# Skriftlig Tentamen
# IE1204 Digital Design
# 2019-01-14, 14:00-18:00

---

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 | Fx |
|---|---|---|---|
| 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
# 2019-01-14, 14:00-18:00

---

**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 | Fx |
|---|---|---|---|
| 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**.
Fx 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

*X=10110001*

*Y=01010001*

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

*X=10110001*

*Y=01010001*

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)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | (1) | 1 | 1 | | | | 1 | | |
| | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | (-79) |
| + | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | (+81) |
| (1) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | (+2) |

No overflow. Carry in and carry out for the last bit (sign bit) is the same.

b) –Y=10101111

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | (0) | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | (-79) |
| + | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | (-81) |
| (1) | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | (+96) |

Overflow. Carry in and carry out for the last bit (sign bit) is different. Also, sign bits in and sign bit out are not consistent (two negative numbers added cannot become positive).

**1.2. Svenska:** Antag 8-bitars 2-komplements binära tal

*X=10000000*

*Y=00010000*

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

b) Beräkna W=X/Y. Svara på 4-bitars tvåkomplements binär form. (1p)

**English:** Assume 8-bits 2's complement binary numbers

*X=10000000*

*Y=00010000*

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

b) Calculate W=X/Y. Write the result in 4-bit 2's complement form. (1p)

**Suggested Solution**

a)

X=-128. If you do the 2's complement, you get +128, which is X=10000000, because it will overflow in 8 bit precision. This makes this task a little bit tricky. However, if you sign extend X into 16 bits, you start with X=1111 1111 1000 000.

Y=00010000 (16), which is a power of two, so multiplying is easy, just shifting X four steps to the left:

X*Y=1111 1000 0000 0000=H'F800. Decimal: $-2^7*2^4=-2^{11}=-2048$

b)

$-2^7/2^4=-2^3=-8$=B'1000 or
B'1000 000 shifted four steps to the right = 1111 1000 or in 4-bit 2's complementary form B'1000

**1.3. Svenska:** Antag 8-bitars 2-komplements binära tal, med fyra fraktionsbitar

*X=1111.0001*

*Y=0011.1111*

a) Beräkna Z=X+Y. (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

*X=1111.0001*

*Y=0011.1111*

a) Calculate Z=X+Y. (1p)

b) Calculate W=X*Y. Answer in 16-bit 2's complement form. (1p)

**Suggested Solution:**

a)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | | |
| | 1 | 1 | 1 | 1 | . | 0 | 0 | 0 | 1 | (-15/16) |
| + | 0 | 0 | 1 | 1 | . | 1 | 1 | 1 | 1 | (+63/16) |
| ① | 0 | 0 | 1 | 1 | . | 0 | 0 | 0 | 0 | (+48/16=3) |

b)

X in 2's complement is 0000.1111
Y in 2's complement is 1100.0001
The numbers contain a lot of 1's so multiplying them directly in binary form is not so wise. It will require a lot of effort. However, here we can use the fact that a lot of ones can be written as a power of two minus one (or minus the LSB power in this case).
Thus, X can be written as $-(1-2^{-4})$ and Y can be written as $(4-2^{-4}=2^2-2^{-4})$. Since it is only shifting, we can keep the sign bits. So, the solution is X shifted two steps to the left, and then subtract itself shifted four steps to the right OR Y as it is, subtracted with itself shifted four steps to the right, and then 2's complemented because of the negative sign.

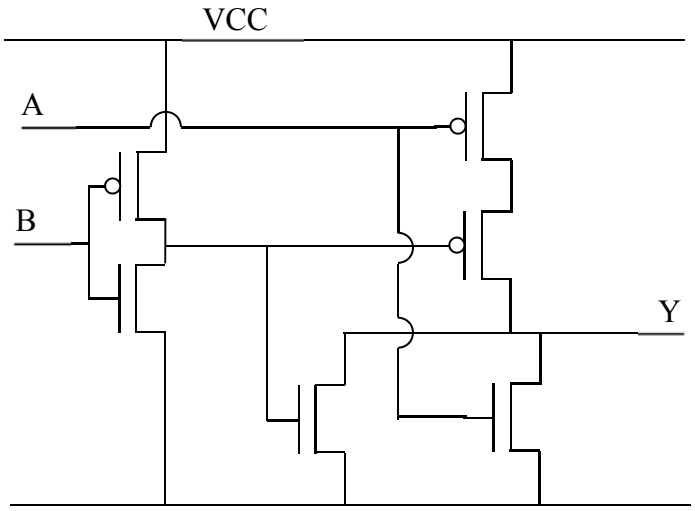
1) $X*(2^2-2^{-4})$ => 1111 1100.0100 0000 – 1111 1111.1111 0001 =
1111 1100.0100 0000 + 0000 0000.0000 1111 = 1111 1100.0100 1111

2) $Y*(1-2^{-4})$ => 0000 0011.1111 0000 – 0000 0000.0011 1111=
0000 0011.1111 0000 + 1111 1111.1100 0001

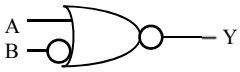
| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | . | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | +63/16 |
| + | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | . | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | (-63/256) |
| ① | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | . | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | (+48/16=3) |

However, since X is negative, in 2's complement, W =X*Y=1111 1100.0100 1111

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

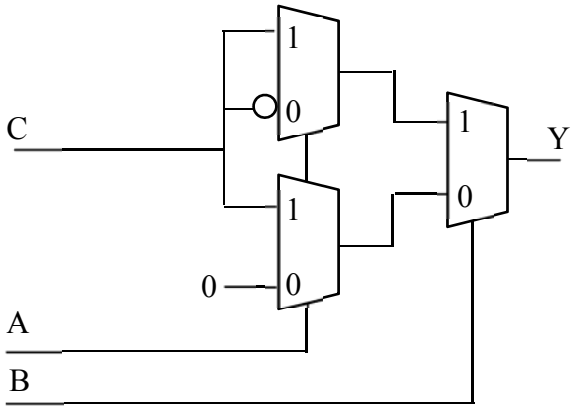



**Suggested solution**




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**1.5. Svenska:** Rita sanningstabellen för följande Multiplexer-krets. (1 p)
**English:** Draw the Truth table for the following Multiplexor circuit. (1 p)

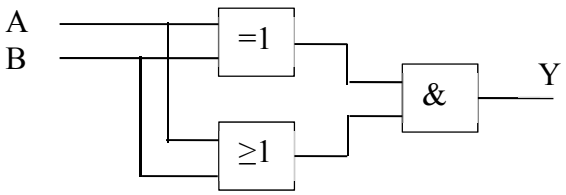



**Suggested Solution**

| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |

| A | B | C | Y |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

**1.6. Svenska:** Rita sanningstabellen för följande circuit. (1 p)
**English:** Draw the Truth table for the following circuit. (1 p)




**Suggested Solution**

| A | B | A⊕B | A+B | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |

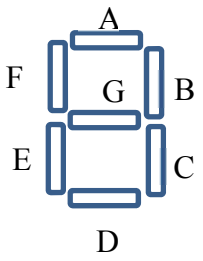
## Del 2/Part 2

2.1**. Svenska:** Konstruera en avkodare som kodar talen 0-9 (BCD-kod - $b_3b_2b_1b_0$) till rätt signaler som kontrollerar lysdioderna i en 7-segments display. Notera att för att en lysdiod ska vara tänd måste motsvarande utgång ha en logisk 0:a, se exemplet i figuren nedan.

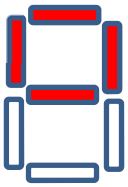
a) Rita hela 16 x 7 sanningstabellen för BCD till 7-segmentsavkodaren. Oanvända ingångskombinationer skall sättas till Don't Care ('-'). (3,5 p)
b) Gör K-maps för segmenten A och B och ta fram minimala Boolska uttryck. (2 p)
Använd insignalsordningen som den visas i K-map nedan. Rita dina K-maps i din lösning.
c) Konstruera och rita kretsen för segment C med enbart NAND-grindar. (1 p)
d) Konstruera och rita kretsen för segment D med en 4 till 1 multiplexer och valfria grindar. (1 p)

**English:** Design a decoder that maps the numbers 0-9 (BCD-code – $b_3b_2b_1b_0$) to appropriate signals which controls the LEDs of a seven-segment display. Note that in order to turn the LED on, the corresponding output should be a logic "0", see example below.

a) Draw the complete 16 x 7 Truth table for the BCD to seven-segment decoder. Unused input combinations should be set to Don't Care ('-'). (3,5 p)
b) Make K-maps for segments A and B and extract their minimized Boolean expressions. (2 p)
Use the input orders shown in the K-map below. Redraw the K-map in the answer sheet.
c) Design and draw the circuit for segment C using only NAND gates. (1 p)
d) Design and draw the circuit for segment D using a 4 to 1 multiplexer and any gates. (1 p)




Exempel/
Example



=

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

| $b_3b_2$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

**Suggested Solution**

a)

| $B_3$ | $B_2$ | $B_1$ | $B_0$ | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1/0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| $B_3$ | $B_2$ | $B_1$ | $B_0$ | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1/0 | 1 | 0 | 0 |
| 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)

| $B_3B_2$ \ $B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | - | - | - | - |
| 10 | 0 | 0 | - | - |

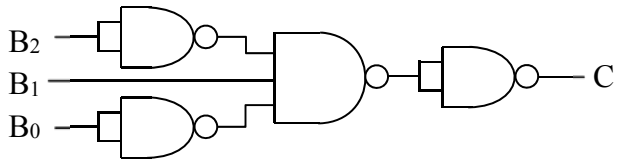
$$A = B_2\bar{B}_1\bar{B}_0 + \bar{B}_3\bar{B}_2\bar{B}_1B_0$$

| $B_3B_2$ \ $B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | - | - | - | - |
| 10 | 0 | 0 | - | - |

$$B = B_2\bar{B}_1B_0 + B_2B_1\bar{B}_0$$

c)

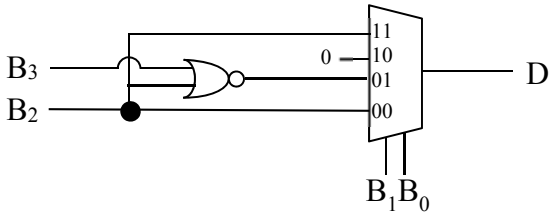
| $B_3B_2$ \ $B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | - | - | - | - |
| 10 | 0 | 0 | - | - |




$$C = \bar{B}_2B_1\bar{B}_0$$

d)

| $B_3B_2$ \ $B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 |
| 01 | 1 | 0 | 1 | 0 |
| 11 | - | - | - | - |
| 10 | 0 | 0 | - | - |



2.2 **Svenska:** För följande funktion:

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

a) Rita Karnaugh-diagrammet för funktionen (0,5p)
b) Ta fram ett minimerat uttryck på ”Product-of-Sum”-form. (1 p)
c) Konstruera och rita kretsen för det minimerade uttrycket med enbart NOR-grindar. (1 p)

**English:** Given the following function:

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

a) Draw the Karnaugh-map for the function (0,5p)
b) Derive the minimized Product-of-Sum expression. (1 p)
c) Design and draw the circuit for the minimized expression using only NOR gates. (1 p)

**Suggested Solution**

a)

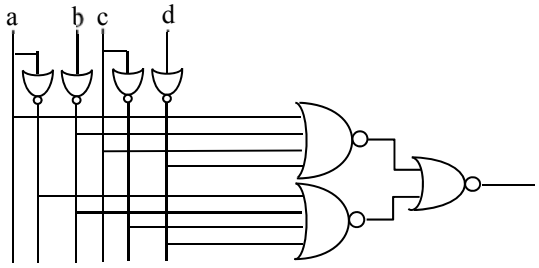
| ab \ cd | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | 1 |
| 01 | 1 | 0 | 1 | 1 |
| 11 | 1 | 1 | 0 | 1 |
| 10 | 1 | 1 | 1 | 1 |

b)

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

c)

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



## Del 3/Part 3

3.1 **Svenska:** Designa en FSM med följande egenskaper:
Om ingången En=1 så skall den stega 3-bitars Gray-kod i ordning, dvs,
000,001,011,010,110,111,101,100. Sedan skall sekvensen upprepas.
Om insignalen En=0 så skall den stanna i nuvarande tillstånd.

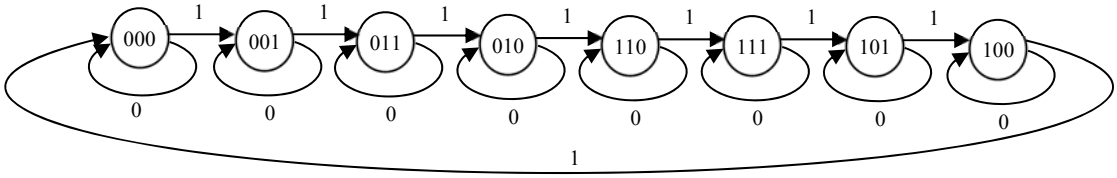
a) Rita tillståndsdiagrammet. (1p)
b) Skriv ner tillståndstabellen. Förutsätt att D-vippor används. (1,5p)
c) Gör K-maps och ta fram uttrycken för nästa tillstånd. (1,5p)

**English:** Design an FSM with the following behaviour:
If the input En=1 the FSM shall go through the 3-bit Gray-code sequence in order, i.e.,
000,001,011,010,110,111,101,100. The sequence should then be repeated.
If the input En=0, the FSM shall stay in its present state.

a) Draw the FSM. (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**

a)

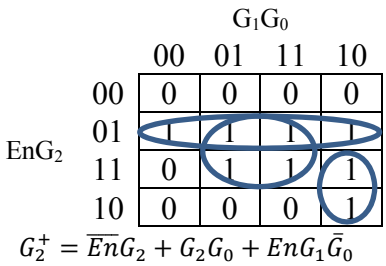



b)

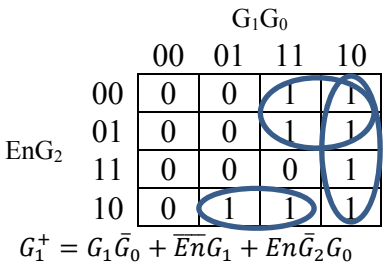
| En | $G_2$ | $G_1$ | $G_0$ | $G_2^+$ | $G_1^+$ | $G_0^+$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |

| En | $G_2$ | $G_1$ | $G_0$ | $G_2^+$ | $G_1^+$ | $G_0^+$ |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 |

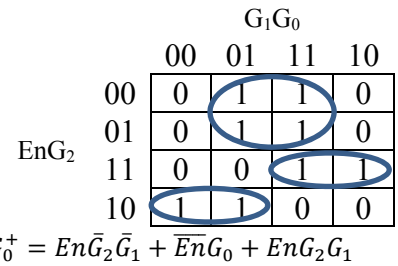
c)




$G_2^+ = \overline{En}G_2 + G_2G_0 + EnG_1\bar{G}_0$




$G_1^+ = G_1\bar{G}_0 + \overline{En}G_1 + En\bar{G}_2G_0$




$G_0^+ = En\bar{G}_2\bar{G}_1 + \overline{En}G_0 + EnG_2G_1$

**3.2. Svenska:** Konstruera en Asynkron sekvenskrets som räknar pulser på ingången *X*. Om antalet pulser på ingången *X* är udda så skall utgången Y vara 1. Om antalet pulser på ingången *X* är jämnt så skall utgången *Y* vara 0. (**Ledning**. Det räcker med fyra tillstånd).

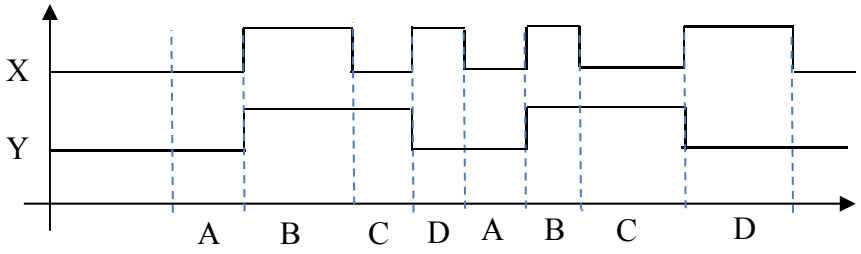
**a)** Ta fram och rita tillståndsdiagrammet och ställ upp flödesdiagrammet för funktionen. (1p)
b) Gör en passande state assignment så att excitationstabellen är fri från kritiska signal-race. (0,5p)
c) Ta fram hazard-fria uttryck för nästa tillstånd och output. (0,5p)

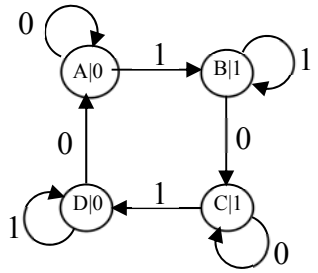
**English**: Design an Asynchronous sequential circuit that counts pulses on the input *X*. If the number of pulses on the input *X* is odd, the output *Y* should be 1. If the number of pulses on the input *X* is even, the output *Y* should be 0. (**Hint**. Four states are enough).

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)



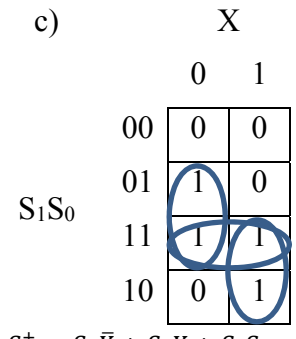




| State | X | | Y |
|---|---|---|---|
| | 0 | 1 | |
| A | A | B | 0 |
| B | C | B | 1 |
| C | C | D | 1 |
| D | A | D | 0 |

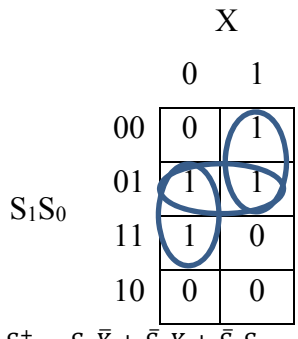
b)

A=00, B=01, C=11, D=10 is free from races (only bit changes per state bit=requirement for asynchronous circuits)

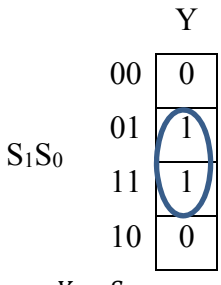
c)




$S_1^+ = S_0\bar{X} + S_1X + S_1S_0$




$S_1^+ = S_0\bar{X} + \bar{S}_1X + \bar{S}_1S_0$




$Y = S_0$

3.3. **Svenska:** Givet den synkrona sekvenskretsen nedan. Antag att start-tillståndet är (000).

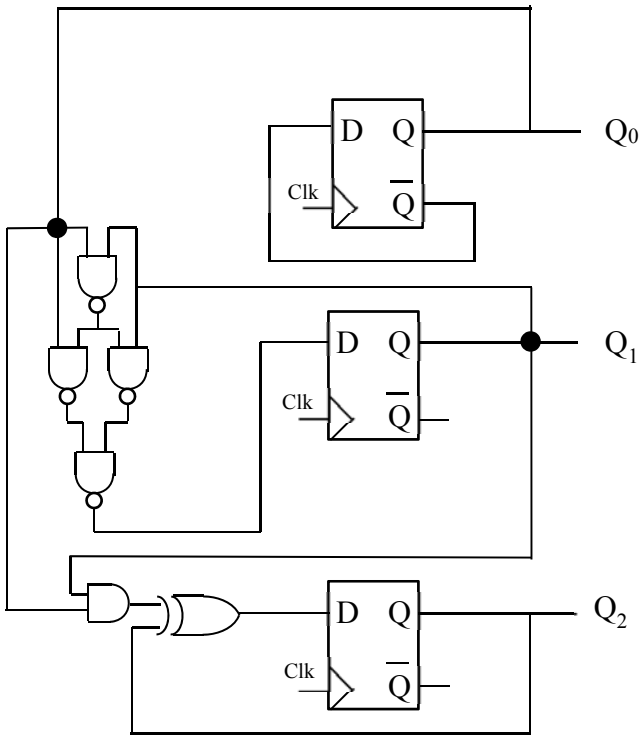
a) Beräkna minimitiden T [ns] mellan klockpulser för säker operation. (1 p)
b) Uppfylls villkoret för "hold time" i kretsen? Om inte, fixa felet. Förklara din lösning. (1 p)
c) Ta fram Boolska uttryck för nästa tillstånd. (1 p)
d) Rita tillståndsdiagrammet. (1 p)

Grindar: $t_{pdNAND} = 2$ ns, $t_{pdAND} = 3$ ns, $t_{pdXOR} = 4$ ns, $t_{cdNAND} = 1$ ns, $t_{cdAND} = 2$ ns, $t_{cdXOR} = 2$ 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 (000).

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)

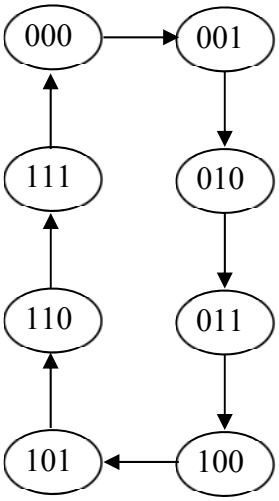
Gates: $t_{pdNAND} = 2$ ns, $t_{pdAND} = 3$ ns, $t_{pdXOR} = 4$ ns, $t_{cdNAND} = 1$ ns, $t_{cdAND} = 2$ ns, $t_{cdXOR} = 2$ ns
Flip-flops: $t_{setup} = 3$ ns, $t_{hold} = 2$ ns, $t_{pcq} = 2$ ns, $t_{ccq} = 1$ ns




**Suggested Solution**

a) Critical Path = $T_{CP}$ = ($Q_0$->AND->XOR->$D_2$)=$t_{pcq}+t_{pdAND}+t_{pdXOR}+t_{setup}$=2+3+4+3=12 ns
b) No. ($Q_0$)'->$D_0$, $T_{ccq}$=1ns<$T_{Hold}$. Fix: Connect Q0 to D0 via an inverter/nand => $T_{ccq}+T_{Nand}$=1+2=3<$T_{Hold}$.
c) $D_0 = \bar{Q}_0,\ D_1 = Q_0 \oplus Q_1,\ D_2 = (Q_0 Q_1) \oplus Q_2$
d)

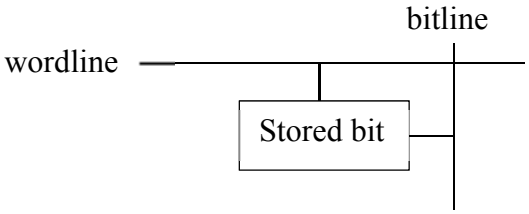
| State | Next State |
|---|---|
| 000 | 001 |
| 001 | 010 |
| 010 | 011 |
| 011 | 100 |
| 100 | 101 |
| 101 | 110 |
| 110 | 111 |
| 111 | 000 |



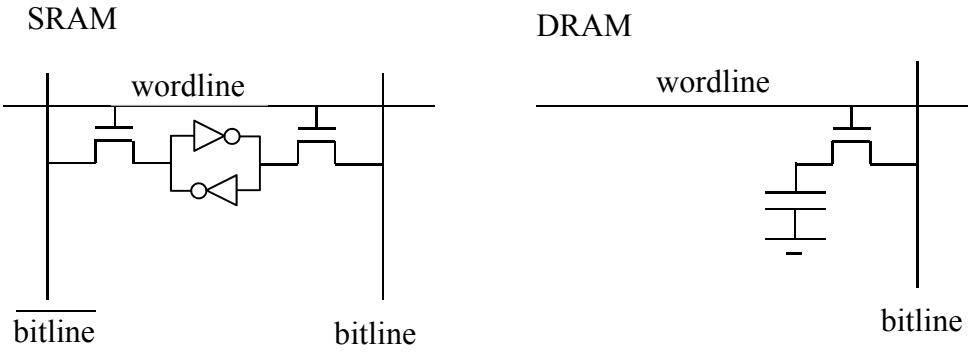
## Del 4/Part 4

4.1. **Svenska:** Ett minne kan grovt beskrivas enligt figuren nedan. Beroende på vilken sorts minne det är ser själva minneselementet olika ut.
a) Hur ser minneselementet för ett SRAM ut? Beskriv kort hur det fungerar vid skrivning och läsning. (1p)
b) Hur ser minneselementet för ett DRAM ut? Beskriv kort hur det fungerar vid skrivning och läsning. (1p)

**English:** A memory can be roughly drawn according to the figure below. Depending on the type of memory, the actual memory cell is different.
a) What does the memory cell for an SRAM look like? Describe shortly how it works during reads and writes. (1p)
b) What does the memory cell for a DRAM look like. Describe shortly how it works during reads and writes. (1p)




**Suggested Solution**




a) SRAM: Write -> A value is presented on the bitline. The inverters has a low driving capability, so the value on the bitline takes over, and the inverter-couple is set to the value on the bitline.

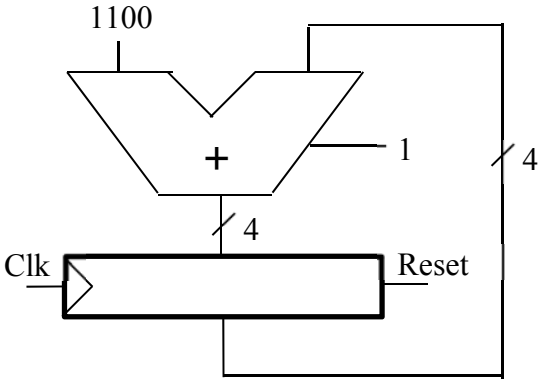
   Read -> The write driver of the bitline is set to 'Z', so the value seen on the bitline is the value of the inverter-couple pointed to by the active wordline.

b) DRAM: Write -> The value on the bitline charges/discharges the capacitor in the DRAM-cell.

   Read -> The write driver of the bitline is set to 'Z', so the value seen on the bitline is the value of the capacitor in DRAM-cell pointed to by the active wordline. The current on the bitline is amplified and is written back after the read. Since the capacitor discharges during the read, any value 1 must be replaced after the read is completed.

4.2. **Svenska:** En räknare består av en 4-bitars adderare och fyra D-vippor. Vipporna klockas samtidigt. Räknaren startar med att alla vippor är vid noll ($q_3q_2q_1q_0 = 0000$). Vad blir hela sekvensen? (2 p)
**English:** A counter consists of a 4-bit adder and four D flip-flops. The flip-flops are clocked simultaneously. The counter starts with all the flip-flops at zero ($q_3q_2q_1q_0 = 0000$). What is the full sequence? (2 p)

0000 -> ____ -> _____ -> etc…




**Suggested Solution**

The circuit adds 1101 (-3) for every turn, so the sequence is

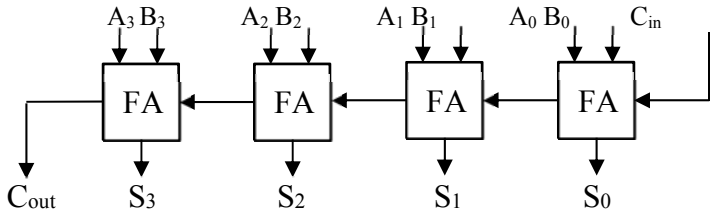
0000->1101->1010->0111->0100->0001->1110->1011->1000->0101->0010->1111->1100->1001->0110->0011->0000

4.3 **Svenska:** Du skall bygga en fyra bitars ripple-carry adderare mha fyra stycken full-adderare (dvs 4 st enbits-adderare).
a) Ställ upp sanningstabellen för fulladderen. (1p)
b) Härled de minimala booleska ekvationerna för utgångarna Sum och Cout. (1 p)
c) Rita den interna strukturen för 4-bitars ripple-carry adderaren. Varje full-adderares interna struktur skall vara synlig. Identifiera den längsta kritiska vägen (vilken ingång/utgång och hur många grindar). (1 p)
d) Hur många transistorer består ripple-carry adderaren av? (1 p)

Antag att antalet transistorer per grind är: NOT: 2, NAND/NOR: 4, AND/OR: 6, XOR: 8 och lägg till 2 transistorer för varje extra ingång. Förklara eventuella antaganden.

**English:** You shall build a four bit ripple-carry adder using four full-adders (i.e., 4 onebit adders).
a) Derive the truth table for the fulladder. (1 p)
b) Derive the minimized boolean equations for the outputs Sum and Cout. (1 p)
c) Draw the internal structure of the four-bit ripple-carry adder. Every full-adders internal structure should be visible. Identify the longest critical path (which input/output and the number of gates). (1 p)
d) How many transistors is the ripple-carry adder built of? (1 p)

Assume the number of transistors per gate as NOT: 2, NAND/NOR: 4, AND/OR: 6, XOR: 8, and add 2 transistors per extra input. Explain your assumptions if needed.




a)

| A | B | $C_{in}$ | $C_{out}$ | Sum |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |

| A | B | $C_{in}$ | $C_{out}$ | Sum |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

b)

| A \ $BC_{in}$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

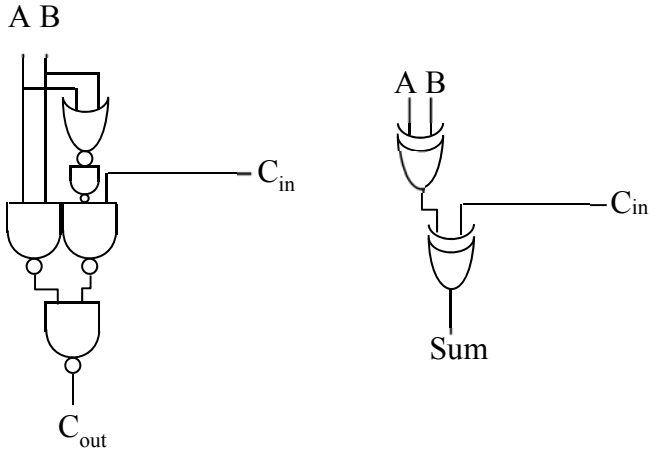
$C_{out} = AC_{in} + BC_{in} + AB,$

| A \ $BC_{in}$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

$Sum = A \oplus B \oplus C_{in}$

c) $C_{out} = AC_{in} + BC_{in} + AB = C_{in}(A+B) + AB = \overline{\overline{C_{in}(A+B)}\,\overline{AB}} = \overline{(\overline{C_{in}(A+B)})(\overline{AB})}$

This rewrite makes the critical path slightly faster (~0.5$T_{NAND2}$/FA), except for the first bit in the FA-chain, at the same area as the straight-forward circuit.



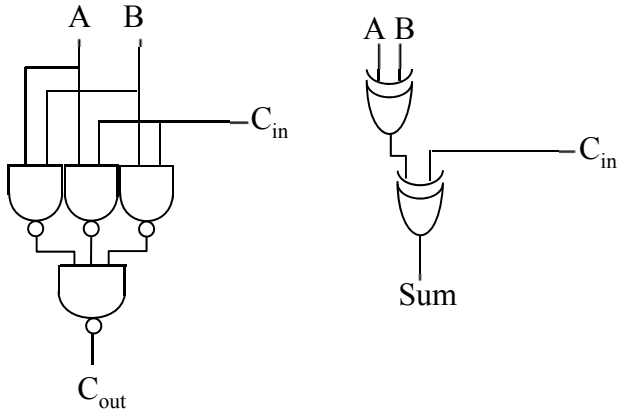

Bit 0: $T_{CP}$=(AB)->$T_{NOR2}$+$T_{inv}$+2*$T_{NAND2}$->$C_{out}$)=1.5$T_{NAND2}$+2*$T_{NAND2}$=3.5$T_{NAND2}$

Bit 1-3:$T_{CP}$=($C_{in}$)->2*$T_{NAND2}$->($C_{out}$)

$CP_{(N\ bits\ ripple\text{-}carry\text{-}adder)}$=$T_{CP}$=$T_{NOR2}$+$T_{inv}$+N*2*$T_{NAND2}$=$T_{NOR2}$+$T_{inv}$+4*2$T_{NAND2}$=9.5$T_{NAND2}$

Area=4*($A_{NOR2}$+$A_{inv}$+3*$A_{NAND2}$+2*$A_{XOR2}$)=4*(4+2+3*4+2*8)=4*34=136 Transistors.

The straight-forward solution uses a 3-input NAND, and a two-level $C_{out}$-circuit (i.e., no NOR-gate):




Bit 0-4: $T_{CP}$=($C_{in}$)->$T_{NAND2}$+$T_{NAND3}$=2.5$T_{NAND2}$

$CP_{(N\ bits\ ripple\text{-}carry\text{-}adder)}$=$T_{CP}$=N*2.5$T_{NAND2}$=10$T_{NAND2}$.

Area=4*(3*$A_{NAND2}$+$A_{NAND3}$+2*$A_{XOR2}$)=4*(3*4+6+2*8)=4*34=134 Transistors.

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

*X=0xC080 0000*

*Y=0x3F80 0000*

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

b) Utför multiplikationen W=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*

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

b) Calculate the multiplication W=X*Y. Answer in 32-bit IEEE 754 floating point form. (1p)

**Suggested Solution**

X=1(100 0000 1)000 0000 …. = $-2^{(129-127)}$*(1).0=-4
Y=0(011 1111 1)000 0000 …. = $+2^{(127-127)}$*(1).0=1

a) Z=-4+1=-3
   -3=$-2^{(128-127)}$*(1).5 = 1(100 0000 0)100 0000 …. = 0xC040 0000
b) W=X*Y = -4*1 = -4 (=X) = 0xC080 0000