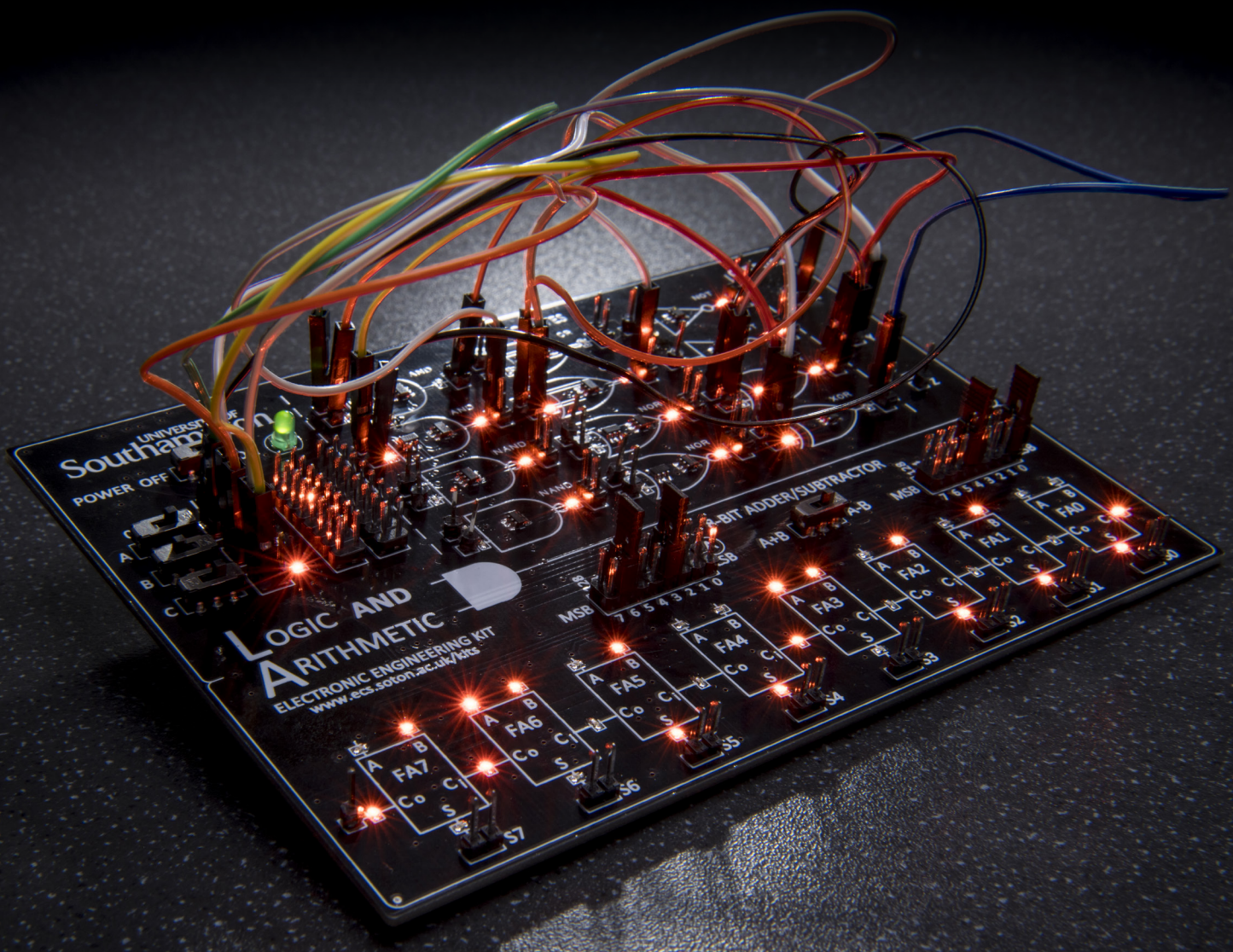
UNIVERSITY OF Southampton

UK Electronics Skills Foundation

Logic and Arithmetic Electronic Engineering Kit

LOGIC PROBLEMS



These logic problems were created by Kiran Patel, a UKESF Scholar and undergraduate student at the University of Southampton studying Electronic Engineering with Computer Systems.

About the University of Southampton

The University of Southampton is a global centre for excellence in research and education, and a founding member of the prestigious Russell Group. Southampton graduates are highly regarded by leading employers and the university works closely with industrial partners, both in teaching and research.

- We build on over 70 years' expertise across electronics and computer science to develop technologies for an ever more connected and automated world;
- Our range of electronic engineering and computer science degrees are informed by our research, and advanced theory is underpinned by hands-on application in our state-of-the-art labs;
- Through our research led teaching and exceptional industry connections, we prepare our students for future challenges not yet imagined and jobs not yet thought of.

www.ecs.soton.ac.uk

/ECSUoS @ECSUoS

About the UKESF

The purpose of the UKESF is to tackle the skills shortage in a coherent way. Our aim is to: "Encourage more young people to study Electronics and to pursue engineering careers in the sector."

To achieve the aim, we have four strategic priorities:

- Ensure more schoolchildren are aware of Electronics. Show these children, their parents and teachers that there are exciting and worthwhile careers available as designers and engineers in the Electronics sector.
- With our partners, provide opportunities for them to develop their interest in Electronics and engineering, through to university study and/or apprenticeship.
- At university, ensure that undergraduates are encouraged to pursue careers in the Electronics sector and are supported in their professional development, so they are equipped with work-ready skills and experience upon graduation.
- After graduation from university, we will help create a community of Electronic engineers to secure the future pipeline. We will build relationships and act as the representative voice for the sector on skills.

We are an independent charitable foundation, established in 2010, at the nexus of an extensive network of partners and collaborators. On behalf of the Electronics sector, we build relationships, provide thought leadership and act as the representative voice on skills-related matters. Our undergraduate Scholarship Scheme is recognised as being 'best in class'; we have supported hundreds of students since it started in 2010. In that time it is has grown considerably; we now collaborate with many of the leading universities in the UK and work with a range of companies across the whole Electronics eco-system.

Registered charity number: SC043940

www.ukesf.org

/UKESF @theUKESF

This set of Logic Problems is designed to accompany our Logic and Arithmetic Electronic Engineering Kit[1], designed specifically to assist the teaching of computer science in schools and colleges, in particular covering aspects of Boolean operations, logic gates and base 2 (binary) number systems.

The logic problems are based on real engineering examples. There are a total of 10 different problems, of increasing difficulty. The intention is not that students work through the full set of problems from start to finish, but rather that teachers will select an appropriate problem(s) for an individual class or student.

Solving the problems using the kit

The problems can all be solved using only the logic gates available on the kit, namely:

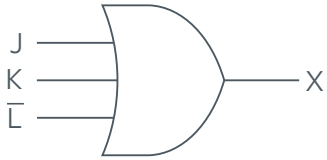
- Two 2-input AND gates
- Two 2-input OR gates
- Two 2-input NAND gates
- Two 2-input NOR gates
- Two 2-input XOR gates
- Two NOT gates (inverters)

This means that gates with more than two inputs are not available and must be constructed using a pair of 2-input gates. Inverted inputs are also not available and must be produced using NOT gates.

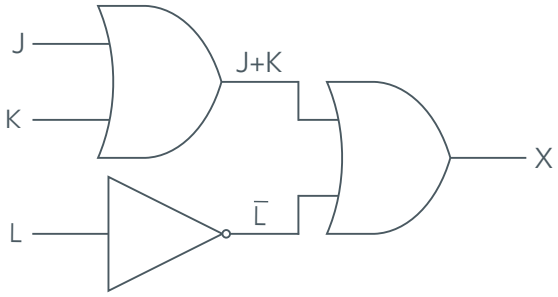
The intended solutions to all the problems are based on the above restrictions. As an example, the solution to a problem might be:

$$X = J + K + \overline{L}$$

With no restrictions, this could be accomplished using only one gate, if a 3-input OR gate and inverted inputs were available:



With the resource restrictions, three gates are required: one NOT gate to invert L and two 2-input OR gates to produce the 3-input OR gate



The solution may therefore be better represented as:

$$X = (J + K) + \overline{L}$$

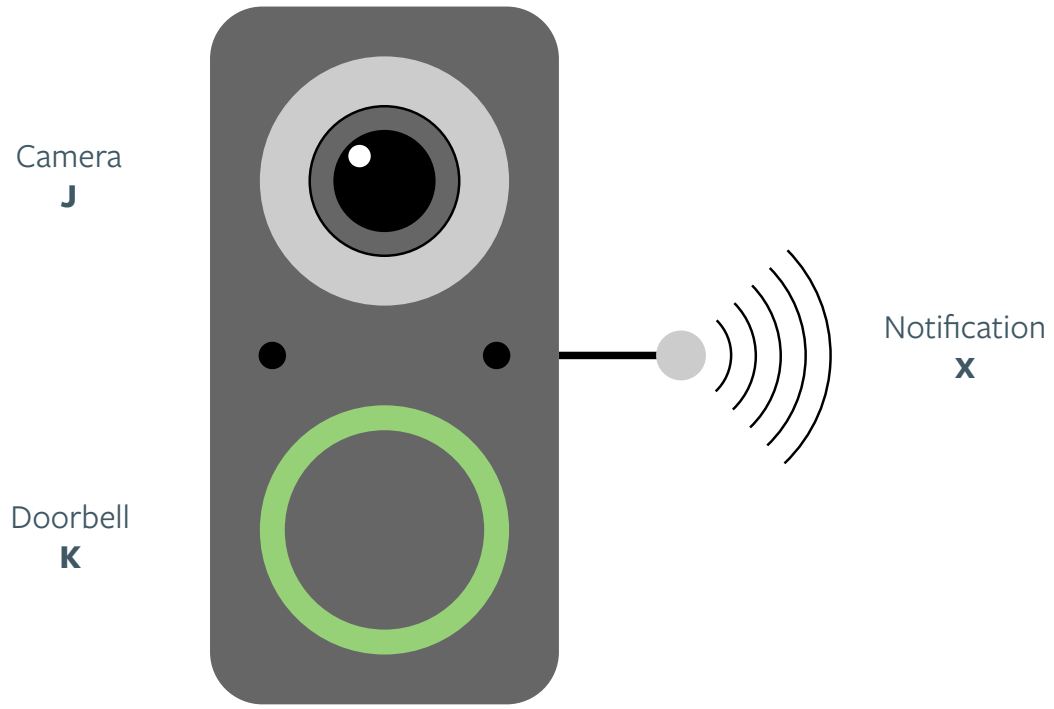
1 www.ecs.soton.ac.uk/kits

PROBLEMS

Problem 1: Smart Doorbell

Smart doorbells allow people to answer their door remotely. This means the user can talk to their visitors while they are not at home, or if they are unable to answer the door in person for any other reason, such as a disability.

Smart doorbells include a camera in addition to a doorbell. If the camera detects movement, or if the doorbell is pressed, then a notification is sent to the user's smartphone. An app can then be used to view the camera and to listen to or speak with the visitor.



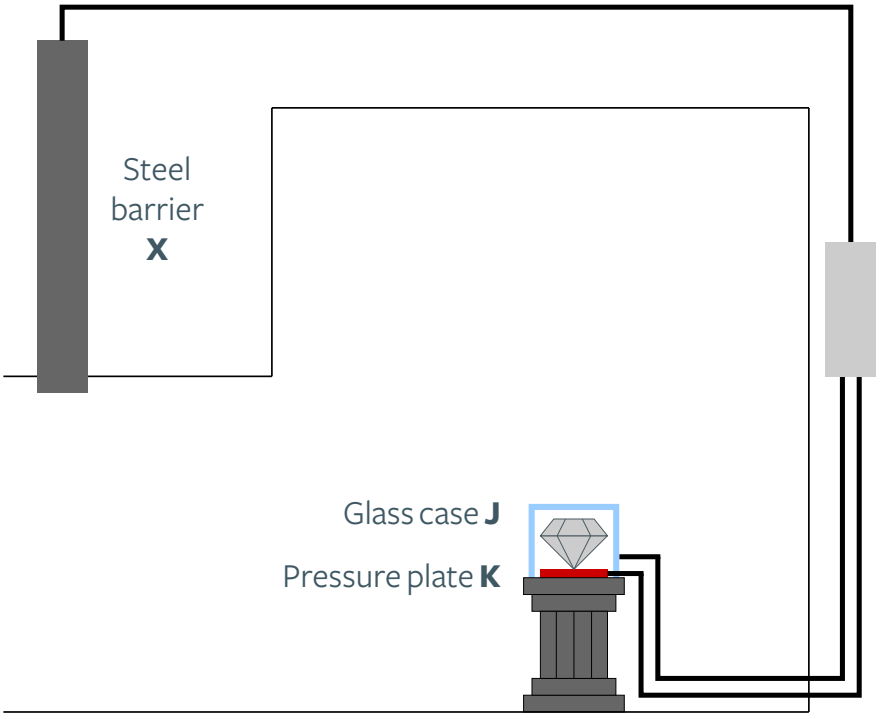
Design the logic circuit required to send the notification, modelling the camera, doorbell and notification as follows:

→ Camera: (input **J**)	0 – Movement not detected 1 – Movement detected
→ Doorbell: (input **K**)	0 – Doorbell not pressed 1 – Doorbell pressed
→ Notification: (output **X**)	0 – Notification not sent 1 – Notification sent

Problem 2: Private Collector's Trap

A private collector has received a valuable gemstone which they wish to put on display. Due to its value, the collector has proposed a trap to prevent thieves from stealing the gemstone or escaping after attempting to steal it.

The gemstone rests on top of a pressure plate on a pedestal, surrounded by a glass case. If the glass is broken and the gemstone's weight is removed from the pressure plate, the trap is set off. A steel barrier is then lowered, blocking the only entrance to the room and trapping any thieves.



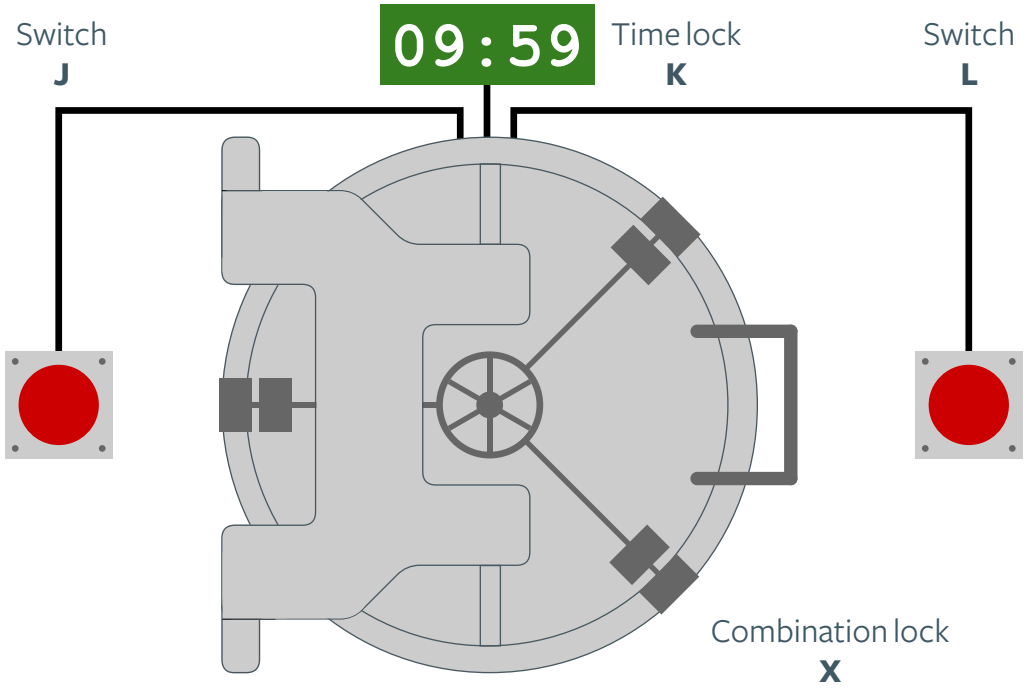
Design the logic circuit required for this trap, modelling the glass case, pressure plate and steel barrier as follows:

→ Glass case: (input **J**)	0 – Glass not broken 1 – Glass broken
→ Pressure plate: (input **K**)	0 – Weight removed (gemstone missing) 1 – Weight applied (gemstone not missing)
→ Steel barrier: (output **X**)	0 – Barrier raised (entrance open) 1 – Barrier lowered (entrance blocked)

Problem 3: Bank Vault Lock

Bank vaults use many different types of locks to prevent theft and unauthorised access. One bank improves its security by controlling the combination lock with an electronic system, allowing the combination lock to be enabled and disabled electronically. While disabled, the vault will not open even if the correct combination is used.

The combination lock is enabled when two switches, located in separate rooms, are pressed. Additionally, to prevent access outside of the bank's opening hours even if the combination has been stolen, a time lock is used. This means that the combination lock is only enabled while the time lock is off as well.



Design the logic circuit required to turn the combination lock on and off, modelling the switches, time lock and combination lock as follows:

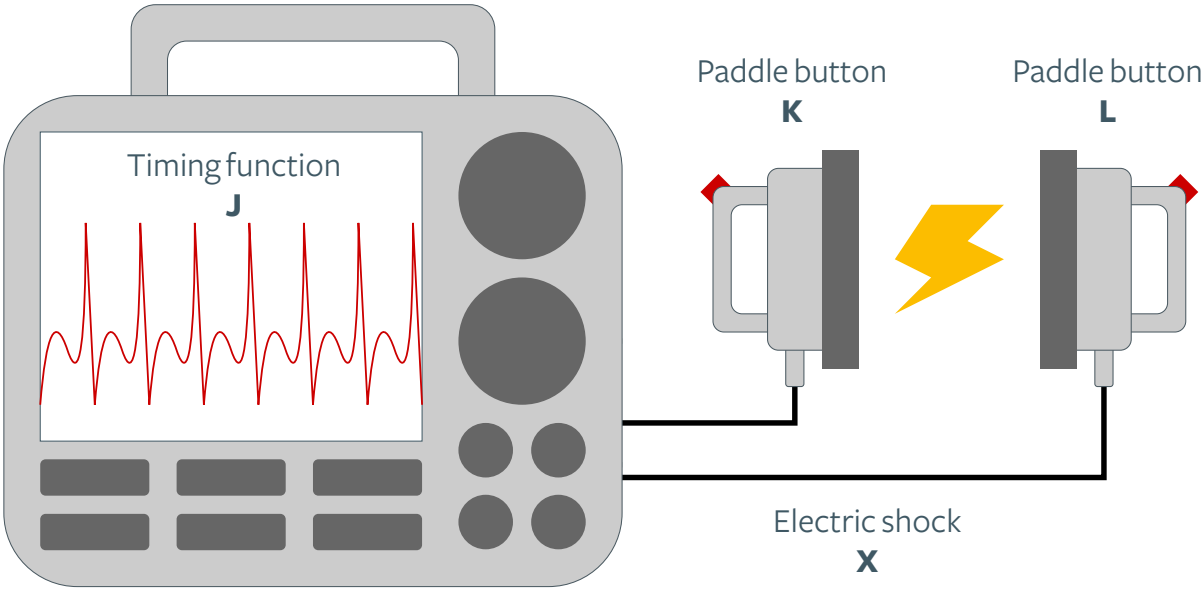
- Switches: (inputs **J** and **L**)
  - 0 – Switch not pressed
  - 1 – Switch pressed
- Time lock: (input **K**)
  - 0 – Time lock off
  - 1 – Time lock on
- Combination lock: (output **X**)
  - 0 – Combination lock disabled
  - 1 – Combination lock enabled

Problem 4: Synchronised Defibrillator

University of Southampton

Defibrillators are life-saving devices which apply an electric shock to a patient to return their heartbeat to normal. In some cases, this shock must be applied at a specific time (i.e. the shock must be synchronised with the patient's heartbeat).

When this is the case, the paddles are placed on the patient's chest and their heartbeat is monitored. Once a timing function has determined the correct moment to apply the shock, the user pushes a button on each paddle. The shock is then automatically applied at the appropriate time to correct the patient's heartbeat.



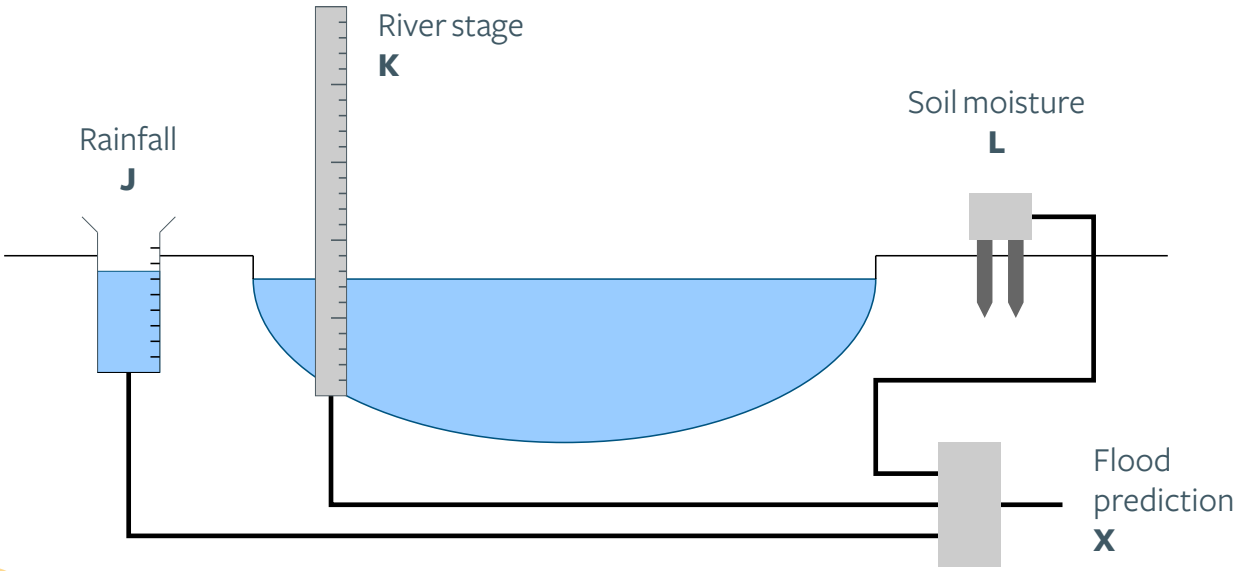
Design the logic circuit required for this defibrillator, modelling the timing function, paddle buttons and electric shock as follows:

→ Timing function: (input **J**)	0 – Correct time for shock not determined 1 – Correct time for shock determined
→ Paddle buttons: (inputs **K** and **L**)	0 – Button pressed 1 – Button not pressed
→ Electric shock: (output **X**)	0 – Shock not applied 1 – Shock applied

Problem 5: Flood Predictor

Flooding can be predicted ahead of time, allowing those who may be affected to set up defences or to evacuate. A simple prediction uses the amount of rainfall, as well as other conditions such as river stage (the water level in the river) and soil moisture, to determine whether a flood is likely to occur.

If the river stage is high, it may be close to overflowing, while if the soil moisture is high, it is unable to hold more water. High rainfall at the same time as either of these would then cause flooding. A prediction should therefore be made in these conditions.



Design the logic circuit required for this prediction system, modelling the rainfall, river stage, soil moisture and flood prediction as follows:

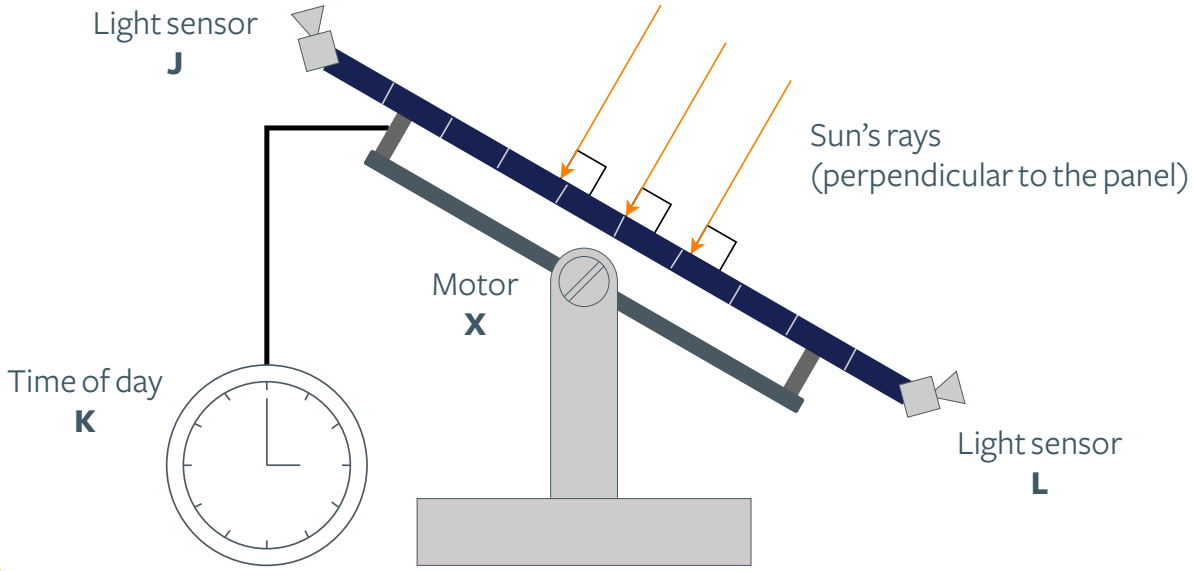
→ Rainfall: (input **J**)	0 – Low rainfall 1 – High rainfall
→ River stage: (input **K**)	0 – Low river stage 1 – High river stage
→ Soil moisture: (input **L**)	0 – Low soil moisture 1 – High soil moisture
→ Flood prediction: (output **X**)	0 – Flooding not predicted 1 – Flooding predicted

Problem 6: Active Solar Tracker

Solar panels generate electricity by absorbing sunlight. The amount of electricity generated depends on the angle of the Sun's rays. To maximise the amount of electricity generated, the panel should be perpendicular to the Sun's rays. Therefore, some panels are equipped with two light sensors and a motor. These keep the panel perpendicular to the Sun's rays as the Sun moves across the sky.

The two sensors produce an output based on how much light they receive: logic 1 in bright light and logic 0 at other times. When the Sun's rays are perpendicular to the panel, the two sensors produce the same output and the motor is off. The values of the outputs do not matter as long as they are the same.

When the Sun's rays are not perpendicular, the sensors produce differing outputs. In this case, the motor should move the panel until it is perpendicular to the Sun's rays. However, solar panels do not generate electricity at night, so the motor should also only be used during the day.



Design the logic circuit required for this tracking mechanism, modelling the light sensors, time of day and motor as follows:

→ Light sensors: (inputs **J** and **L**)	0 – Dim or no light received 1 – Bright light received
→ Time of day: (input **K**)	0 – Night time 1 – Daytime
→ Motor: (output **X**)	0 – Motor off 1 – Motor on

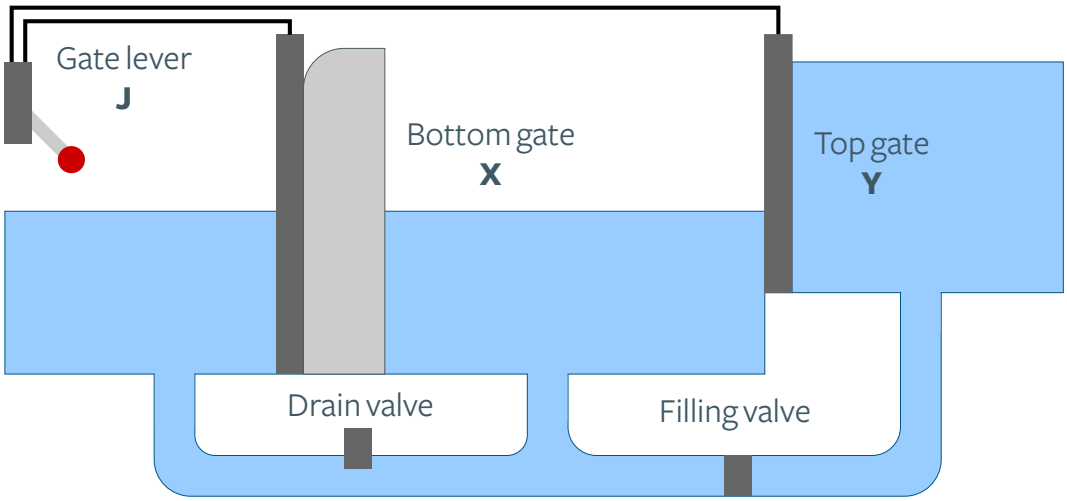
Problem 7: Pound Lock Controller

A pound lock is used to raise and lower boats on rivers and canals. They consist of a pair of gates creating a chamber (lock) in which the water level is varied using valves.

To go upstream, the boat enters the drained lock through the bottom gate and the gate is closed behind it. The lock is then filled with water by opening the filling valve. Once full, the top gate is opened and the boat leaves.

To go downstream, the boat enters the filled lock through the top gate and the gate is closed behind it. The lock is then drained by opening the drain valve. Once drained, the bottom gate is opened and the boat leaves.

In both cases, the gates are controlled manually by a single lever. When the lever is not pulled, both gates should be closed. When pulled, it must open only the correct gate for the current water level. Hence, the top gate must only open if the lock is full, and the bottom gate must only open if the lock is drained.



Design the logic circuit required for the lock controller, modelling the gate lever, valve control and gates as follows:

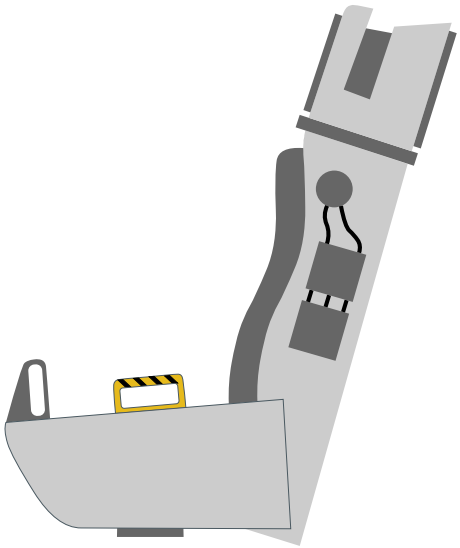
→ Gate lever: (input **J**)	0 – Closes both gates 1 – Opens one gate depending on water level
→ Valve control: (input **K**)	0 – Lock drained (drain valve open) 1 – Lock filled (filler valve open)
→ Gates: (outputs **X** (bottom) and **Y** (top))	0 – Gate closed 1 – Gate opened

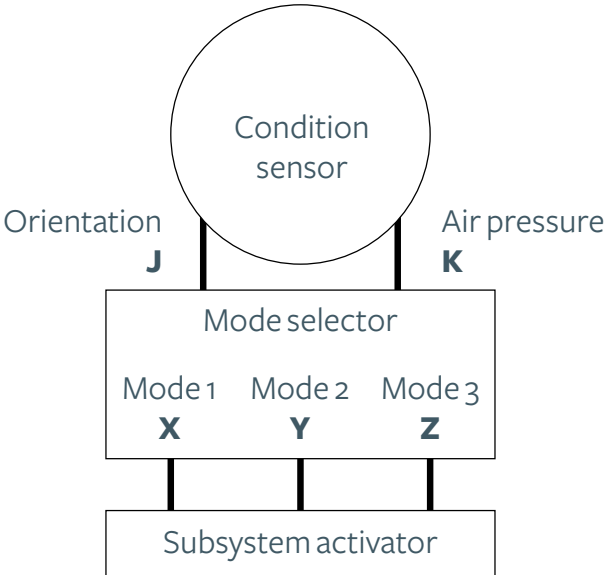
Problem 8: Ejector Seat Mode Selector

Ejector seats are designed to rescue the crew of an aircraft during an emergency. To reduce the risk of injury to the occupants during ejection, several ejection modes are available depending on the environmental conditions. For example, one ejector seat selects from three modes based on the orientation of the aircraft and the air pressure:

- Mode 1 – Used while the aircraft is upright and air pressure is high
- Mode 2 – Used while the aircraft is upright and air pressure is low
- Mode 3 – Used while the aircraft is upside-down regardless of air pressure

A condition sensor measures the orientation and air pressure and sends this information to a mode selector. The mode selector then selects the appropriate mode and activates subsystems, such as rockets and parachutes, for that mode.





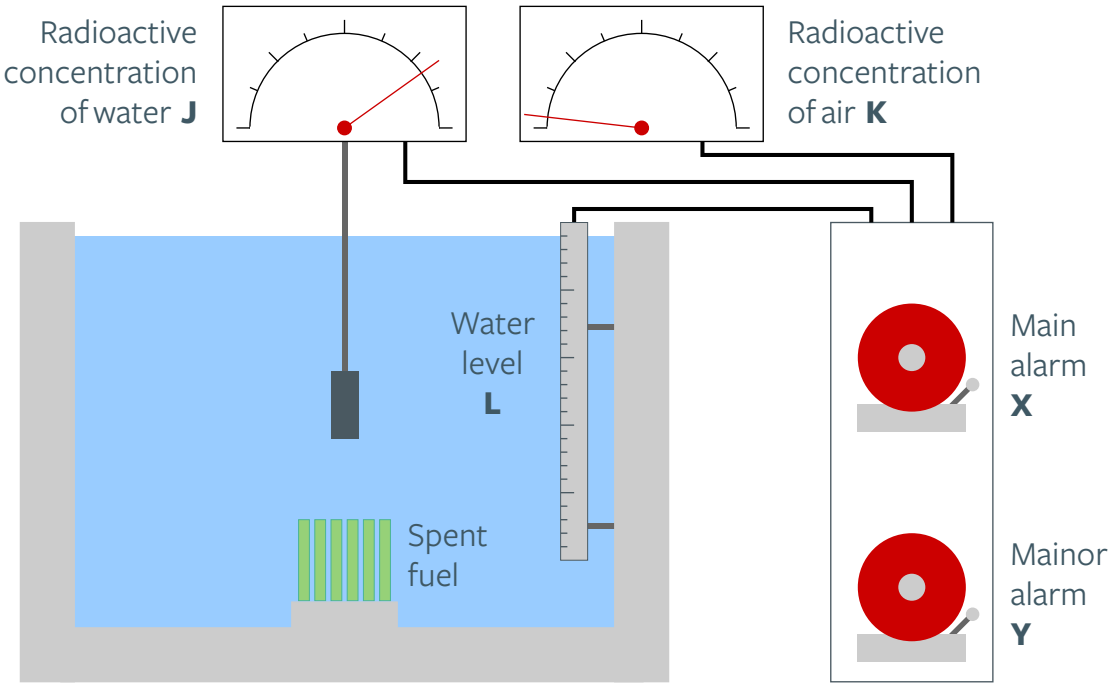
Design the logic circuit required to select the mode, modelling the orientation, air pressure and mode selector as follows:

→ Orientation: (input **J**)	0 – Aircraft upright 1 – Aircraft upside-down
→ Air pressure: (input **K**)	0 – High air pressure 1 – Low air pressure
→ Mode selection: (outputs **X**, **Y** and **Z**)	**X** at logic 1 – Mode 1 **Y** at logic 1 – Mode 2 **Z** at logic 1 – Mode 3 Other outputs in each mode should be logic 0.

Problem 9: Spent Fuel Pool Alarms

Spent fuel from nuclear reactors remains dangerously radioactive for long periods of time. This radiation must be contained. Spent fuel pools are often used to store the spent fuel underwater, because water cools the fuel down and blocks the radiation. Several conditions must be monitored and controlled to ensure that radioactive materials are not released into the environment.

One facility monitors the radioactive concentrations of both the water in the pool and the air surrounding the pool. If both concentrations differ from normal levels, then the main alarm turns on, indicating a major problem. The water level of the pool is also monitored. If the water level drops below a certain threshold, or if only one of the concentrations differ from normal levels, then the minor alarm turns on.



Design the logic circuit required to turn the alarms on and off, modelling the radioactive concentrations, water level and alarms as follows:

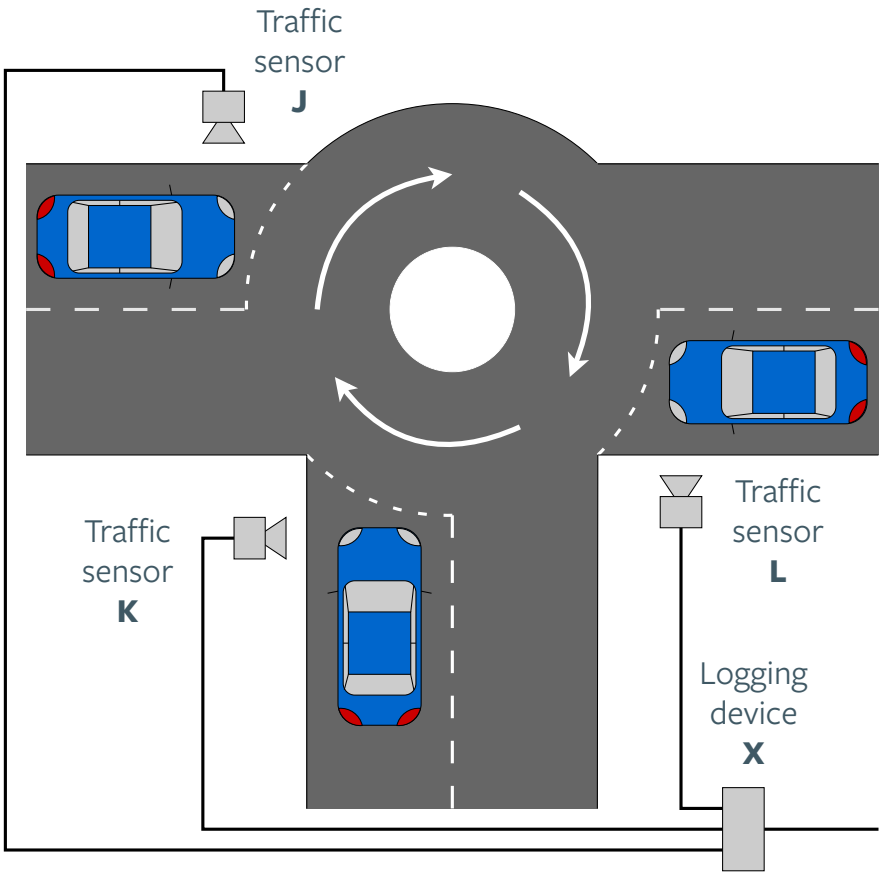
- Radioactive concentrations: (inputs **J** (water) and **K** (air))
  - 0 – Same as normal level
  - 1 – Differs from normal level
- Water level: (input **L**)
  - 0 – Above or at safe threshold
  - 1 – Below safe threshold
- Alarms: (outputs **X** (main) and **Y** (minor))
  - 0 – Alarm on
  - 1 – Alarm off

Problem 10: Roundabout Congestion Detector

University of Southampton

A local council has received numerous reports of congestion occurring at a roundabout. In order to confirm these reports and determine the severity of the issue, a monitoring system has been proposed to analyse traffic coming from the three entrances to the roundabout.

At each entrance, a sensor has been set up to detect whether traffic is waiting at that entrance. A logging device is then activated when at least two of these sensors have been triggered, allowing the council to evaluate how often and how long traffic is waiting, and whether any improvements to the roads are required.



Design the logic circuit required for this congestion detection system, modelling the sensors and logging device as follows:

- Sensors: (inputs **J**, **K** and **L**)
  - 0 – Entrance clear (no traffic waiting)
  - 1 – Traffic waiting
- Logging device: (output **X**)
  - 0 – Inactive (no congestion detected)
  - 1 – Logging (congestion detected)

SOLUTIONS

Problem 1 (Solution): Smart Doorbell

University of Southampton

1 **Create a truth table:**

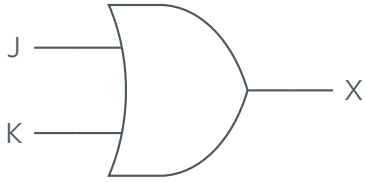
J	K	X
0	0	0
0	1	1
1	0	1
1	1	1

2 **Determine that an OR gate is required:**

$X = J + K$

Since the solution is one gate, it can be reached by matching the truth table with that of an OR gate, or by recognising the wording in the problem ("if the camera detects movement OR if the doorbell is pressed").

3 **Create a logic circuit:**



Problem 2 (Solution): Private Collector's Trap

University of Southampton

1 **Create a truth table:**

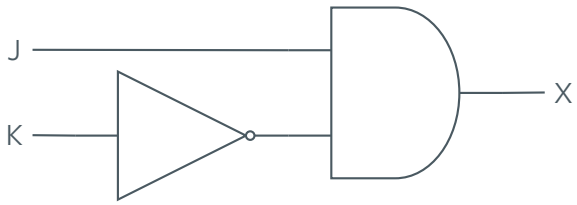
J	K	X
0	0	0
0	1	0
1	0	1
1	1	0

2 **Derive the logic equation:**

$X = J \cdot \overline{K}$

This equation is reached by reading all the combinations of inputs that result in X = 1 from the truth table. There is only one combination in this case, so this combination is the solution.

3 **Create a logic circuit:**



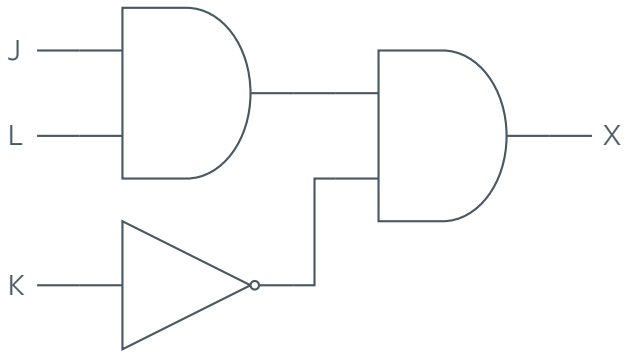
1 **Create a truth table:**

J	K	L	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

2 **Derive the logic equation:**

$X = J \cdot L \cdot \overline{K}$

3 **Create a logic circuit:**



Problem 4 (Solution): Synchronised Defibrillator

University of Southampton

1 **Create a truth table:**

J	K	L	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

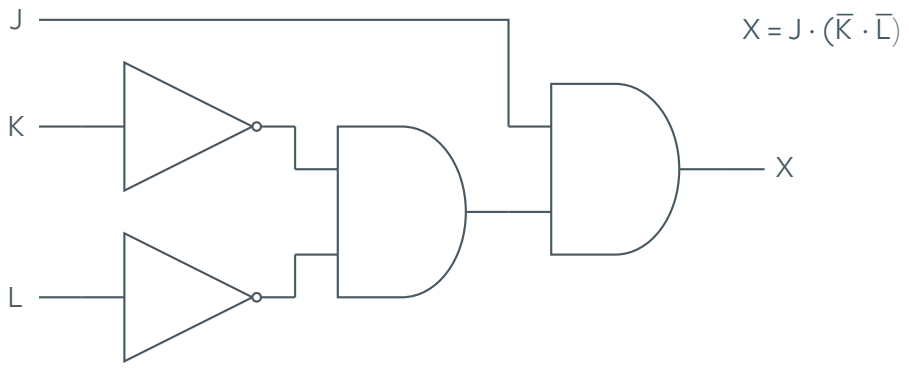
2 **Derive the logic equation:**

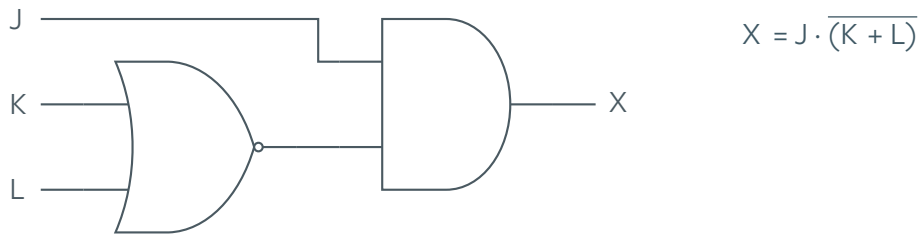
$X = J \cdot \overline{K} \cdot \overline{L}$ The first line is obtained by reading the truth table.

$= J \cdot \overline{(K + L)}$ To reach the second line, use De Morgan's Law or otherwise to identify that a NOR gate is required.

Either line is a valid solution and circuits for both are given. The second line results in an optimised circuit and should be treated as an extension.

3 **Create a logic circuit:**





1

Create a truth table:

J	K	L	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2

Derive the logic equation:

a. Read the clauses resulting in X = 1 from the truth table.

$$X = J \cdot \overline{K} \cdot L \; + \; J \cdot K \cdot \overline{L} \; + \; J \cdot K \cdot L$$

b. Factorise out J.

$$X = J \cdot (\overline{K} \cdot L \; + \; K \cdot \overline{L} \; + \; K \cdot L)$$

c. Factorise out either K or L inside the brackets.

$$X = J \cdot (\overline{K} \cdot L \; + \; K \cdot (\overline{L} \; + \; L))$$

d. $\overline{L} + L = 1$, therefore that clause can be removed ($K \cdot 1 = K$).

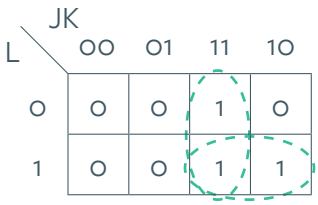
$$X = J \cdot (\overline{K} \cdot L \; + \; K)$$

e. can be removed by Boolean constraint propagation.
For the brackets:

- If K = 1, the clause in the brackets is equal to 1.
- If K = 0, the clause in the brackets is equal to L.

$$X = J \cdot (L + K)$$

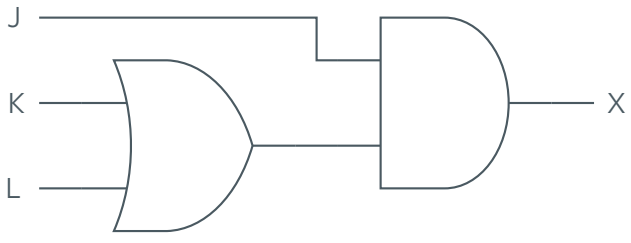
Alternatively, the solution is easily obtained from a Karnaugh map:



$$X = J \cdot K + J \cdot L$$
$$= J \cdot (L + K)$$

3

Create a logic circuit:



Problem 6 (Solution): Active Solar Tracker

University of Southampton

1 **Create a truth table:**

J	K	L	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

2 **Derive the logic equation:**

a. Read the clauses resulting in X = 1 from the truth table.

$$X = \overline{J} \cdot K \cdot L \; + \; J \cdot K \cdot \overline{L}$$

b. Factorise out K.

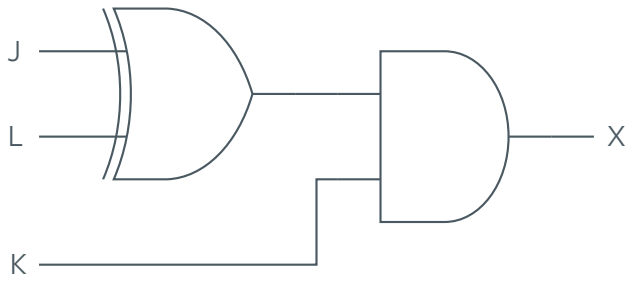
$$X = K \cdot (\overline{J} \cdot L \; + \; J \cdot \overline{L})$$

c. Simplify the brackets using

$$\overline{J} \cdot L \; + \; J \cdot \overline{L} = J \oplus L.$$

$$X = K \cdot (J \oplus L)$$

3 **Create a logic circuit:**



Problem 7 (Solution): Pound Lock Controller

University of Southampton

1 **Create a truth table:**

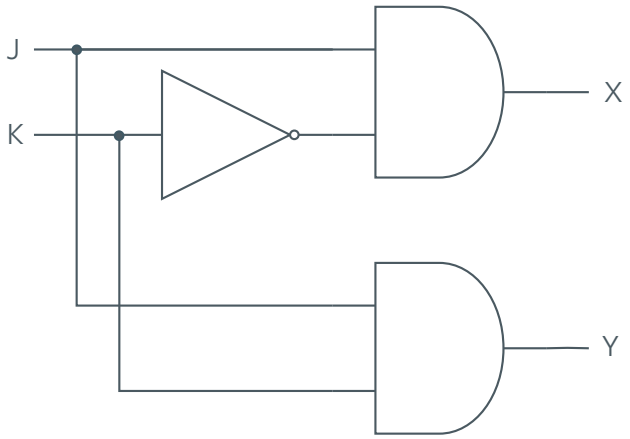
J	K	X	Y
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

2 **Derive the logic equation:**

$X = J \cdot \overline{K}$

$Y = J \cdot K$

3 **Create a logic circuit:**



Problem 8 (Solution): Ejector Seat Mode Selector

University of Southampton

1

Create a truth table:

J	K	X	Y	Z
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	0	0	1

2

Derive the logic equation:

$X = \overline{J} \cdot \overline{K}$
$= \overline{J + K}$

For X, use De Morgan's Law or otherwise to identify that a NOR gate is required.

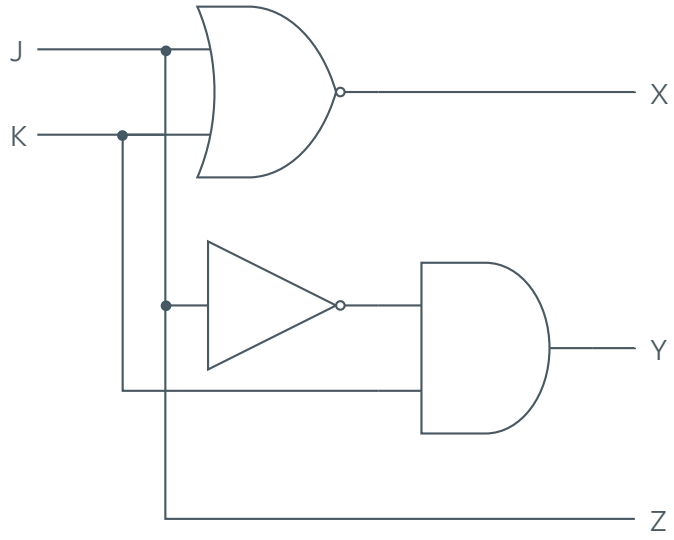
$Y = \overline{J} \cdot K$

$Z = J \cdot \overline{K} + J \cdot K$
$= J \cdot (\overline{K} + K)$
$= J$

For Z, factorise out J, then, therefore that clause can be removed. Alternatively, inspect the truth table and deduce the solution.

3

Create a logic circuit:



1

Create a truth table:

J	K	X
0	0	1
0	1	1
1	0	1
1	1	0

J	K	L	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

2

Derive the logic equation:

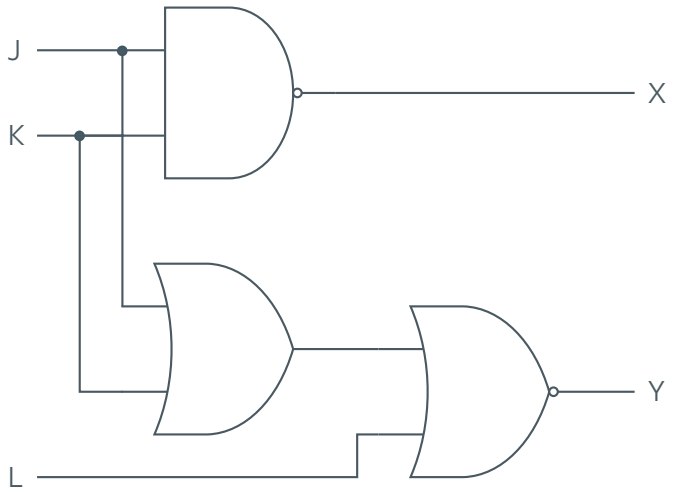
$$X = \overline{J \cdot K}$$

$$Y = \overline{J + K + L}$$
$$= \overline{(J + K) + L}$$

Y needs a 3-input NOR gate, which is not on the logic board. It can instead be created by connecting two of the inputs to an OR gate, then connecting the output of the OR gate and the third input to a NOR gate.

3

Create a logic circuit:



Problem 10 (Solution): Roundabout Congestion Detector

University of Southampton

1

Create a truth table:

J	K	L	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2

Derive the logic equation:

a. Read the clauses resulting in X = 1 from the truth table.

$$X = \overline{J} \cdot K \cdot L + J \cdot \overline{K} \cdot L + J \cdot K \cdot \overline{L} + J \cdot K \cdot L$$

b. Factorise out J · K (or any pair of inputs).

$$X = J \cdot K \cdot (\overline{L} + L) + \overline{J} \cdot K \cdot L + J \cdot \overline{K} \cdot L$$

c. $\overline{L} + L = 1$, therefore that clause can be removed.

$$X = J \cdot K + \overline{J} \cdot K \cdot L + J \cdot \overline{K} \cdot L$$

d. Factorise out L (or the input that wasn't factorised out in step b).

$$X = J \cdot K + L \cdot (\overline{J} \cdot K + J \cdot \overline{K})$$

e. Simplify the brackets using

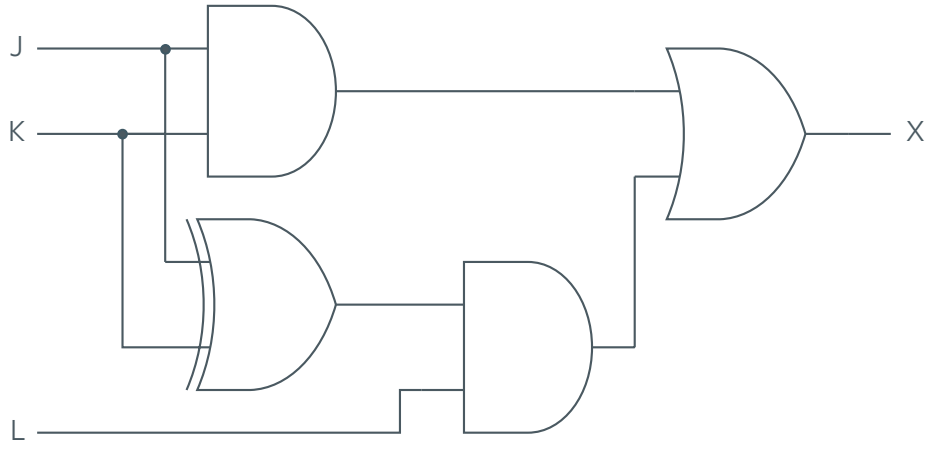
$$\overline{J} \cdot K + J \cdot \overline{K} = J \oplus K.$$

$$X = J \cdot K + L\,(J \oplus K)$$

Note that in this case, the XOR could be replaced with an OR without changing the function of the circuit.

3

Create a logic circuit:



Find out more:
www.ecs.soton.ac.uk/kits