



Skriftlig Tentamen

IE1204 Digital Design

2019-04-15, 08:00-12:00

Examiner/Examinator: Carl-Mikael Zetterling (IE1204)

Responsible teacher/Ansvarig lärare: Johnny Öberg

Swedish/Svenska:

Tentamenstexten ska lämnas in när lösningarna lämnas in. Inga tillåtna hjälpmedel utom linjal.

Examen består av fyra delar med 2 – 6 uppgifter och 10 poäng per del, och totalt 40 poäng.

Uppgifterna är inte ordnade efter svårighetsgrad.

Del 1: From Zero to One (Chapter 1)

Del 2: Combinational Logic Design (Chapter 2)

Del 3: Sequential Logic Design (Chapter 3)

Del 4: Digital Building Blocks (Chapter 5)

X = 1 om studenten har minst 2 poäng per del

Y = 1 om studenten har minst 20 poäng totalt

P = 1 om studenter får godkänt på tentamen

Fx = 1 om studenten kan godkännas efter en extra uppgift

X	Y	P	F _X
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

För godkänt krävs **minst 2 poäng per del OCH 20 poäng totalt.**

Fx om (19 poäng OCH 2 poäng per del) ELLER (20 poäng och 2 poäng på 3 delar).

Betygskalan för tentamen. Resultat meddelas inom tre veckor.

0-19	20-23	24-27	28-31	32-35	36-39	40
F	E	D	C	B	A	A+

Written Exam

IE1204 Digital Design

2019-04-15, 08:00-12:00

English:

The exam consists of four parts with 2 – 6 exercises and 10 points per part, for a total of 40 points. The exercises are not in order of difficulty.

- Part 1: From Zero to One (Chapter 1)
- Part 2: Combinational Logic Design (Chapter 2)
- Part 3: Sequential Logic Design (Chapter 3)
- Part 4: Digital Building Blocks (Chapter 5)

X = 1 if student has at least 2 points from each module

Y = 1 if student has at least 20 points in total

P = 1 if student passes exam

Fx = 1 if a student can pass after an extra task

X	Y	P	F _X
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

To pass the exam requires **at least 2 points from each module AND 20 points in total.**

F_x if (19 points AND 2 points per module) OR (20 points AND 2 points from 3 modules).

Grades are given as follows. The result will be announced within three weeks.

0-19	20-23	24-27	28-31	32-35	36-39	40
F	E	D	C	B	A	A+

Del 1/Part 1

1.1.Svenska: Antag 8-bitars 2-komplements binära tal. Alla beräkningar skall utföras mha binär aritmetik i två-komplements-form.

$$X=10110101$$

$$Y=11010000$$

- a) Beräkna $Z=X+Y$. Blir det overflow i ett 8-bitars resultat? (1p)
 b) Beräkna $W=X-Y$. Blir det overflow i ett 8-bitars resultat? (1p)

English: Assume 8-bits 2's complement binary numbers. All calculations should be done using binary arithmetic and two's complement algebra.

$$X=10110101$$

$$Y=11010000$$

- a) Calculate $Z=X+Y$. Is there an overflow in an 8-bit result? (1p)
 b) Calculate $W=X-Y$. Is there an overflow in an 8-bit result? (1p)

Suggested Solution:

- a) $C_{in} = C_{out}$ for last bit, No overflow.

	(1)	1	1					<u>Ctrl</u>		
	1	0	1	1	0	1	0	1	(-75)	
+	1	1	0	1	0	0	0	0	(-48)	
	(1)	1	0	0	0	0	1	0	1	(-123)

- b) $Y=11010000 \Rightarrow -Y=00110000$. $C_{in} = C_{out}$ for last bit, No overflow.

	(0)	1	1					<u>Ctrl</u>		
	1	0	1	1	0	1	0	1	(-75)	
+	0	0	1	1	0	0	0	0	(+48)	
	(0)	1	1	1	0	0	1	0	1	(-27)

1.2. Svenska: Antag 8-bitars 2-komplements binära tal. Alla beräkningar skall utföras mha binär aritmetik i två-komplements-form.

$$X=01111110$$

$$Y=00000110$$

- a) Beräkna $Z=X*Y$. Skriv 16-bitars resultatet på hexadecimal form. (1p)
 b) Beräkna $W=X/Y$. Skriv resultatet på hexadecimal form. (1p)

English: Assume 8-bits 2's complement binary numbers. All calculations should be done using binary arithmetic and two's complement algebra.

$$X=01111110$$

$$Y=00000110$$

- a) Calculate $Z=X*Y$. Write the 16-bit result on hexadecimal form. (1p)
 b) Calculate $W=X/Y$. Write the result in hexadecimal form. (1p)

Suggested Solution:

- a) To avoid multiplying the whole numbers together, we notice that $X=126$ or $(128-2)$ and $Y=6=(4+2)=(8-2)$. Thus, we can perform the multiplication by adding to shifted numbers together, $X*6=(X*4+X*2)$:

$$X*4=0111\ 1110\ 00$$

$$X*2=00111\ 1110\ 0$$

<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>				<u>Ctrl</u>
	0	1	1	1	1	1	1	0	0	0	(126*4=504)
+	0	0	1	1	1	1	1	1	0	0	(126*2=252)
0000 00	1	0	1	1	1	1	0	1	0	0	(126*6=756)

Answer: $X_{16}=0000\ 0010\ 1111\ 0100 = 02F4$

- b) We perform the division straight on:

<u>110</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>			<u>Ctrl</u>	
0	1	1	1	1	1	1	1	0				126/6=21=15 ₁₆
	-	1	1	0								
			1	1	1							
		-	1	1	0							
				1	1	0						
				-	1	1	0					
							0					

1.3. Svenska: Antag 8-bitars 2-komplements binära tal, med fyra fraktionsbitar. Alla beräkningar skall utföras mha binär aritmetik i två-komplements-form.

$$X=0110.0101$$

$$Y=0001.1110$$

- a) Beräkna $Z=X+Y$. Genererar beräkningen ett overflow? (1p)
 b) Beräkna $W=X*Y$. Svara på 16-bitars två-komplement form. (1p)

English: Assume 8-bits 2's complement binary numbers, with four fraction bits. All calculations should be done using binary arithmetic and two's complement algebra.

$$X=0110.0101$$

$$Y=0001.1110$$

- a) Calculate $Z=X+Y$. Does the calculation generate an overflow? (1p)
 b) Calculate $W=X*Y$. Answer in 16-bit 2's complement form. (1p)

Suggested Solution:

a) Cin is not = Cout for last bit, Overflow.

	①	1	1	1	1	0	1	0	1	Ctrl
	0	1	1	0	.	0	1	0	1	(6.3125)
+	0	0	0	1	.	1	1	1	0	(1.875)
①	1	0	0	0	.	0	0	1	1	(-7.8125)

b) $Y=0001.1110 \Rightarrow Y=(0010.0000-0000.0010)$. Cin = Cout for last bit, No overflow.

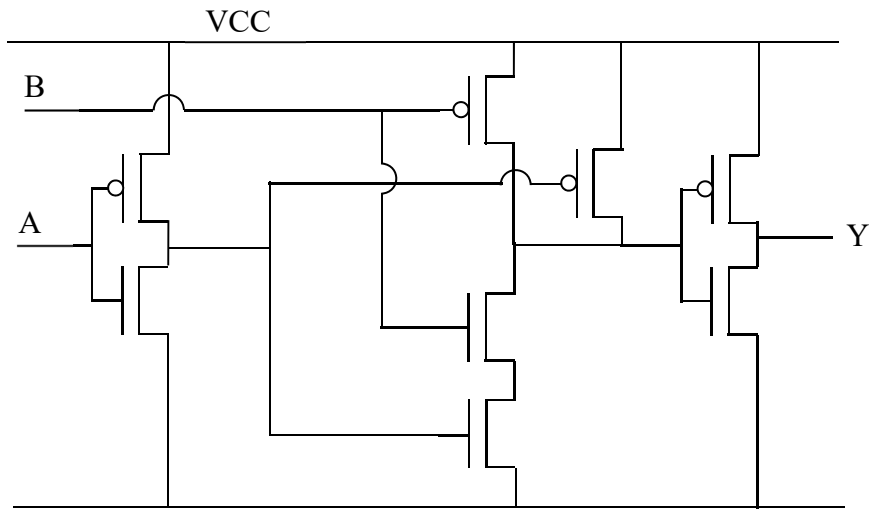
$$X*2=1100.1010, X/8=0000.1100 1010$$

	10	10	10	10	.	10	10	10	10	Ctrl
	1	1	0	0	.	1	0	1	0	12.625
-	0	0	0	0	.	1	1	0	0	0.7890625
0000	1	0	1	1	.	1	1	0	1	11.8359375

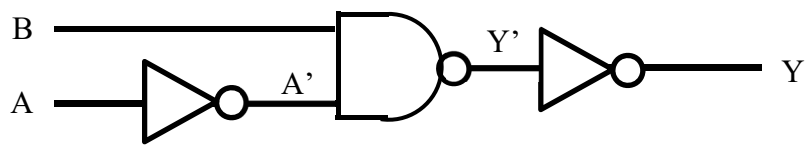
$$\text{Ctrl: } 6.3125*1.875=11.8359375$$

(Answer: 0B.D6)

1.4. Svenska: Härled sanningstabellen för följande CMOS-krets. (2 p)
English: Derive the Truth table for the following CMOS circuit. (2 p)



Suggested Solution:



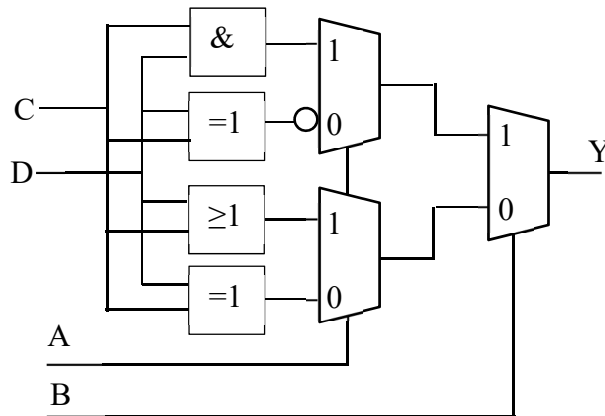
A	B	A'	Y'	Y
0	0	1	1	0
0	1	1	0	1
1	0	0	1	0
1	1	0	1	0

$$Y = (\overline{A}B)$$

1.5. Svenska: En student försöker att implementera en sanningstabell mha muxar och diverse grindar. Har studenten gjort rätt? Om inte, rätta till kretsen så att den överensstämmer med sanningstabellens Karnaugh-diagram. (2 p)

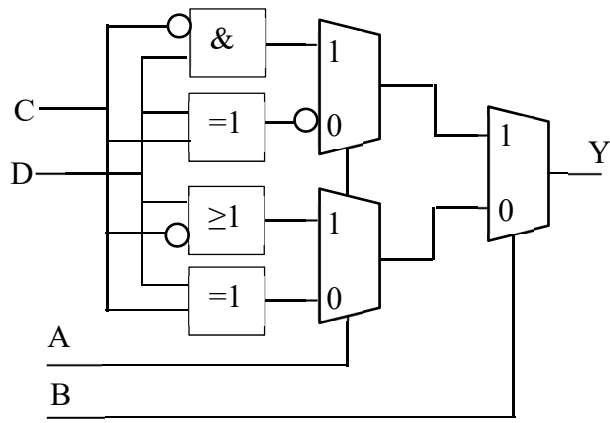
English: A student is trying to implement a truth-table using muxes and a couple of gates. Have the student done it correctly? If not, correct the circuit so that it corresponds to the Karnaugh-map of the truth-table. (2 p)

		CD			
		00	01	11	10
AB	00	0	1	0	1
	01	1	0	1	0
	11	0	1	0	0
	10	1	1	1	0



Suggested Solution:

		CD				
		00	01	11	10	
AB	00	0	1	0	1	$C \oplus D$
	01	1	0	1	0	$\overline{C \oplus D}$
	11	0	1	0	0	$\bar{C}D$
	10	1	1	1	0	$\overline{\bar{C}D} = \bar{C} + D$



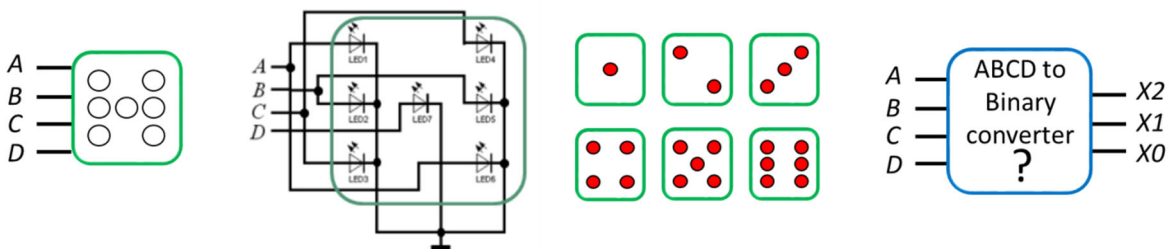
Del 2/Part 2

2.1 **Svenska:** En display för en elektronisk tärning består av sju LED-dioder som kontrolleras av fyra signaler A, B, C, och D enligt figuren nedan. Som synes kan var och en av signalerna A, B, och C slå på två LED:ar, medan D bara kan slå på en LED. Vi vill konstruera en ABCD till binärkod konverterare ($ABCD \rightarrow x_2x_1x_0$) för siffrorna 1,2,3,4,5, och 6. T.ex., när $ABCD = "1000"$ så visar tärningsdisplayen "2" och $x_2x_1x_0 = "010"$.

- Skriv den kompletta sanningstabellen för ABCD-till-binärkod-konverteraren. Antag att ingenting annat än de angivna tärningsslagen skall synas. Använd Don't Care för övriga möjliga mönsterkombinationer. (2p)
- Gör K-diagram för de tre binärkodsutgångarna (x_2 , x_1 och x_0) och extrahera de minimala Booleska ekvationerna i Summa-produkt-form. (2p)
- Konstruera och rita kretsen för kod-konverteraren mha av enbart NAND-grindar. (1p)

English: A display for an electronic dice consists of seven LEDs that are controlled by four signals A, B, C, and D according to the figure below. As shown in the figure, A, B, and C turns two LEDs "on" while D turns one LED "on". We want to design an **ABCD to binary code** converter ($ABCD \rightarrow x_2x_1x_0$) for digits 1, 2, 3, 4, 5, and 6. For example, when $ABCD = "1000"$ then the dice displays "2" and $x_2x_1x_0 = "010"$.

- Draw the complete Truth table for the ABCD to binary code converter. Assume that nothing more than the illustrated dice dot patterns appear on the electronic dice. Use Don't Care for other dot patterns. (2p)
- Make K-maps for the three bits of the binary-code (x_2 , x_1 , and x_0) and extract their minimized Boolean expressions on the Sum-of-Product form. (2p)
- Design and draw the circuit for the converter using only NAND gates. (1p)



Suggested Solution:

2.1 (a)

A	B	C	D	X2	X1	X0
0	0	0	0	-	-	-
0	0	0	1	0	0	1
0	0	1	0	-	-	-
0	0	1	1	0	1	1
0	1	0	0	-	-	-
0	1	0	1	-	-	-
0	1	1	0	-	-	-
0	1	1	1	-	-	-
1	0	0	0	0	1	0
1	0	0	1	-	-	-
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	-	-	-
1	1	0	1	-	-	-
1	1	1	0	1	1	0
1	1	1	1	-	-	-

(b)

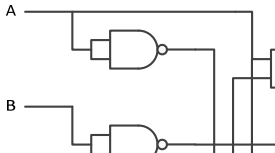
AB \ CD	00	01	11	10
00	-	-	-	0
01	0	-	-	-
11	0	-	-	1
10	-	-	1	1

AB \ CD	00	01	11	10
00	-	-	-	1
01	0	-	-	-
11	1	-	-	0
10	-	-	1	0

AB \ CD	00	01	11	10
00	-	-	-	0
01	1	-	-	-
11	1	-	-	1
10	-	-	0	0

OR

(c)



2.2 Svenska: För följande funktion:

$$f(a, b, c, d) = \bar{b}\bar{d} + b\bar{c} + b\bar{d}$$

- a) Ta fram ett minimerat uttryck på "Product-of-Sum"-form. (1 p)
 b) Konstruera och rita kretsen för det minimerade uttrycket med enbart NOR-grindar. (1 p)

English: Given the following function:

$$f(a, b, c, d) = \bar{b}\bar{d} + b\bar{c} + b\bar{d}$$

- a) Derive the minimized "Product-of-Sum" expression. (1p)
 b) Design and draw the circuit for the minimized expression using only NOR gates. (1p)

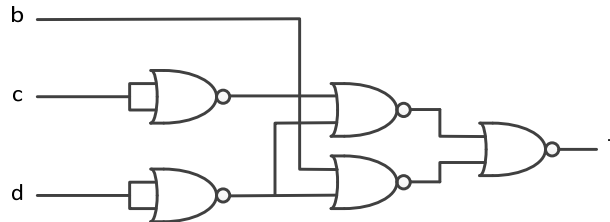
Suggested Solution:

2.2 (a)

		ab				f
		00	01	11	10	
cd	00	1	1	1	1	
	01	0	1	1	0	
	11	0	0	0	0	
	10	1	1	1	1	

$$f = (b + \bar{d})(\bar{c} + \bar{d})$$

(b)



2.3 Svenska: Följande krets har potentiella "glitchar" när en av ingångarna ändras.

(a) Gör en K-map för att förklara varför "glitchar" kan förekomma. (1 p)

Använd insignalsordningen som den visas i K-map nedan. Rita dina K-maps i din lösning.

(b) Modifiera det Boolska uttrycket och K-map för att undvika "glitches". (1 p)

(c) Rita den modifierade kretsen. (1 p)

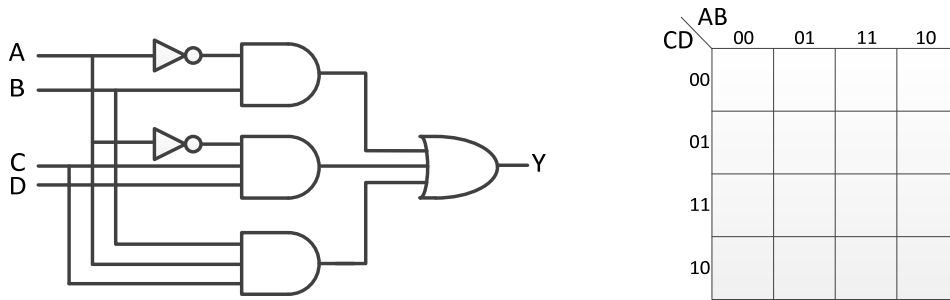
English: The following circuit has potential glitches when one of the inputs changes.

(a) Make a K-map to explain why glitches can occur. (1 p)

Use the input orders shown in the K-map. Redraw the K-map in the answer sheet.

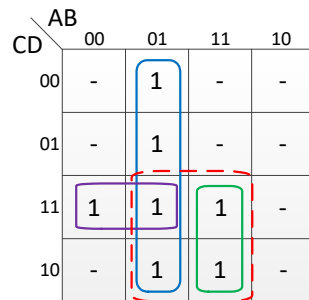
(b) Modify the Boolean expression and K-map to avoid glitches. (1 p)

(c) Draw the modified circuit. (1 p)

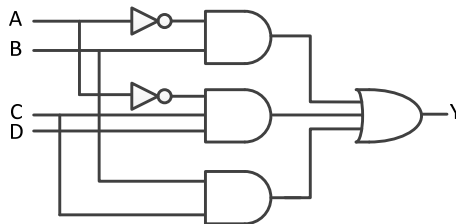


Suggested Solution:

2.3 (a)



(b)



Del 3/Part 3

3.1 **Svenska:** Designa en FSM med följande egenskaper:

Om ingången $En=1$ så skall den stega ett steg framåt.

Om ingången $En=0$ så skall den stega två steg bakåt.

Sekvensen repeteras enligt: 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, etc

- a) Rita tillståndsdigrammet. (1p)
- b) Skriv ner tillståndstabellen. Förutsätt att D-vippor används. (1,5p)
- c) Gör K-maps och ta fram uttrycken för nästa tillstånd. (1,5p)

English: Design an FSM with the following behavior:

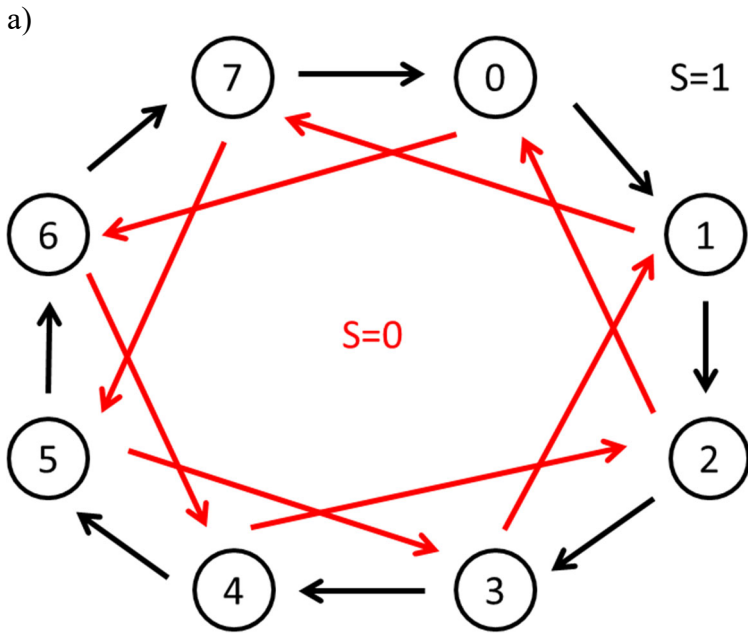
If the input $S=1$ the FSM should count one step forward.

If the input $S=0$ the FSM should count two steps backward.

The sequence repeats as such: 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, etc

- a) Draw the state diagram. (1p)
- b) Write down the state table. Assume that D-flipflops are used. (1,5p)
- c) Gör K-maps och ta fram uttrycken för nästa tillstånd (1,5p)

Suggested Solution:



b)

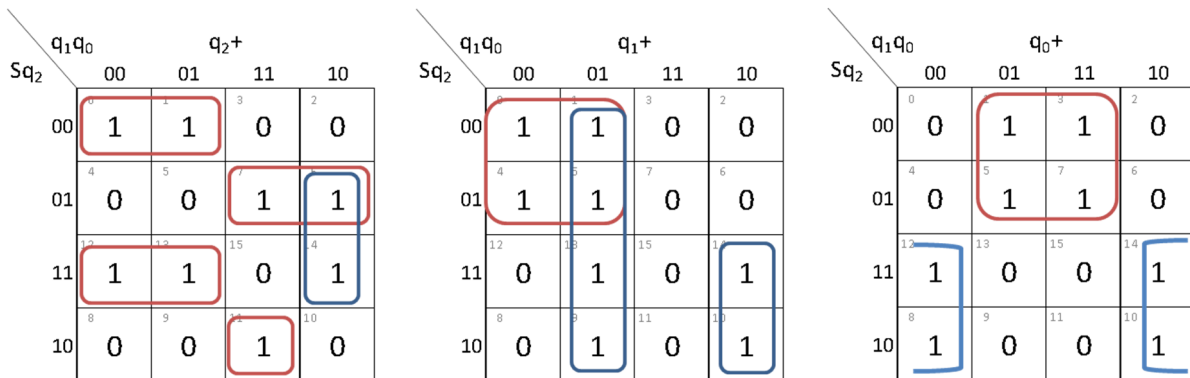
Previous State	Code	Next state S=1 (forward 1)	Next state S=0 (reverse 2)
0	0 0 0	0 0 1	1 1 0
1	0 0 1	0 1 0	1 1 1
2	0 1 0	0 1 1	0 0 0
3	0 1 1	1 0 0	0 0 1
4	1 0 0	1 0 1	0 1 0
5	1 0 1	1 1 0	0 1 1
6	1 1 0	1 1 1	1 0 0
7	1 1 1	0 0 0	1 0 1

c)

$$q_2 += \bar{S}\bar{q}_2q_1 + \bar{S}q_2q_1 + Sq_2q_1 + q_2q_1\bar{q}_0 + S\bar{q}_2q_1q_0$$

$$q_1 += \bar{S}\bar{q}_1 + \bar{q}_1q_0 + Sq_1\bar{q}_0$$

$$q_0 += \bar{S}q_0 + S\bar{q}_0$$

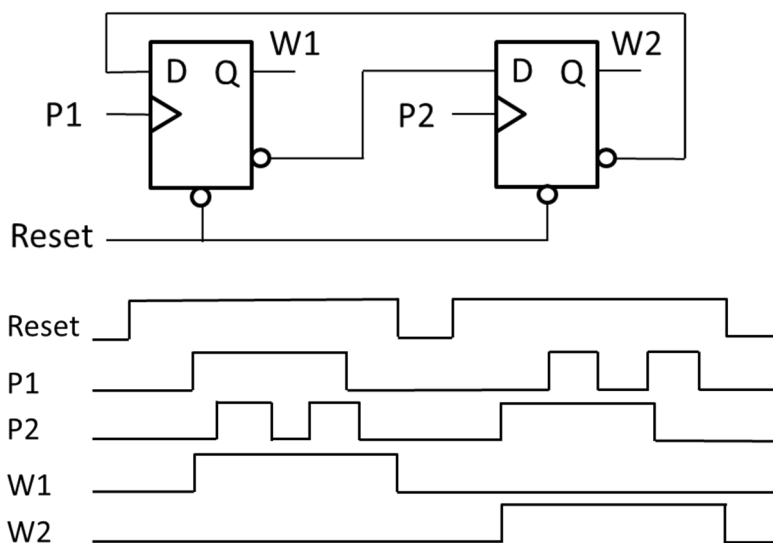


3.2. Svenska: Den synkrona kretsen nedan detekterar vilken av pulserna P1 och P2 som kommer först. Bara den första pulsen räknas. Efter Reset startar det om. Nu vill vi ha en Asynkron krets (utan vippor) som gör samma sak. **Ledning:** hantera Reset separat.

- a) Ta fram och rita tillståndsdigrammet och ställ upp flödesdiagrammet för funktionen. (1p)
- b) Gör en passande state assignment så att excitationstabellen är fri från kritiska signal-race. (0,5p)
- c) Ta fram hazard-fria uttryck för nästa tillstånd och output (0,5p)

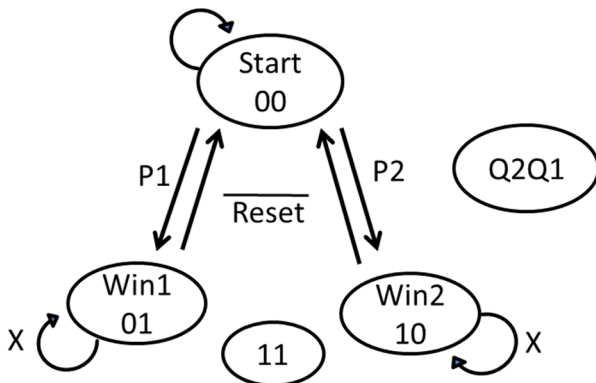
English: The Synchronous sequential circuit below detects which of the pulses P1 and P2 arrives first. Only the first pulse is counted. After Reset it starts over. Now we want an Asynchronous sequential circuit (without flip-flops) that does the same thing. **Hint:** treat Reset separately.

- a) Draw the state diagram and set up a proper flow table for the sequential circuit. (1p)
- b) Make a suitable state assignment with an excitation table that provides circuits that are free from critical race. (0.5p)
- c) Develop hazard free expressions for the next state and output. (0.5p)



Suggested Solution:

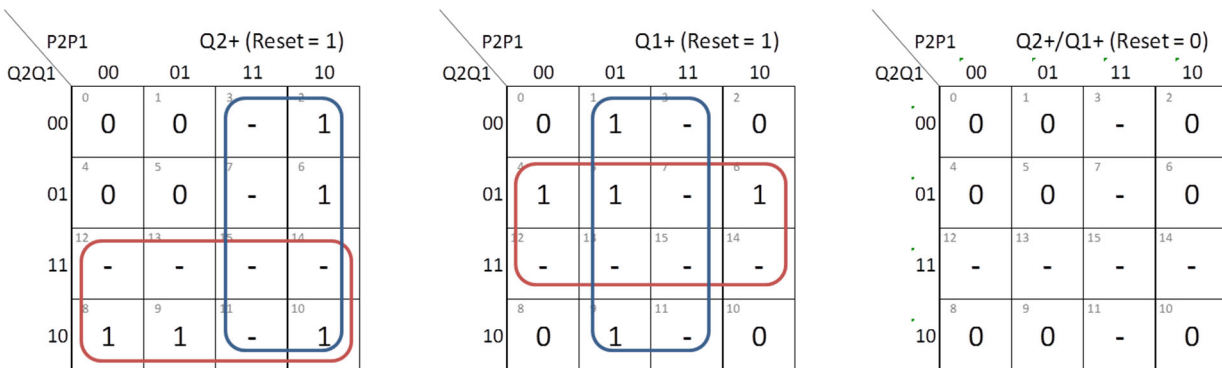
a) State diagram with state assignments



b) Excitation table for Reset = 1 (For Reset = 0 all entries are 00)

Present state Q2Q1	Next state P2P1=00	Next state P2P1=01	Next state P2P1=11	Next state P2P1=10
00	00	01	XX	10
01	01	01	XX	01
11	XX	XX	XX	XX
10	10	10	XX	10

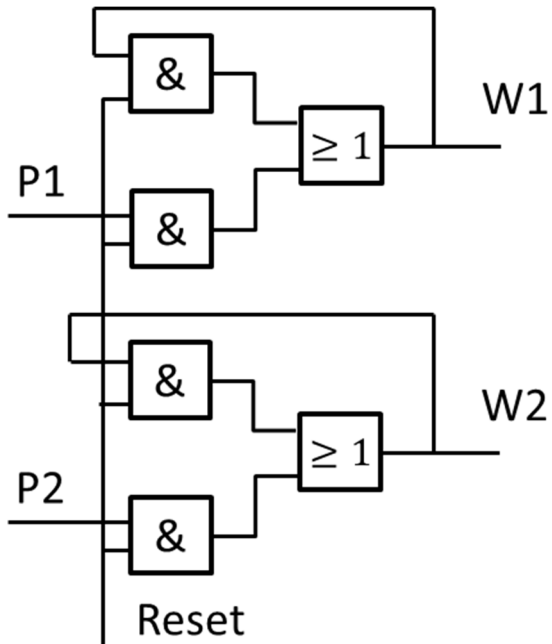
c) K-maps and Hazard free expressions



$$Q2+ = Q2 \cdot \text{Reset} + P2 \cdot \text{Reset}$$

$$Q1+ = Q1 \cdot \text{Reset} + P1 \cdot \text{Reset}$$

Schematic (not required)



3.3. **Svenska:** Givet den synkrona sekvenskretsen nedan. Antag att start-tillståndet är (00).

- a) Beräkna minimitiden T [ns] mellan klockpulser för säker operation. (1 p)
- b) Uppfylls villkoret för "hold time" i kretsen? Om inte, fixa felet. Förklara din lösning. (1 p)
- c) Ta fram Boolska uttryck för nästa tillstånd. (1 p)
- d) Rita tillståndsdigrammet. (1 p)

Grindar: $t_{pdAND} = 3$ ns, $t_{pdOR} = 3$ ns, $t_{cdAND} = 2$ ns, $t_{cdOR} = 2$ ns

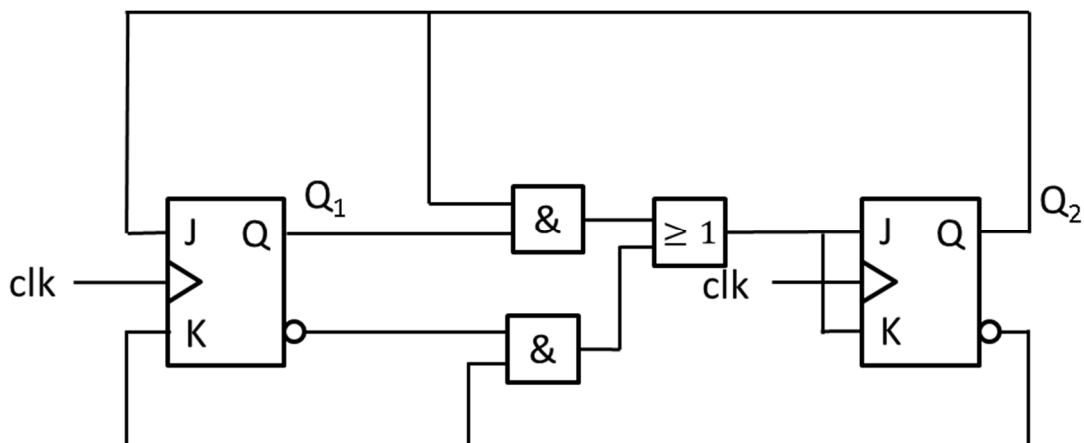
Vippor: $t_{setup} = 3$ ns, $t_{hold} = 2$ ns, $t_{pcq} = 2$ ns, $t_{ccq} = 1$ ns

English: The figure shows a synchronous sequential circuit. Assume the initialized state is (00).

- a) Calculate the minimum time T [ns] between the clock pulses that provides safe operation. (1 p)
- b) Is hold time constraint met in the circuit? If not, fix it. Explain your solution. (1 p)
- c) Derive the Boolean expressions for the next states. (1 p)
- d) Draw the state diagram. (1 p)

Gates: $t_{pdAND} = 3$ ns, $t_{pdOR} = 3$ ns, $t_{cdAND} = 2$ ns, $t_{cdOR} = 2$ ns

Flip-flops: $t_{setup} = 3$ ns, $t_{hold} = 2$ ns, $t_{pcq} = 2$ ns, $t_{ccq} = 1$ ns



J	K	Q
0	0	Q_0
0	1	0
1	0	1
1	1	$\overline{Q_0}$

Suggested Solution:

a) Critical Path = $T_{CP} = (Q_2 \rightarrow \text{AND} \rightarrow \text{OR} \rightarrow D_2) = t_{pcq} + t_{pd\text{AND}} + t_{pd\text{OR}} + t_{\text{setup}} = 2 + 3 + 3 + 3 = 11 \text{ ns}$

b) No. $Q_2 \rightarrow J_1$, $t_{ccq} = 1 \text{ ns} < t_{\text{hold}}$. Fix: Connect J_1 and K_1 via buffers (AND gate) \Rightarrow
 $t_{ccq} + t_{\text{buffer}} = 1 + 2 = 3 > t_{\text{hold}}$

c) Draw a state table:

Present		J_2 / K_2	Next	
Q_2	Q_1		Q_{2+}	Q_{1+}
0	0	1	1	0
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

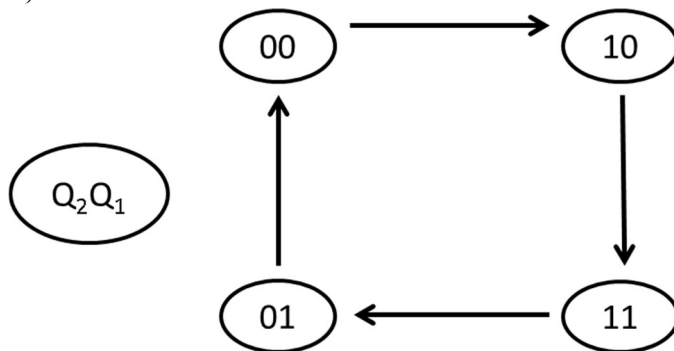
$Q_{1+} = Q_2$

$J_2 / K_2 = Q_1 \cdot Q_2 + \overline{Q_1} \cdot \overline{Q_2} = Q_1 \text{ XNOR } Q_2$

Q_2 toggles if $(Q_2 \text{ AND } Q_1) \text{ OR } (\text{NOT } Q_2 \text{ AND } \text{NOT } Q_1)$ (first and last row)

So $Q_{2+} = \text{NOT } Q_1$

d)



(2-bit Reverse Grey Code counter)

Del 4/Part 4

4.1. Svenska: Antag 32-bitars flyttal, IEEE standard 754 (sign bit, 8-bit exponent, 23-bit fraction)

$$X=0xC0F0\ 0000$$

$$Y=0x3FF0\ 0000$$

Utför additionen $Z=X+Y$. Svara på 32-bitars IEEE 754 flyttalsform. (1p)

English: Assume 32 bit floating point, IEEE standard 754 (sign bit, 8-bit exponent, 23-bit fraction)

$$X=0xC080\ 0000$$

$$Y=0x3F80\ 0000$$

Calculate the addition $Z=X+Y$. Answer in 32-bit IEEE 754 floating point form. (1p)

Suggested Solution:

$$X = 1100\ 0000\ 1111\ 0000\ [0]_{16} \Rightarrow \text{Sign}=1\ \text{Exp}=100\ 0000\ 1\ \text{Mantissa}=(1).111\ 000$$

$$Y = 0011\ 1111\ 1111\ 0000\ [0]_{16} \Rightarrow \text{Sign}=0\ \text{Exp}=011\ 1111\ 1\ \text{Mantissa}=(1).111\ 000$$

$$X=-1.875*2^{(129-127)}=-1.875*2^2 = -111100_2 \quad \text{Ctrl: } -7.5$$

$$Y=+1.875*2^{(127-127)}=+1.875 = +001111_2 \quad \text{Ctrl: } 1.875$$

$$\text{Ctrl: } Z=X+Y=-7.5+1.875=-5.625$$

			<u>10</u>	<u>10</u>	<u>10</u>	<u>10</u>	
	1	1	1	1	0	0	abs(X)
-	0	0	1	1	1	1	-abs(Y)
	1	0	1	1	0	1	abs(Z)

$$Z=\text{Sign}=1\ \text{Exp}=100\ 0000\ 1\ \text{Mantiss}=(1).011\ 0100 \quad \text{Ctrl: } -1.40625*2^{(129-127)}=-5.625$$

$$\text{Answer: } Z=0xC0B4\ 0000_{16}$$

Copy-paste-error English version:

$$X=1000\ 0000\ 1000\ 0000\ [0]_{16} = \text{Sign}=1\ \text{Exp}=100\ 0000\ 1\ \text{Mantissa}=(1).000$$

$$Y=0011\ 1111\ 1000\ 0000\ [0]_{16} = \text{Sign}=0\ \text{Exp}=011\ 1111\ 1\ \text{Mantissa}=(1).000$$

$$X=-1*2^{(129-127)}=-4$$

$$Y=1*2^{(127-127)}=1$$

$$Z=X+Y=-3 = -1.5*2 = \text{Sign}=1\ \text{Exp}=100\ 0000\ 0\ \text{Mantissa}=(1).100\ 0000$$

$$Z=1100\ 0000\ 0100\ 0000\ [0]_{16} = 0xC040\ 0000$$

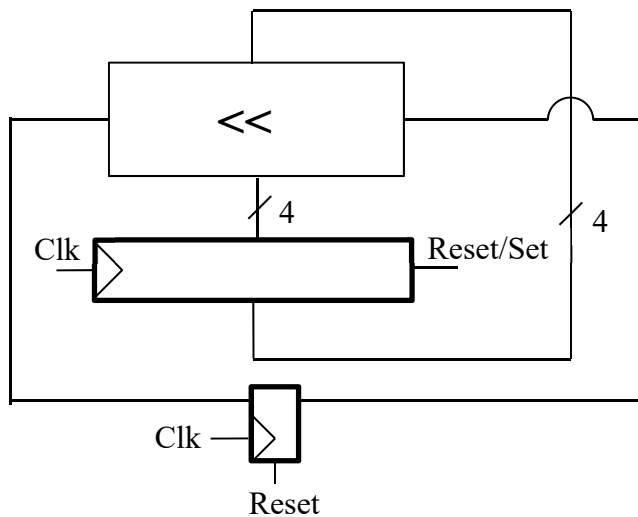
4.2. Svenska: En räknare består av en 4-bitars left-shifter, där shift ut har kopplats till shift in via en D-vippa, och fyra D-vippor. Vipporna klockas samtidigt. Räkaren startar med en reset/set som sätter alla vippor till ($q_3q_2q_1q_0 = 0001$) och carry-vippan till 0. Vad blir hela sekvensen?

(2 p)

English: A counter consists of a 4-bit left-shifter, where shift out has been connected to shift in via a D-flip-flop, and four D flip-flops. The flip-flops are clocked simultaneously. The counter starts with all the flip-flops at ($q_3q_2q_1q_0 = 0001$) and the carry-flip set to 0. What is the full sequence?

(2 p)

0001 -> ____ -> ____ -> etc...

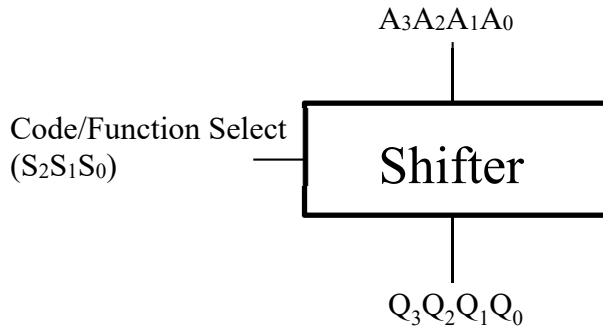


Answer:

(Carry DFF in parenthesis) Register 4 bits

(0) 0001 -> (0) 0010 -> (0) 0100 -> (0) 1000 -> (1) 0000 -> (0) 0001

4.3. Svenska: Du skall bygga en fyra bitars shifter. Shiftern skall ha följande Ingångar/funktioner:



Kod/Code (S ₂ S ₁ S ₀)	Funktion/Function	Ingång/Input	Resultat/Result (Q ₃ Q ₂ Q ₁ Q ₀)
000	Pass	A ₃ A ₂ A ₁ A ₀	A ₃ A ₂ A ₁ A ₀
001	<< 1		A ₂ A ₁ A ₀ "0"
010	<< 2		A ₁ A ₀ "00"
011	<< 3		A ₀ "000"
100	>>4		"0000"
101	>>3		"000" A ₃
110	>>2		"00" A ₃ A ₂
111	>>1		"0" A ₃ A ₂ A ₁

- Ställ upp Karnaugh-diagrammen för utgångarna (Q₃Q₂Q₁Q₀). K-diagrammet skall ha 3 ingångar (S₂S₁S₀). Använd A_i istället för 1:or. Härled de booleska ekvationerna för kretsen. (4p)
- Implementera kretsen mha en 3:8-dekoder och grindar. Rita kretsen, inklusive den interna strukturen på 3:8-dekodern. (1p)
- Identifiera och markera den längsta kritiska vägen (vilken ingång/utgång och hur många grindar). (1 p)
- Hur många transistorer består shiftern av? (1 p)

Antalet transistorer per grind är: NOT: 2, NAND₂/NOR₂: 4, AND₂/OR₂: 6, och lägg till 2 transistorer för varje extra ingång. Förklara eventuella antaganden.

English: You shall build a four bit shifter. The shifter should function according to the table above

- Draw the K-maps for the outputs (Q₃Q₂Q₁Q₀). The K-maps should have 3 inputs (S₂S₁S₀). Use A_i instead of 1:s. Derive the Boolean equations for the circuit. (4 p)
- Implement the circuit using a 3:8-decoder and gates. Draw the circuit, including the internal structure of the 3:8-decoder. (1p)
- Identify the longest critical path (which input/output and the number of gates). (1 p)
- How many transistors does the shifter consist of? (1 p)

The number of transistors per gate is: NOT - 2, NAND₂/NOR₂ - 4, AND₂/OR₂ - 6, and add 2 transistors per extra input. Explain your assumptions if needed.

Suggested Solution:

a) Larger Implicants must contain the same A_i (there are none in this example):

		S_1S_0			
		00	01	11	10
S_2	0	A3	A2	A0	A1
	1	0	0	0	0

$Q_3 = S_{000}A_3 + S_{001}A_2 + S_{010}A_1 + S_{011}A_0$

		S_1S_0			
		00	01	11	10
S_2	0	A2	A1	0	A0
	1	0	0	A3	0

$Q_2 = S_{000}A_2 + S_{001}A_1 + S_{010}A_0 + S_{111}A_3$

		S_1S_0			
		00	01	11	10
S_2	0	A1	A0	0	0
	1	0	0	A2	A3

$Q_1 = S_{000}A_1 + S_{001}A_0 + S_{110}A_3 + S_{111}A_2$

		S_1S_0			
		00	01	11	10
S_2	0	A0	0	0	0
	1	0	A3	A1	A2

$Q_0 = S_{000}A_0 + S_{101}A_3 + S_{110}A_2 + S_{111}A_3$

c) See next page

d) Critical path goes through one inverter on $S_i \rightarrow Q_i$ for instance $S_0 \rightarrow Q_3$.
4 Gates, Inv \rightarrow AND3 \rightarrow AND2 \rightarrow OR4

e)

Gate Type	Transform	Transistor Count	Total
3 Inverter (2 tr)		3*2	6
8 And3 = NAND3+Inv (6+2)		8*8	64
16 AND2 \Rightarrow 16 NAND2 (4)	DeMorgan	16*4	64
4 OR4 \Rightarrow 4 NAND4 (8)		4*8	32
Sum			166

