Problem 1

(a) The disjunctive normal form is the $\vee$-sum of the minterms in the on-set:

$$\begin{aligned} f_{\text{DNF}} &= m_0 \vee m_2 \vee m_6 \vee m_7 \vee m_8 \vee m_{10} \vee m_{14} \\ &= x_1'x_2'x_3'x_4' \vee x_1'x_2'x_3x_4' \vee x_1'x_2x_3x_4' \vee x_1'x_2x_3x_4 \vee \\ &\quad\; x_1x_2'x_3'x_4' \vee x_1x_2'x_3x_4' \vee x_1x_2x_3x_4' \end{aligned}$$

(b) The minimal disjunctive form can be found by looking at the Karnaugh map below:

$$f_{\text{MDF}} = x_2'x_4' \vee x_3x_4' \vee x_1'x_2x_3$$

(c) The minimal conjunctive form, with corresponding Karnaugh map is:

$$f_{\text{MCF}} = (x_2' \vee x_3)(x_2 \vee x_4')(x_1' \vee x_4')$$

Note that $x_3' \vee x_4$ shall not be used.

f_{MDF} (x_1x_2 \ x_3x_4)	00	01	11	10
00	1	0	0	1
01	0	0	1	1
11	0	0	0	1
10	1	0	0	1

f_{MCF} (x_1x_2 \ x_3x_4)	00	01	11	10
00	1	0	0	1
01	0	0	1	1
11	0	0	0	1
10	1	0	0	1

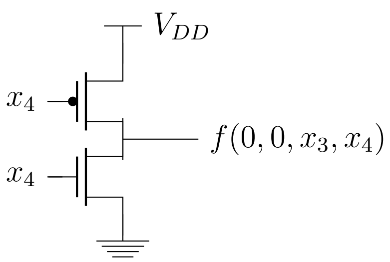
(d) f can be expressed on RMF as:

$$f_{\text{RMF}} = 1 \oplus x_2 \oplus x_4 \oplus x_2x_3 \oplus x_2x_4 \oplus x_1x_2x_3x_4$$

(e) $f(0,0,x_3,x_4)$ gives the following truth table:

x_3	x_4	$f(0,0,x_3,x_4)$
0	0	1
0	1	0
1	0	1
1	1	0

which gives $f(0,0,x_3,x_4) = x_4'$. With CMOS transistors we have:



Problem 2

(a) A state-transition table is

	00	01, 10	11
s_{--}	$s_{0-}/0$	$s_{1-}/0$	$s_{2-}/0$
s_{0-}	$s_{00}/0$	$s_{10}/0$	$s_{20}/0$
s_{1-}	$s_{01}/0$	$s_{11}/0$	$s_{21}/0$
s_{2-}	$s_{02}/0$	$s_{12}/0$	$s_{22}/0$
s_{00}	$s_{00}/0$	$s_{10}/0$	$s_{20}/0$
s_{10}	$s_{01}/0$	$s_{11}/0$	$s_{21}/0$
s_{20}	$s_{02}/0$	$s_{12}/0$	$s_{22}/0$
s_{01}	$s_{00}/0$	$s_{10}/0$	$s_{20}/0$
s_{11}	$s_{01}/0$	$s_{11}/0$	$s_{21}/0$
s_{21}	$s_{02}/0$	$s_{12}/0$	$s_{22}/1$
s_{02}	$s_{00}/0$	$s_{10}/0$	$s_{20}/1$
s_{12}	$s_{01}/0$	$s_{11}/0$	$s_{21}/1$
s_{22}	$s_{02}/0$	$s_{12}/1$	$s_{22}/0$.

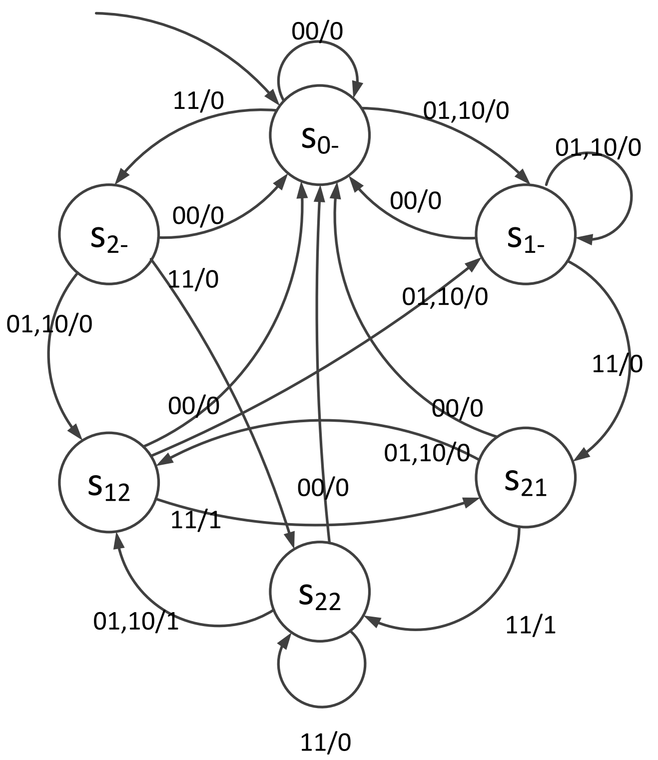
To find the reduced form of the given graph we apply the RF algorithm. We obtain the following partitions:

$$P1 : \{s_{--}, s_{0-}, s_{1-}, s_{2-}, s_{00}, s_{01}, s_{02}, s_{10}, s_{11}, s_{20}\}, \{s_{12}, s_{21}\}, \{s_{22}\}$$
$$P2 : \{s_{--}, s_{0-}, s_{00}, s_{01}, s_{02}\}, \{s_{1-}, s_{10}, s_{11}\}, \{s_{2-}, s_{20}\}, \{s_{12}\}, \{s_{21}\}, \{s_{22}\}$$
$$P3 : \underbrace{\{s_{--}, s_{0-}, s_{00}, s_{01}, s_{02}\}}_{s_{0-}}, \underbrace{\{s_{1-}, s_{10}, s_{11}\}}_{s_{1-}}, \underbrace{\{s_{2-}, s_{20}\}}_{s_{2-}}, \{s_{12}\}, \{s_{21}\}, \{s_{22}\}.$$

Thus, we have the reduced set of states $\{s_{-0}, s_{1-}, s_{2-}, s_{12}, s_{21}, s_{22}\}$, and the reduced state-transition table:

	00	01, 10	11
s_{0-}	$s_{0-}/0$	$s_{1-}/0$	$s_{2-}/0$
s_{1-}	$s_{0-}/0$	$s_{1-}/0$	$s_{21}/0$
s_{2-}	$s_{0-}/0$	$s_{12}/0$	$s_{22}/0$
s_{12}	$s_{0-}/0$	$s_{1-}/0$	$s_{21}/1$
s_{21}	$s_{0-}/0$	$s_{12}/0$	$s_{22}/1$
s_{22}	$s_{0-}/0$	$s_{12}/1$	$s_{22}/0$,

which has a state-transition graph



It is clear that the given graph is also in minimized form for the starting state s_{0-}.

(b) Since the number of states in the minimal graph is 6, we will need 3 state variables, $q_1q_2q_3$. We use the following state assignment (other state assignments are also possible)

s	q_1	q_2	q_3
s_-	0	0	0
s_{1-}	0	0	1
s_{2-}	0	1	0
s_{12}	1	0	0
s_{21}	1	1	0
s_{22}	1	1	1

Which results in the following state-transition table

q_1	q_2	q_3	x_1	x_2	q_1^+	q_2^+	q_3^+	u
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	1	0	0
0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	1	0
0	0	1	1	0	0	0	1	0
0	0	1	1	1	1	1	0	0
0	1	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	0
0	1	0	1	0	1	0	0	0
0	1	0	1	1	1	1	1	0
1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	1	0
1	0	0	1	1	1	1	0	1
1	1	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0
1	1	1	0	1	1	0	0	1
1	1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	1	0.

Minimising the functions q_1^+, q_2^+, q_3^+ and u can be done using Karnaugh maps ($q_1 = 0$ to the left, $q_1 = 1$ to the right).

q_1^+ (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	-	-	-	-
10	0	1	1	1

q_1^+ (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	0	1	0
01	-	-	-	-
11	0	1	1	1
10	0	1	1	1

q_2^+ (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	-	-	-	-
10	0	0	1	0

q_2^+ (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	0	1	0
01	-	-	-	-
11	0	0	1	0
10	0	0	1	0

q_3^+ (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	-	-	-	-
10	0	0	1	0

q_3^+ (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	1	0	1
01	-	-	-	-
11	0	0	1	0
10	0	0	1	0

u (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	-	-	-	-
10	0	0	0	0

u (q_2q_3 \ x_1x_2)	00	01	11	10
00	0	0	1	0
01	-	-	-	-
11	0	1	0	1
10	0	0	1	0

We finally arrive to the following minimal expressions:

$$
\begin{aligned}
q_1^+ &= q_2x_1 \vee q_2x_2 \vee q_1x_1x_2 \vee q_3x_1x_2 \\
q_2^+ &= x_1x_2 \\
q_3^+ &= q_2'x_1'x_2 \vee q_2'x_1x_2' \vee q_2x_1x_2 \\
u &= q_1q_3'x_1x_2 \vee q_1q_3x_1'x_2 \vee q_1q_3x_1x_2'
\end{aligned}
$$

The realisation is omitted, but it is required for full points on the exam.

Problem 3

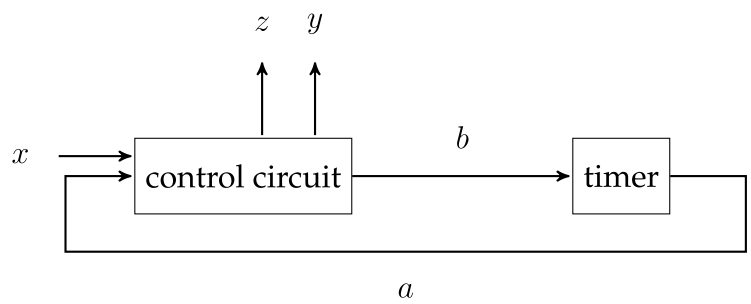
We start by looking at the overall sequence of events that will take place in the beer bottle filling machine. The conveyor belt with the bottles will move the beer bottles towards the sensor. When a bottle is in front of the sensor, we want to wait until it has passed, and then stop the conveyor belt. The bottle will then be located directly below the filling mechanism, so we open the tap, and then wait for 2.8 seconds. When the time has passed, we stop the filling mechanism, and start the conveyor belt again, and can now start over, waiting for the next bottle.

From the description above we note that there are three possible states the beer bottle filling machine can be in: waiting for a bottle to reach the sensor, waiting for the bottle to pass the sensor, and waiting for the bottle to be filled.

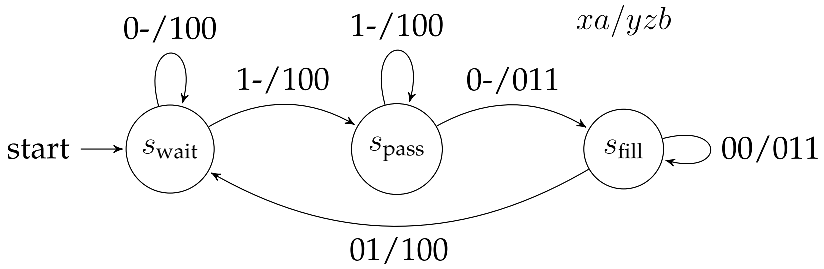
We already have the input signal x which tells us the value of the sensor. We also introduce an extra input signal a which is 0 if the timer has not yet reached 2.8 seconds, and 1 when it has reached 2.8 s. We will realize a later.

We will also have three output signals: y which controls if the conveyor belt is running, z which controls the tap, and b which controls if the timer should be running or not. b is 0 to have the timer stopped, and 1 to have it running.

The overall design can be summarized in this overview sketch.



We now start by constructing a state transition graph of the control circuit above. The control circuit has as inputs x and a, and outputs y, z, and b according to the description above. The behaviour above gives us the following state transition graph:



We note that the two output signals z and b always have the same value, and thus we only need to realize one of them. Furthermore, we see that $b = z = y'$. With the state assignment below, we get the following Karnaugh maps and expressions for the control circuit:

	s_{wait}	s_{pass}	s_{fill}
q_0q_1	00	01	10

q_0^+ (q_0q_1 \ xa)	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	-	-	-	-
10	1	0	-	-

q_1^+ (q_0q_1 \ xa)	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	-	-	-	-
10	0	0	-	-

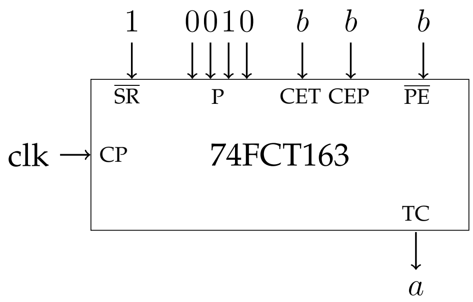
y (q_0q_1 \ xa)	00	01	11	10
00	1	1	1	1
01	0	0	1	1
11	-	-	-	-
10	0	1	-	-

Finally we can write down the minimal boolean expressions, and after this we are done with the control circuit.

$$
\begin{aligned}
q_0^+ &= q_1x' \lor q_0a' \\
q_1^+ &= x \\
y &= q_0'q_1' \lor x \lor q_0a \\
z = b &= y'
\end{aligned}
$$

We also need to construct a timer which can wait for 2.8 seconds. We use the 74FCT163 as given in the assignment. Since we have a clock with frequency 5 Hz, i.e. a period of 0.2 s, we note that waiting 14 clock pulses corresponds to exactly 2.8 s. 74FCT163 is a 4-bit counter, i.e. a modulo 16-counter. We can convert it to a module 14 counter by letting the counter go between the values 2–15, just as done in the labs. If we start at the value 2, we note that when we reach 15, the TC output signal will get the value 1 (just before the counter overflows). Thus we connect TC to the output a to signal that we have waited 2.8 s. When not counting, i.e. when b is 0, we want to load the value 2 into the counter. Thus we connect b to $\overline{\text{PE}}$, and put the value 0010 on the P-input for the parallel load. Since we only want to count when filling the bottles, we activate the counter by setting CET and CEP to the signal b from the state machine above, so that b starts the timer.

This wall of text can be summarized with the drawing below:



By combining the timer and the control circuit according to the previously shown overview picture, the assignment is complete.

Problem 4

(a) We want to express the D-transform of the sequence as $S(D) = \frac{P(D)}{C(D)}$. Since we only have the starting state, we need to calculate the coefficients of $P(D)$. We can do this for example by using (7.121) from the book:

$$\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Thus $P(D) = D \oplus D^4$. Thus, the complete sequence is:

$$S(D) = \frac{D \oplus D^4}{1 \oplus D \oplus D^5 \oplus D^6}$$

(b) To find the shortest LFSR that generates $S(D)$ we can try to simplify the previous fraction. By using Euclid's algorithm to find the greatest common divisor between the nominator and denominator, we find that $\gcd(D \oplus D^4, 1 \oplus D \oplus D^5 \oplus D^6) = 1 \oplus D$. Thus we can simplify the fraction as:

$$S(D) = \frac{D \oplus D^4}{1 \oplus D \oplus D^5 \oplus D^6} = \frac{(1 \oplus D)(D \oplus D^2 \oplus D^3)}{(1 \oplus D)(1 \oplus D^5)} = \frac{D \oplus D^2 \oplus D^3}{1 \oplus D^5}$$

Thus the connection polynomial for the shortest LFSR that generates the sequence is $1 \oplus D^5$.

(c) Using a long division of $\frac{1}{C(D)}$ we find that the period is 10.

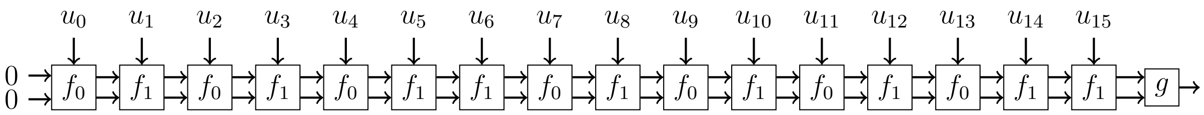
(d) The maximum period for an LFSR of length 6 is: $2^6 - 1 = 63$.

Problem 5

(a) We can solve the equation $53x = 2018$ in $\mathbb{Z}_{22187}$ as follows. We rewrite the equation as $x = 53^{-1} \cdot 2018$. We start by finding 53^{-1}. First we note that 53 and 22187 are relatively prime, since 53 is not divisible by either 11 or 2017 (both are primes). This means that $\gcd(53, 22187) = 1$ and thus the Diophantine equation $53a + 22187b = 1$ has an integer solution, and we can find an inverse to 53.
Using Euclid's algorithm, followed by Bézout's identity, we can find $a = 3349$ and $b = -8$, i.e. we have that $53 \cdot 3349 - 8 \cdot 22187 = 1$. Since we're in $\mathbb{Z}_{22187}$, this means that we can discard all multiples of 22187, and then find that $53^{-1} = 3349$.
Multiplying this inverse by 2018, we get:
$x = 53^{-1} \cdot 2018 = 3349 \cdot 2018 = 13434 \mod 22187$.

(b) Any multiple of 11 or 2017 will not have an inverse in $\mathbb{Z}_{22187}$. See Theorem 3.10 of the book.

(c) One possible solution is to do a bitwise comparison between $\mathbf{u}$ and 22187 written in binary: 0101011010101011. If we start comparing from the most significant bit, we note that as soon as one bit differs from 22187, we can immediately say that $\mathbf{u}$ is either less or greater than 22187. Thus, for each bit comparison, we have three different possibilities: the bits are equal, one is less than the other, or one is greater than the other.
We construct a network with two different kind of boxes: one box for when we expect a 1 in the binary representation of 22187. and one box for when we expect a 0. Thus we have the following circuit, where u_0 is the most significant bit.



We define that each f_i box has two inputs x and y which comes from the previous box. Each f_i box also has a third input u_i which contains a bit from the vector $\mathbf{u}$. We define the following meaning to xy: 00 means $\mathbf{u}$ *equal* to 22187, 01 means *don't care*, 10 means $\mathbf{u}$ *less than 22187*, and finally 11 to mean $\mathbf{u}$ *greater than 22187*. This gives the following truth tables for f_0, f_1, and g respectively.

f_0

x	y	u_i	a	b
0	0	0	0	0
0	0	1	1	1
0	1	0	-	-
0	1	1	-	-
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

f_1

x	y	u_i	a	b
0	0	0	1	0
0	0	1	0	0
0	1	0	-	-
0	1	1	-	-
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

g

x	y	GEQ22187
0	0	1
0	1	-
1	0	0
1	1	1

Minimising the functions f_0, f_1, and g can be done using Karnaugh maps, which gives the following minimal expressions:

$$f_{0_a} = x \vee u$$
$$f_{0_b} = y \vee x'u$$
$$f_{1_a} = u' \vee x$$
$$f_{1_b} = y$$
$$\text{GEQ22187} = g = x' \vee y$$