IE1204/5 Digital Design **example exam**

- **Part A1** (10p in total) eight short ***Analysis* tasks,** 1p or 2p each

- They are corrected only as Right/Wrong!
- You need to get at least 6p on A1, otherwise parts A2 and B will not be marked at all!

- **Part A2** (10p in total) two ***Methodology*** tasks.

Marked only if there is at least 6p in part A1.

- **Part B** (10p in total) two ***Design problems***.

Marked Only if there is at least 11p in parts A1+A2.

Pass-limit for the entire exam is at least **11p**, A1+A2+(B).
You can pass with no points from part B.

William Sandqvist william@kth.se

Grading scale

A1 10p **A2** 10p **B** 10p

Less than 11p på A1+A2 ⇒ F
Part B will not be marked!

Max 30

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

Less than 6p at A1 ⇒ F
Part A" will not be marked!

- You can theoretical get grade C *without* solving part B.

Part **A1** Analysis

- Right or wrong, 0p / 1p / 2p
- Minimum 6p out of 10p, otherwise parts A2 and B will not be marked at all!

?: Part **A1** (**2**/**1**/**0**) task **1.**

$$f(x,y,z) = z(\overline{x} + x\overline{y}) + xy\overline{z} + xyz = \{SoP\}_{\min} = ?$$

!: Part **A1** (**2**/**1**/**0**) task **1.**

$$f(x,y,z) = z(\overline{x} + x\overline{y}) + xy\overline{z} + xyz = \{SoP\}_{\min} = ?$$

$$f(x,y,z) = \overline{x}\,\overline{y}\,z + \overline{x}yz + x\overline{y}\,z + xyz + xy\overline{z}$$

!: Part **A1** (**2**/**1**/**0**) task **1.**

$$f(x,y,z) = z(\overline{x} + x\overline{y}) + xy\overline{z} + xyz = \{SoP\}_{\min} = ?$$

$$f(x,y,z) = \overline{x}\,\overline{y}\,z + \overline{x}yz + x\overline{y}z + xyz + xy\overline{z}$$

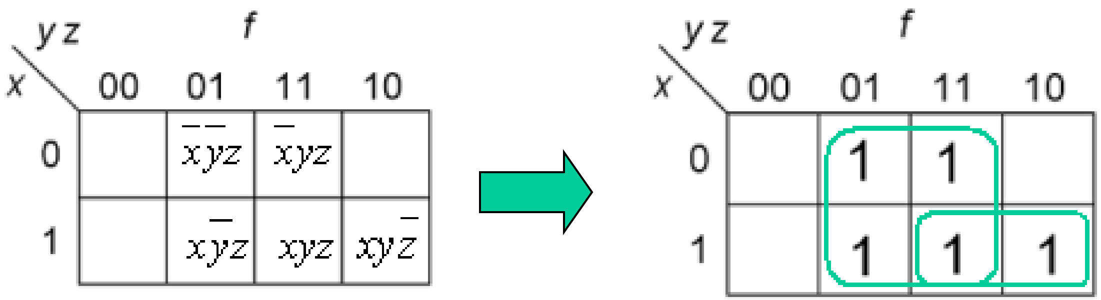
f

$x \backslash yz$	00	01	11	10
0		$\overline{x}\,\overline{y}z$	$\overline{x}yz$	
1		$x\overline{y}z$	xyz	$xy\overline{z}$

!: Part **A1** (2/1/0) task **1.**

$$f(x,y,z) = z(\overline{x} + x\overline{y}) + xy\overline{z} + xyz = \{SoP\}_{\min} = ?$$

$$f(x,y,z) = \overline{x}\,\overline{y}\,z + \overline{x}yz + x\overline{y}z + xyz + xy\overline{z}$$



William Sandqvist william@kth.se

!: Part **A1** (2/1/0) task **1.**

$$f(x,y,z) = z(\bar{x} + x\bar{y}) + xy\bar{z} + xyz = \{SoP\}_{\min} = ?$$

$$f(x,y,z) = \bar{x}\,\bar{y}\,z + \bar{x}yz + x\bar{y}z + xyz + xy\bar{z}$$

f

x \ yz	00	01	11	10
0		$\bar{x}\bar{y}z$	$\bar{x}yz$	
1		$x\bar{y}z$	xyz	$xy\bar{z}$

f

x \ yz	00	01	11	10
0		1	1	
1		1	1	1

$$f(x,y,z) = \{SoP\}_{\min} = \boxed{z + xy}$$

?: Part **A1** (1/0) task **2.**

2’s complement-representation of an 8-bit number .

$$(B7)_{16} - (A6)_{16} = ?_{16}$$

$$\pm ?_{10} - \pm ?_{10} = \pm ?_{10}$$

!: Part **A1** (**1**/**0**) task **2.**

2's complement-representation of an 8-bit number .

Signbit!

$$(B7)_{16} - (A6)_{16} = ?_{16}$$

$$\pm ?_{10} - \pm ?_{10} = \pm ?_{10}$$

$$(B7)_{16} = (10110111)_2 = (-01001001)_2 = (-49)_{16} = (-73)_{10}$$

$$(A6)_{16} = (10100110)_2 = (-01011010)_2 = (-5A)_{16} = (-90)_{10}$$

$$(B7)_{16} - (A6)_{16} = (B7)_{16} + (5A)_{16} = (11)_{16}$$

$$\boxed{(-73)_{10} - (-90)_{10} = (17)_{10}}$$

!: Part **A1** (2/1/0) task **3.**

	$x_3x_2x_1x_0$	f
0	0000	1
1	0001	0
2	0010	1
3	0011	–
4	0100	0
5	0101	1
6	0110	0
7	0111	0

	$x_3x_2x_1x_0$	f
8	1000	1
9	1001	–
10	1010	–
11	1011	0
12	1100	1
13	1101	0
14	1110	–
15	1111	0

$$f(x_3x_2x_1x_0) = \{PoS\}_{\min} = ?$$

!: Part **A1** (2/1/0) task **3.**

	$x_3x_2x_1x_0$	f		$x_3x_2x_1x_0$	f
0	0000	1	8	1000	1
1	0001	0	9	1001	–
2	0010	1	10	1010	–
3	0011	–	11	1011	0
4	0100	0	12	1100	1
5	0101	1	13	1101	0
6	0110	0	14	1110	–
7	0111	0	15	1111	0

$$f(x_3x_2x_1x_0) = \{PoS\}_{\min} = ?$$

Groupings of 0:es

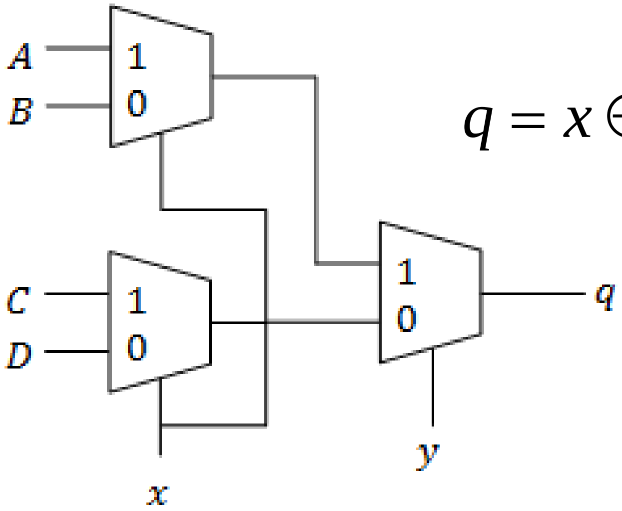
$x_3x_2 \backslash x_1x_0$	00	01	11	10
00	^{0}1	^{1}0	^{3}-	^{2}1
01	^{4}0	^{5}1	^{7}0	^{6}0
11	^{12}1	^{13}0	^{15}0	^{14}-
10	^{8}1	^{9}-	^{11}0	^{10}-

$$f(x_3x_2x_1x_0) =$$

$$(\overline{x}_0 + x_2)(\overline{x}_0 + \overline{x}_3)(\overline{x}_0 + \overline{x}_1)(x_0 + \overline{x}_2 + x_3)$$

William Sandqvist william@kth.se

?: Part **A1** (1/0) task **4.**



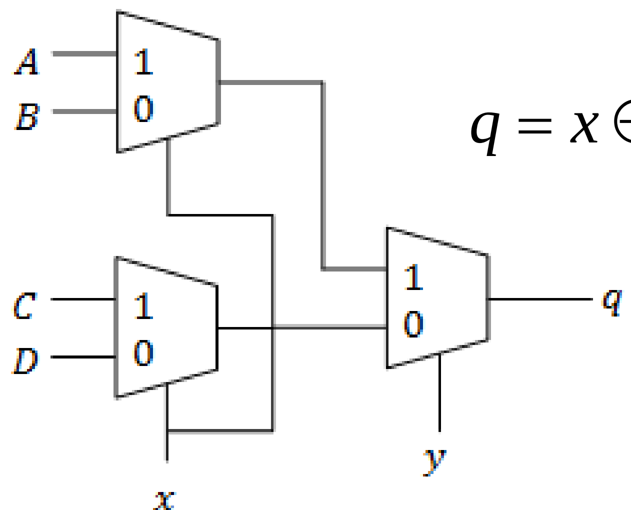
$$q = x \oplus y$$

$A, B, C, D = ?$

!: Part **A1** (**1**/**0**) task **4.**

Lookup-table!

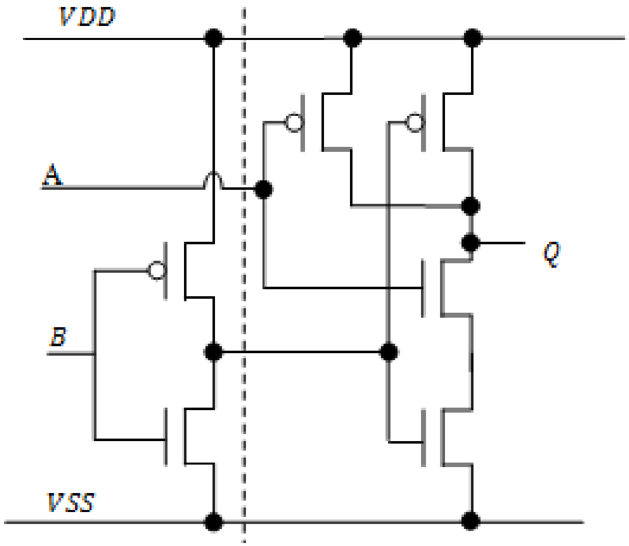
x	y	XOR	
0	0	0	$= D$
0	1	1	$= B$
1	0	1	$= C$
1	1	0	$= A$



$$q = x \oplus y$$

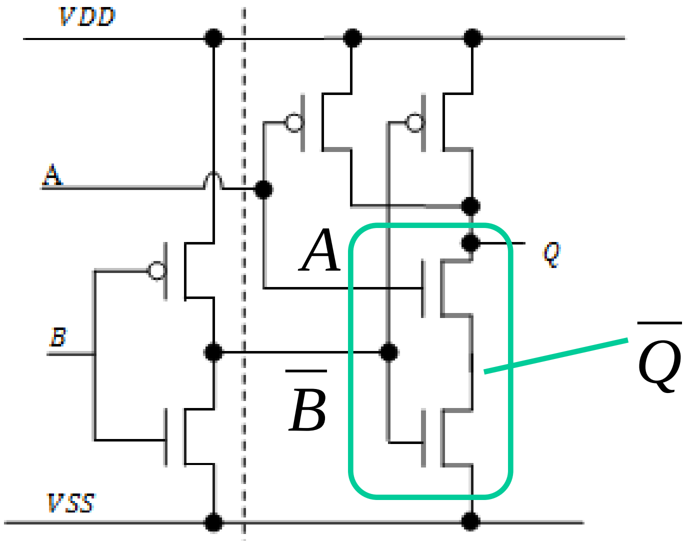
$$\boxed{A = 0 \quad B = 1 \quad C = 1 \quad D = 0}$$

?: Part **A1** (1/0) task **5.**



$$Q(A,B) = ?$$

!: Part **A1** (1/0) task **5.**

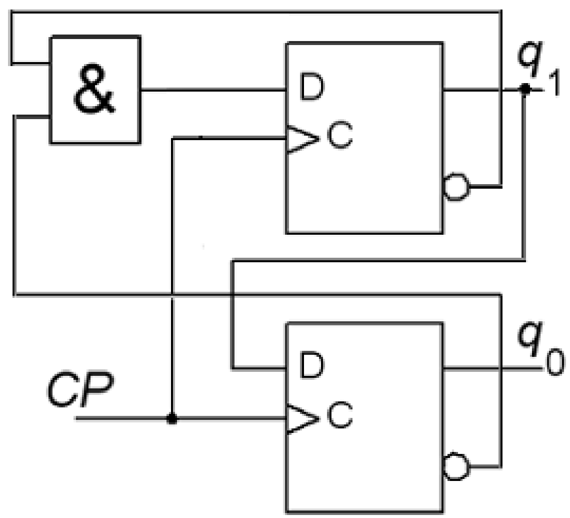


$$Q(A,B) = ?$$

PullDown-circuit

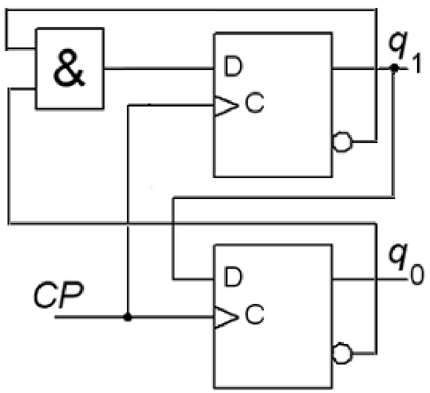
$$\overline{Q} = A \cdot \overline{B} \quad \Rightarrow \quad Q = \overline{A \cdot \overline{B}} = \{dM\} = \boxed{\overline{A} + B}$$

?: Part **A1** (1/**0**) task **6.**



$$(q_1 q_0) = 00 \rightarrow ?? \rightarrow ?? \rightarrow ?? \rightarrow ?? \ldots$$

!: Part **A1** (1/0) task **6.**



Next state

$$q_1^+ = \overline{q}_1 \cdot \overline{q}_0$$

$$q_0^+ = q_1$$

$q_1 q_0$	$q_1^+ = \overline{q}_1 \cdot \overline{q}_0$	$q_0^+ = q_1$	$q_1^+ q_0^+$
00	$1 = 1 \cdot 1$	0	10
10	$0 = 0 \cdot 1$	**1**	01
01	$0 = 1 \cdot 0$	0	00

$$\boxed{(q_1 q_0) = 00 \rightarrow 10 \rightarrow 01 \rightarrow 00 \ldots}$$

William Sandqvist william@kth.se

?: Part **A1** (1/0) task **7.**

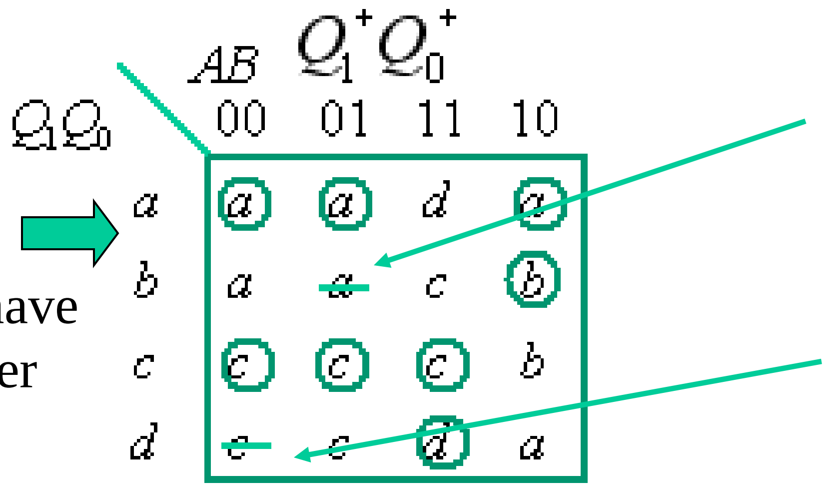
Complete this flow chart (for an asynchronous sequential circuit) with circles around stable states and by crossing out the conditions that can not be reached.

$Q_1^+Q_0^+$

Q_1Q_0 \ AB	00	01	11	10
a	a	a	d	a
b	a	a	c	b
c	c	c	c	b
d	c	c	d	a

!: Part **A1** (1/0) task **7.**

Identify the states that have the row letter



Golden rule:
Only one bit can change at a time.

Inexpensive point for all those who did not "skip" section on asynchronous sequential circuits ...

- **Conclusion:** Do not skip any sections of the course!

?: Part **A1** (**1**/**0**) task **8.**

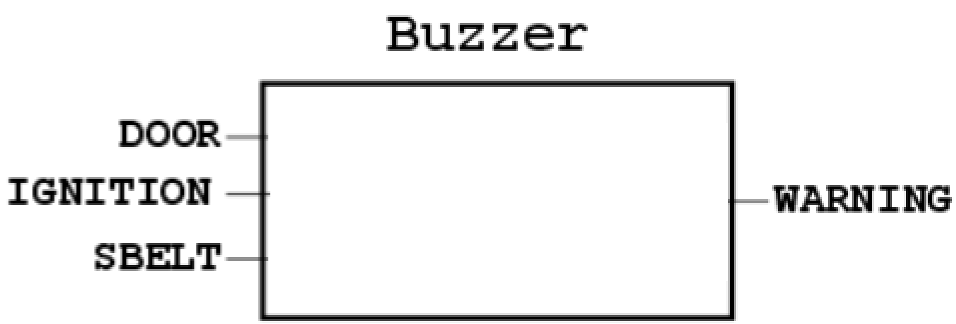
```
-- comment: example of buzzer circuit
entity BUZZER is
   port (DOOR, IGNITION, SBELT: in std_logic;
                       WARNING: out std_logic);
end BUZZER;

architecture behavioral of BUZZER is
begin
  WARNING <= (not DOOR and IGNITION) or
             (not SBELT and IGNITION);
end behavioral;
```

- Draw entity-box with signal names and circuit diagram …

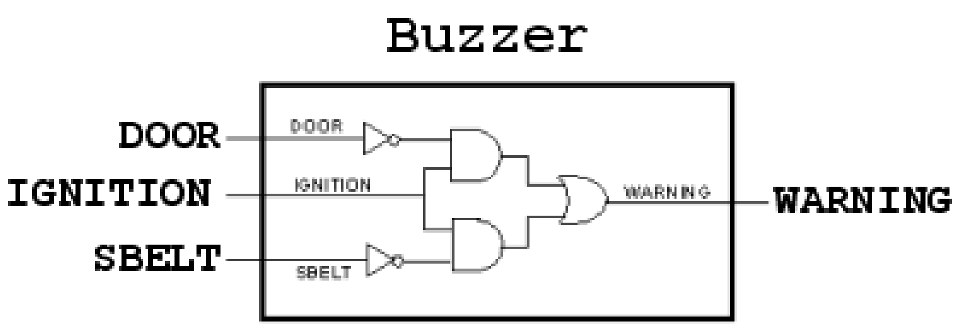
!: Part **A1** (1/0) task **8.**

```
entity BUZZER is
   port (DOOR, IGNITION, SBELT: in std_logic;
                      WARNING: out std_logic);
end BUZZER;
```



!: Part **A1** (1/0) task **8.**

```
architecture behavioral of BUZZER is
begin
  WARNING <= (not DOOR and IGNITION) or
             (not SBELT and IGNITION);
end behavioral;
```



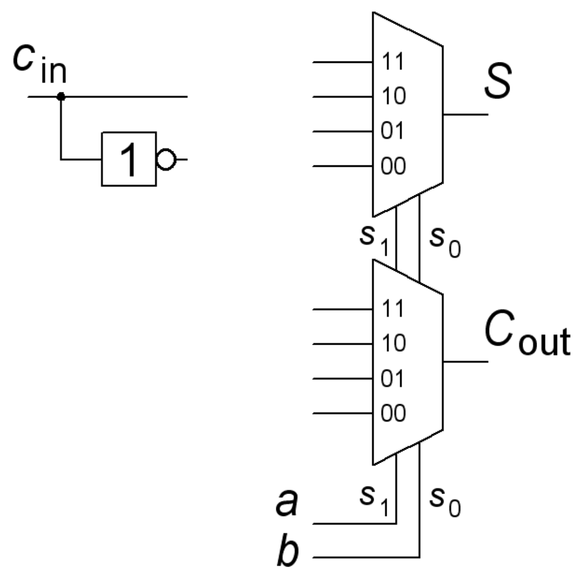
Part **A2** Methodology

- Will be marked if you get at least 6p in part A1
- You need to get at least 11 p in A1 + A2 for the part B to be marked

?: Part **A2** (5p) task **9.**

- First, complete the full adder truth table.
- Then construct a full adder from two 4:1 MUXes.
- We assume the Carry signal cin also available in the complemented form (as shown on Figure).

a	b	c_{in}	C_{out}	S
0	0	0	0	0
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



!: Part **A2** (5p) task **9.**

a	b	c_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

S

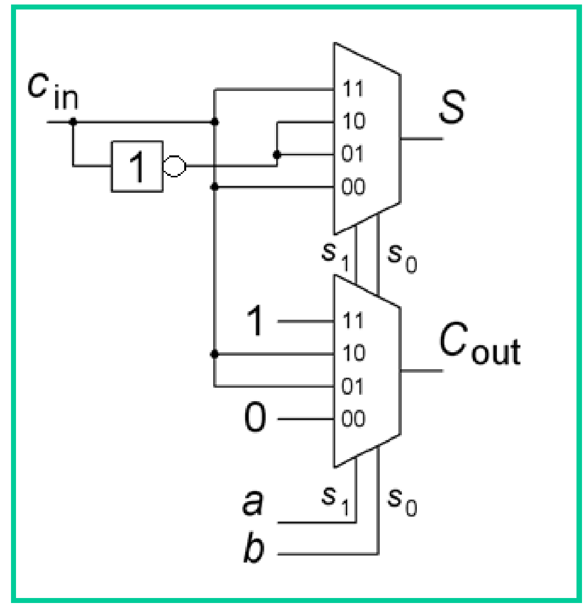
c_{in} \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

c_{in} $\quad \overline{c_{in}}$ $\quad c_{in}$ $\quad \overline{c_{in}}$

C_{out}

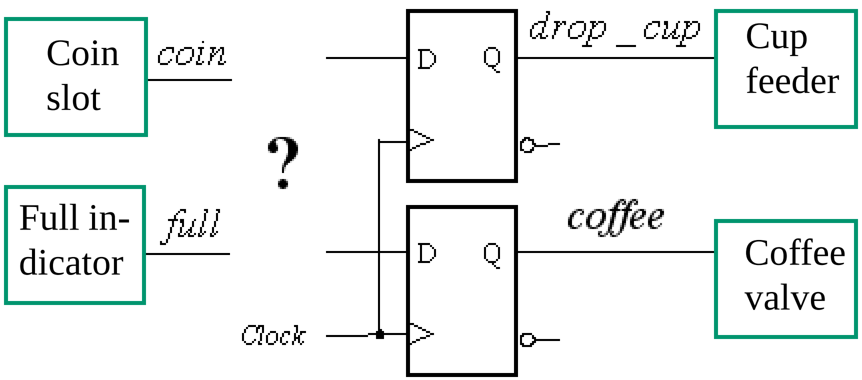
c_{in} \ ab	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$0 \quad c_{in} \quad 1 \quad c_{in}$



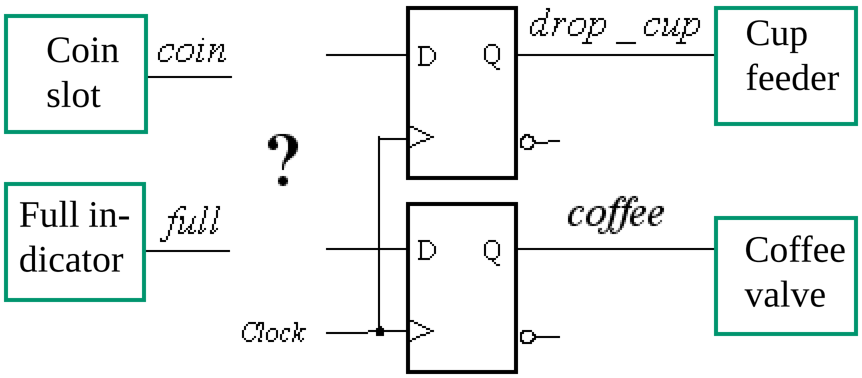
William Sandqvist william@kth.se

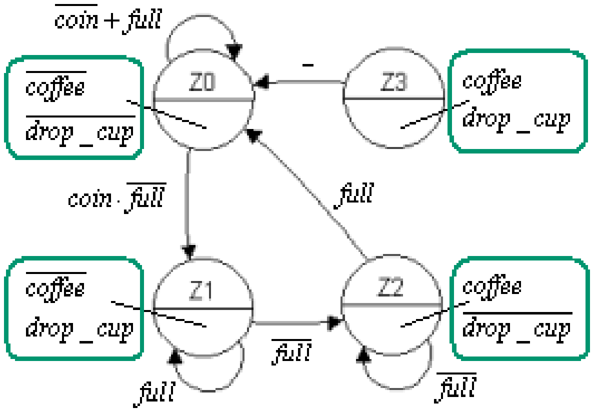
?: Part **A2** (5p) task **10.**



- A coffee machine has two input signals: **coin** from the coin slot to indicate that a coin passed a photocell there and **full** from a user who observes the plastic cup to be filled. **Coin = 1** when the coin passed the photocell. **full = 1** when the cup is full.
- The coffee machine has two output signals: **drop_cup** to a unit containing plastic cups and **coffee** to a magnetic valve for filling the coffee. The cap unit drops a cup was once **drop_cup** becomes ”1” and coffee is filled as long as **coffee = 1**.

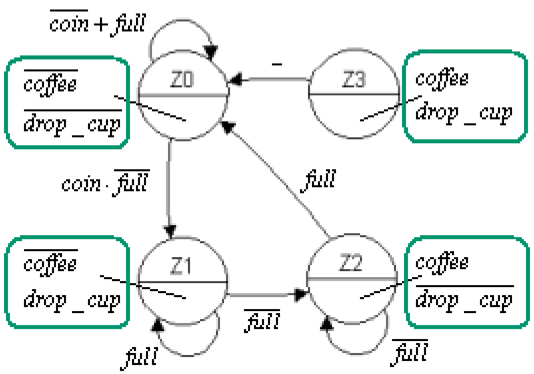
?: Part **A2** (5p) task **10.**





- Construct a synchronous Moore machine that follows the given state diagram.
- State assignments must control the outputs directly. No output decoder is used.
- Use positive edge-triggered D flip-flops and gates of your choice.
- Draw a complete circuit diagram.

!: Part **A2** (5p) task **10.**



I: full coin

$q_1 q_0$ \ $i_1 i_0$	00	01	11	10	
Z0: 00	0 0	0 1	0 0	0 0	$\overline{coffee}\ \overline{drop\_cup}$
Z1: 01	1 0	1 0	0 1	0 1	$\overline{coffee}\ drop\_cup$
Z3: 11	0 0	0 0	0 0	0 0	$coffee\ drop\_cup$
Z2: 10	1 0	1 0	0 0	0 0	$coffee\ \overline{drop\_cup}$

$q_1^+ q_0^+$

q_1^+

$q_1 q_0$ \ $i_1 i_0$	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	0	0	0	0
10	1	1	0	0

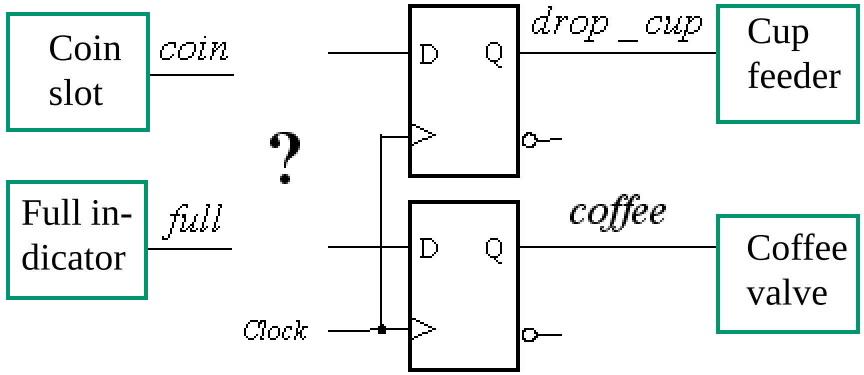
q_0^+

$q_1 q_0$ \ $i_1 i_0$	00	01	11	10
00	0	1	0	0
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

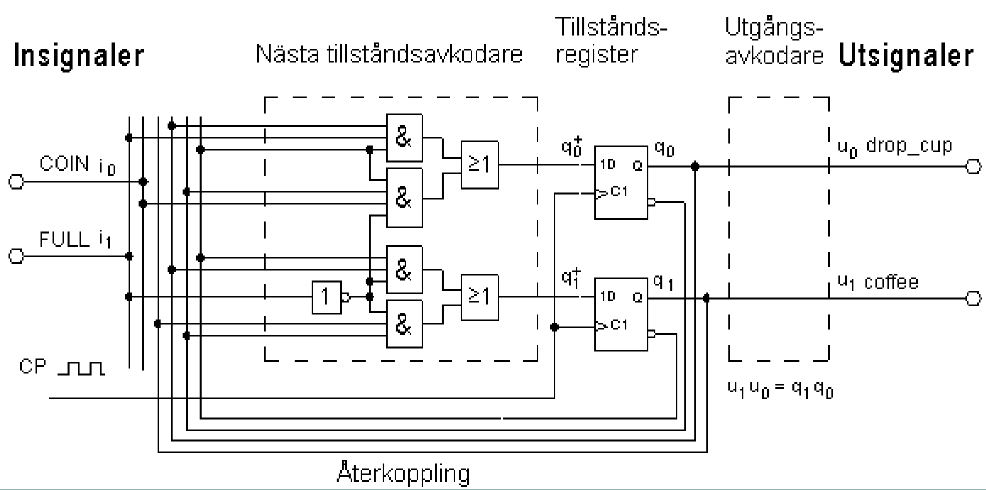
$$q_1^+ = \overline{q}_1 q_0 \overline{i}_1 + q_1 \overline{q}_0 \overline{i}_1 \qquad q_0^+ = \overline{q}_1 \overline{q}_0 \overline{i}_1 i_0 + \overline{q}_1 q_0 i_1$$

William Sandqvist william@kth.se

!: Part **A2** (5p) task **10.**



$$q_1^+ = \overline{q_1} q_0 \overline{i_1} + q_1 \overline{q_0}\, \overline{i_1} \qquad q_0^+ = \overline{q_1}\, \overline{q_0}\, \overline{i_1} i_0 + \overline{q_1} q_0 i_1$$



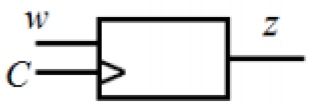
William Sandqvist william@kth.se

Part **B** Design

- We mark part B (10p) if you get at least 11p at A1 + A2
- Tasks can always be solved in different ways. We mark as far as possible taking into account possible errors from previous steps
- Digital design is a creative process

?: Part **B** (5p) task **13.**

Sequence Detector.



Obtain a minimal state table (show that it is minimal) for a synchronous sequential circuit of Moore-type with an input signal (*w*), and an output signal (*z*). The sequence net will generate the output 1 if it detects either input sequence 110 or 101, also overlapping sequences are valid (eg. 1101, is 110 followed by 101, will give output 00011). Derive the state diagram.

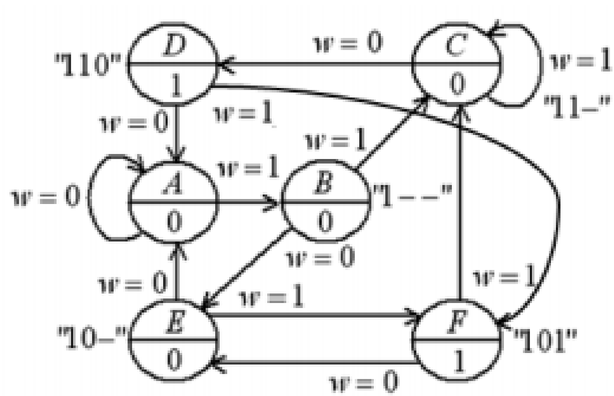
a) (4p) Set up the circuit's **state table**, **show** that it is **minimal**, and draw the **state diagram**.

b) (2p) Use Gray code to encode the states and set up the **encoded state table**. Obtain the **minimized expressions** for the **next state** and **output value**. No schematic of the circuit is needed to be drawn.

!: Part **B** (5p) task **13.**

a) (4p)

present	next $w=0$	next $w=1$	out z
A	A	B	0
B	E	C	0
C	D	C	0
D	A	F	1
E	A	F	0
F	E	C	1



Minimal number of states

$(ABCE)(DF)$

$A_0 \rightarrow (\underline{A}BCE) \quad A_1 \rightarrow (A\underline{B}CE)$

$B_0 \rightarrow (ABC\underline{E}) \quad B_1 \rightarrow (AB\underline{C}E)$

$C_0 \rightarrow (\underline{D}F) \quad C_1 \rightarrow (AB\underline{C}E)$

$E_0 \rightarrow (\underline{A}BCE) \quad E_1 \rightarrow (D\underline{F})$

$(AB)(C)(DF)(E)$

$(AB)(C)(DF)(E)$

$A_0 \rightarrow (\underline{A}B) \quad A_1 \rightarrow (A\underline{B})$

$B_0 \rightarrow (E) \quad B_1 \rightarrow (C)$

$(A)(B)(C)(DF)(E)$

$D_0 \rightarrow (A) \quad D_1 \rightarrow (D\underline{F})$

$F_0 \rightarrow (E) \quad F_1 \rightarrow (C)$

$\boxed{(A)(B)(C)(D)(E)(F)}$

Was minimal from the beginning!

William Sandqvist william@kth.se

!: Part **B** (5p) task **13.**

present	next $w=0$	next $w=1$	out z
A	A	B	0
B	E	C	0
C	D	C	0
D	A	F	1
E	A	F	0
F	E	C	1

Gray code

present	$q_2\,q_1\,q_0$	next $q_2^+q_1^+q_0^+$ $w=0$	next $q_2^+q_1^+q_0^+$ $w=1$	out
A	000	000	001	0
B	001	110	011	0
C	011	010	011	0
D	010	000	111	1
E	110	000	111	0
F	111	110	011	1

$q_2^+q_1^+q_0^+ = f(q_2, q_1, q_0, w)$

wq_2 \ q_1q_0	00	01	11	10
00	000	110	010	000
01	– – –	– – –	110	000
11	– – –	– – –	011	111
10	001	011	011	111

Kmap form

$q_2^+ = f(q_2, q_1, q_0, w)$

wq_2 \ q_1q_0	00	01	11	10
00	0	1	0	0
01	–	–	1	0
11	–	–	0	1
10	0	0	0	1

$$\boxed{q_2^+ = \overline{q}_1 q_0 + \overline{w} q_2 + q_1 \overline{q}_0}$$

$q_1^+ = f(q_2, q_1, q_0, w)$

wq_2 \ q_1q_0	00	01	11	10
00	0	1	1	0
01	–	–	1	0
11	–	–	1	1
10	0	1	1	1

$$\boxed{q_1^+ = q_0 + w q_1}$$

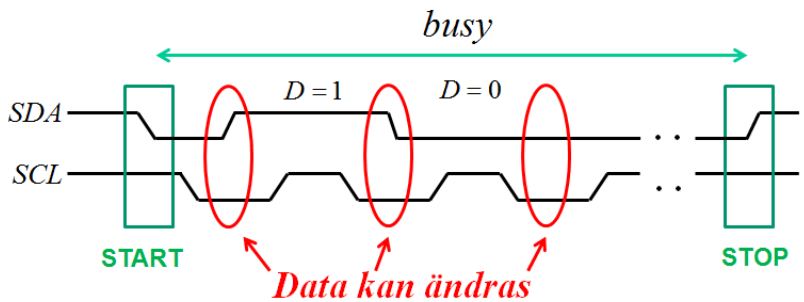
$q_0^+ = f(q_2, q_1, q_0, w)$

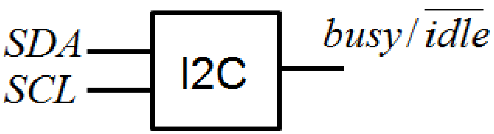
wq_2 \ q_1q_0	00	01	11	10
00	0	0	0	0
01	–	–	0	0
11	–	–	1	1
10	1	1	1	1

$$\boxed{q_0^+ = w}$$

William Sandqvist william@kth.se

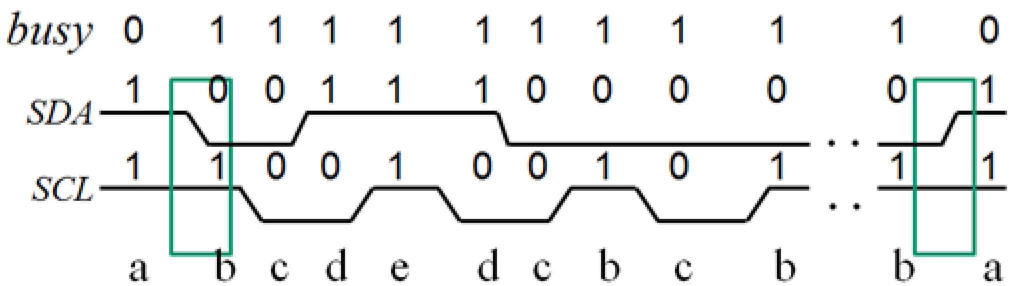
?: Part **B** (5p) task **12.**

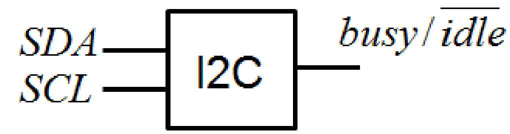




• In order to study the I^2C data transfer we want to construct a Moore-equivalent asynchronous sequential circuit which gives output signal *busy = 1* during the time from the start signal to the stop signal. When no data communication occurs *busy = 0*

!: Part **B** (5p) task **12.**





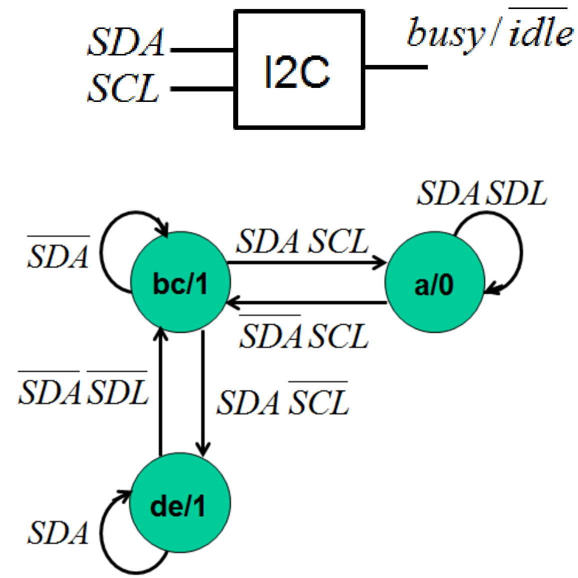
	SDA SCL 00	01	11	10	q
a	–	b	a	*	0
b	c	b	a	–	1
c	c	b	–	d	1
d	c	–	e	d	1
e	–	*	e	d	1

- In a state *a* we "wait" for the start (*b*) , therefore the input signal 10 is impossible (marked with *). The protocol prohibits the alteration of data SDA when SCL is **high**. Therefore, the input signal 01 is impossible in state *e* (marked with *). This gives two additional don’t care positions in the table.
- One see immediately which states can be merged.

!: Del **B** (5p) uppg **12.**

	SDA SCL 00	01	11	10	q
a	–	b	a	*	0
b	c	b	a	–	1
c	c	b	–	d	1
d	c	–	e	d	1
e	–	*	e	d	1

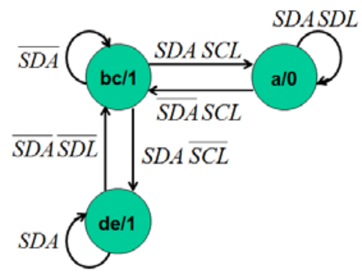
	SDA SCL 00	01	11	10	q
a	–	bc	a	–	0
bc	bc	bc	a	de	1
de	bc	–	de	de	1

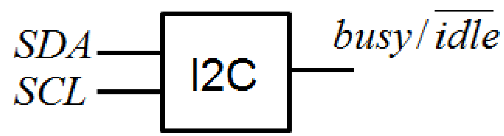


• We can use Gray code for state assignment: *a* = 00, *bc* = 01, *de* = 11. The unused state 01 can have any next state except 01.

!: Del **B** (5p) uppg **12.**

	SDA SCL 00	01	11	10	q
a	–	bc	a	–	0
bc	bc	bc	a	de	1
de	bc	–	de	de	1





q_1q_0	SDA SCL 00	01	11	10	$busy$
00	–	01	00	–	0
01	01	01	00	11	1
11	01	–	11	11	1
10	$\neq 10$	$\neq 10$	$\neq 10$	$\neq 10$	–

$q_1^+ q_0^+$

$busy =$
$= q_0$

q_1q_0	SDA SCL 00	01	11	10
00	–	0	0	–
01	0	0	0	1
11	0	–	1	1
10	–	–	–	–

$$q_1^+ = SDA(\overline{SCL} + q_1)$$

q_1q_0	SDA SCL 00	01	11	10
00	–	1	0	–
01	1	1	0	1
11	1	–	1	1
10	–	–	–	–

$\neq 10$

$$q_0^+ = (\overline{SDA} + \overline{SCL} + q_1)$$

- Note that the resulting implementation is hazard-free becuase all adjacent 1's are covered by the same implicant.

Good Luck!