 10 z

A	A	E	C	A	0
C	-	A	C	D	0
D	E	-	-	D	1
E	A	E	E	E	0

$z = y_1 y_0$

$Y_1 Y_0 = f(y_1 y_0, a, b)$

$y_1 y_0$ ab: 00 01 11 10 z

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

$y_1 y_0$ ab: 00 01 11 10

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

$Y_1 = y_0 \bar{b} + y_1 a + y_1 b + a \bar{b} y_0$

$y_1 b = \text{Hazard cover}$

$y_1 y_0$ ab: 00 01 11 10

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

$Y_0 = y_0 a + a \bar{b} y_1$

Lycka till!

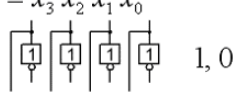
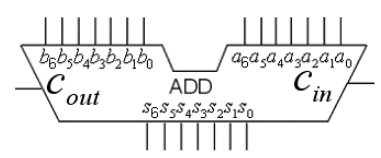
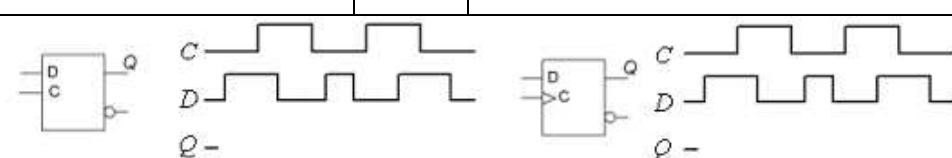
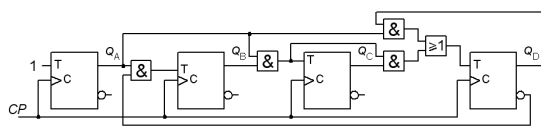
Inlämningsblad för del A Blad 1

(tas loss och lämnas in tillsammans med lösningarna för del A2 och del B)

Efternamn: _____ Förnamn: _____

Personnummer: _____ Blad: **1**

Skriv in dina svar för uppgifterna från del A1 (1 till 8)

Fråga	Svar
1	$f(x, y, z) = \{PoS\} = ?$
2	$y = 7 \cdot x = (8 \cdot x - 1 \cdot x)$ $x = x_3 x_2 x_1 x_0$   $y_6 y_5 y_4 y_3 y_2 y_1 y_0$
3	Signed decimal $\pm??_{10} =$
4	$f(x_3, x_2, x_1, x_0) = \{SoP\}_{\min} = ?$
5	$Y = f(A, B) = ?$
6	$f(c, b, a) = \{PoS\}_{\min} = ?$
7	$Y = f(a, b, c, d) = ?$
8	
9	 $T \text{ [ns]} =$
10	$Y \leq (b \text{ AND } c) \text{ OR } (a \text{ AND NOT } c)$ <div style="border: 1px solid black; width: 100px; height: 20px; display: inline-block; vertical-align: middle;"></div> ;

Nedanstående del fylls i av examinatorn!

Del A1 (10)	Del A2 (10)		Del B (10)		Totalt (30)	
Poäng	11	12	13	14	Summa	Betyg