Hardware Lab 1 : Logic gates

The aim of this lab is to demonstrate the operation of basic logic gates, their functions and how they can be combined to implement different functions i.e. the fundamental building blocks of any computer. These circuits will be constructed on the Feedback Logic Tutor LT345, as shown in figure 1 (located on the shelf under your desk). Contained on this board are a number of Transistor – Transistor logic (TTL) integrated circuits (IC), implementing a range of logical functions e.g. AND gates as shown in figure 2. Each IC is soldered into the printed circuit board (PCB), its inputs and outputs externalised using PCB mounted sockets.

Note, The power supply pins of each IC (14 & 7) are internally wired to a common supply rail (top left **BLACK** and **YELLOW** sockets).

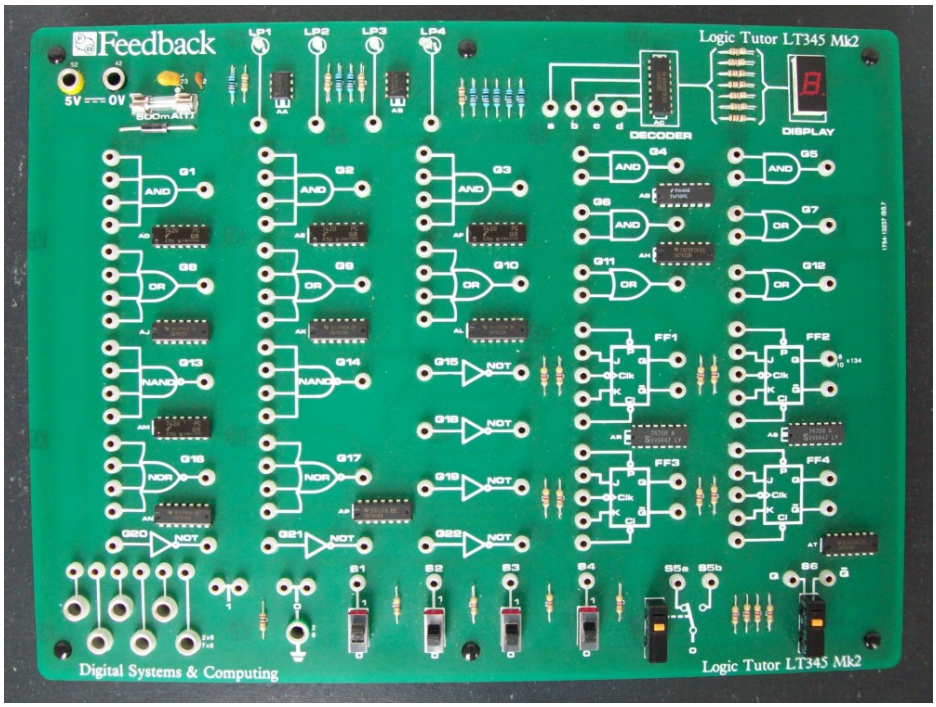


Figure 1: Feedback Logic Tutor

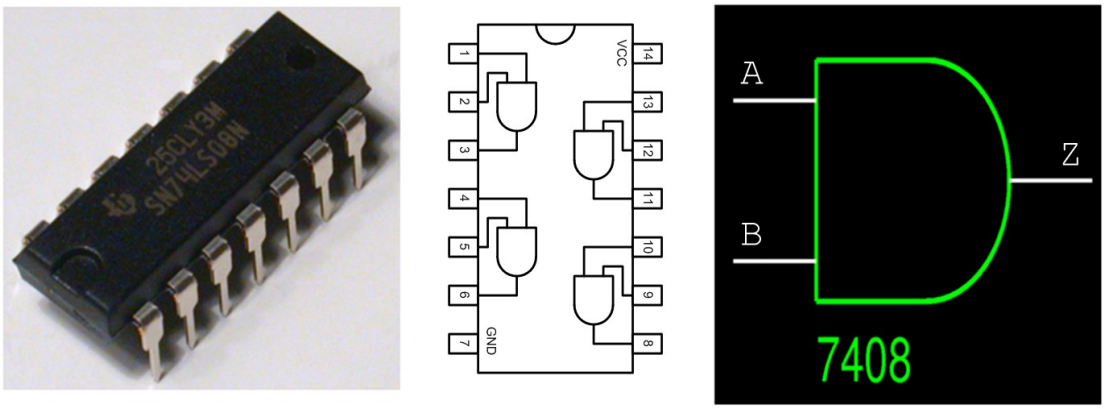
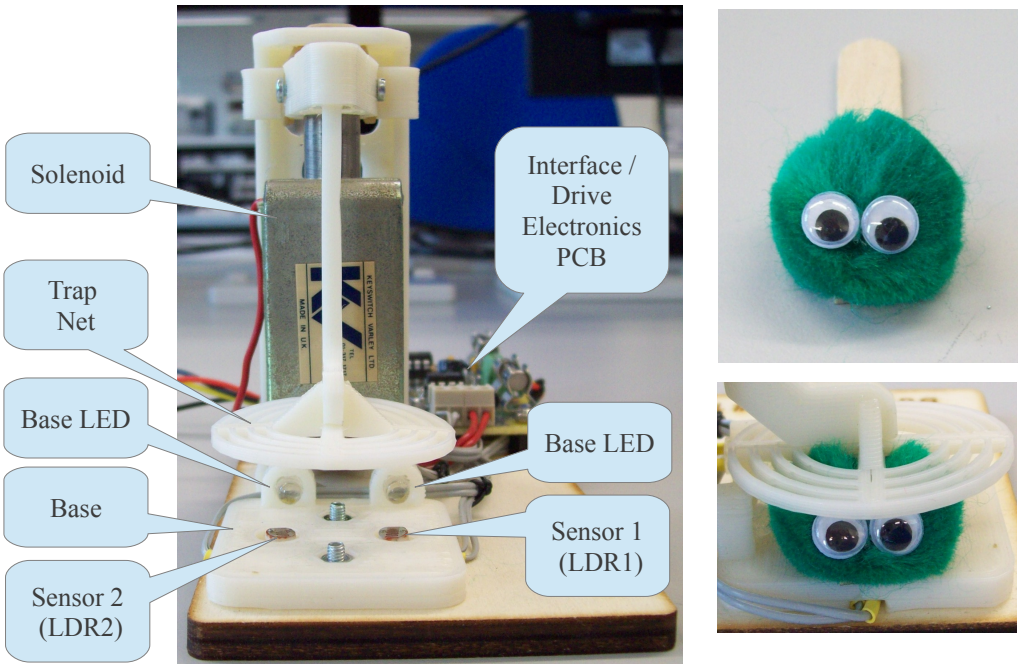


Figure 2: 7408 AND gate IC, 14 pin package and symbol



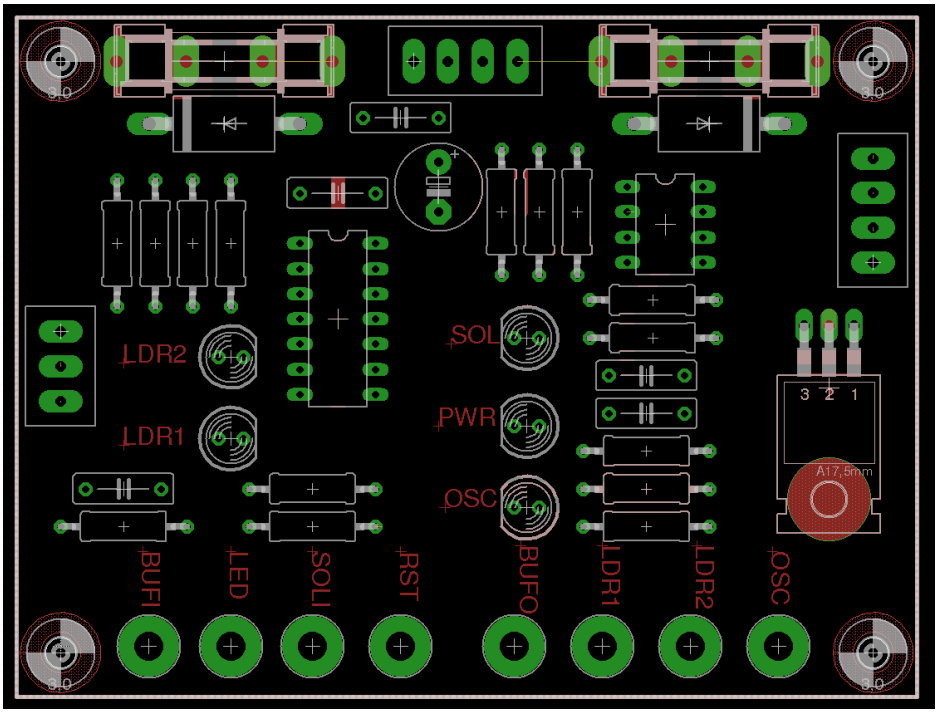


Figure 3: Cockroach trap (top left), Cockroach (top right), Interface / Driver PCB Socket / LED names (bottom)

Task 1

The University is overrun with cockroaches you are required to design an automatic cockroach trap. The required hardware has already been designed as shown in figure 3. This hardware has two main components :

1) Net: a high powered solenoid is connected to a circular net via a connecting arm. When energised the solenoid pushes the net down into the

closed position, trapping the cockroach. Otherwise a rubber band lifts the net up into the open position.

2) Base: this section contains two LEDs illuminating the bait and two Light Dependent Resistors (LDR) i.e. sensor 1 and 2, that detect when a cockroach is in the trap.

The Cockroach trap hardware needs two power supplies, a +12V and a +5V. The fixed +5V output is on the bottom right of the power supply, circled in red in figure 4.

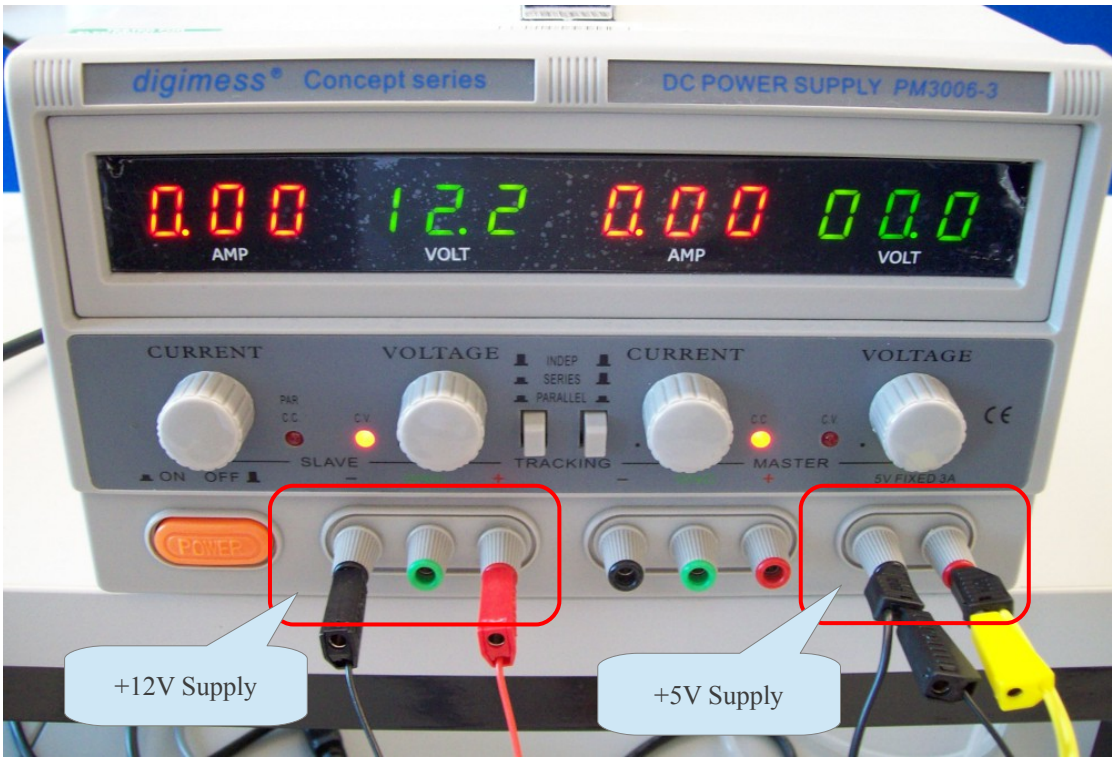


Figure 4: Bench power supply

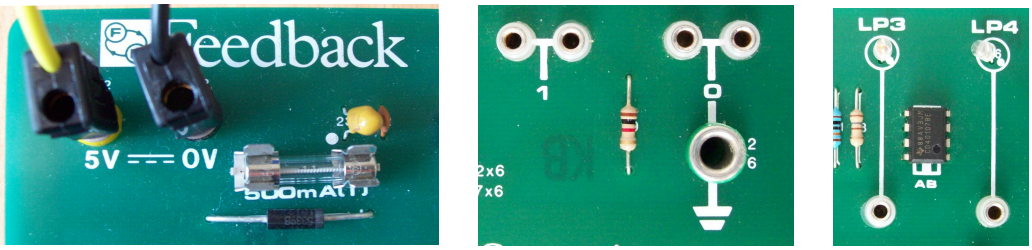


Figure 5: +5V power sockets (left), logic 1/0 (middle), LEDs (right)

The +12V supply can be obtained from one of the two variable voltage supply outputs, as shown in figure 4.

1. Set power supply mode to independent: the two middle grey square push buttons need to be pushed **OUT**, so that they match the illustration labelled INDEP.
2. Set the output to +12V : twist the VOLTAGE knob until the display reads 12.00 (+/- 0.5V). If the CC light is on, twist the CURRENT knob clockwise until it goes off.

Using the larger 4mm plug leads connect the 0V, **BLACK** terminal on the power supply to the **BLACK** socket on the Logic Tutor board. Then connect the +5V, **RED** terminal on the power supply to the **YELLOW** socket on the Logic Tutor board, as shown in figures 4 & 5.

The cockroach trap hardware has four leads, two **BLACK**, one **RED** and one **YELLOW**. Connect the **YELLOW** lead to the +5V terminal and the **RED** lead to the +12V terminal. Next connect one **BLACK** lead to the 0V terminal of the +5V supply, the other to the 0V terminal of the +12V supply (black leads are interchangeable). When complete the power supply should look the same as figure 4.

Note, in this system TRUE = +5V = Logic '1' and FALSE = 0V = Logic '0'. The +12V is used to power the solenoid (stamper).

IMPORTANT: when constructing (building) a circuit **always** ensure the power supply is turned off. Voltage spikes caused when removing or connecting wires, or accidentally connecting a wire to the wrong input or output, will damage the ICs.

The Logic Tutor board and the cockroach trap hardware are protected by fuses, therefore, don't worry you can't do too much damage :)

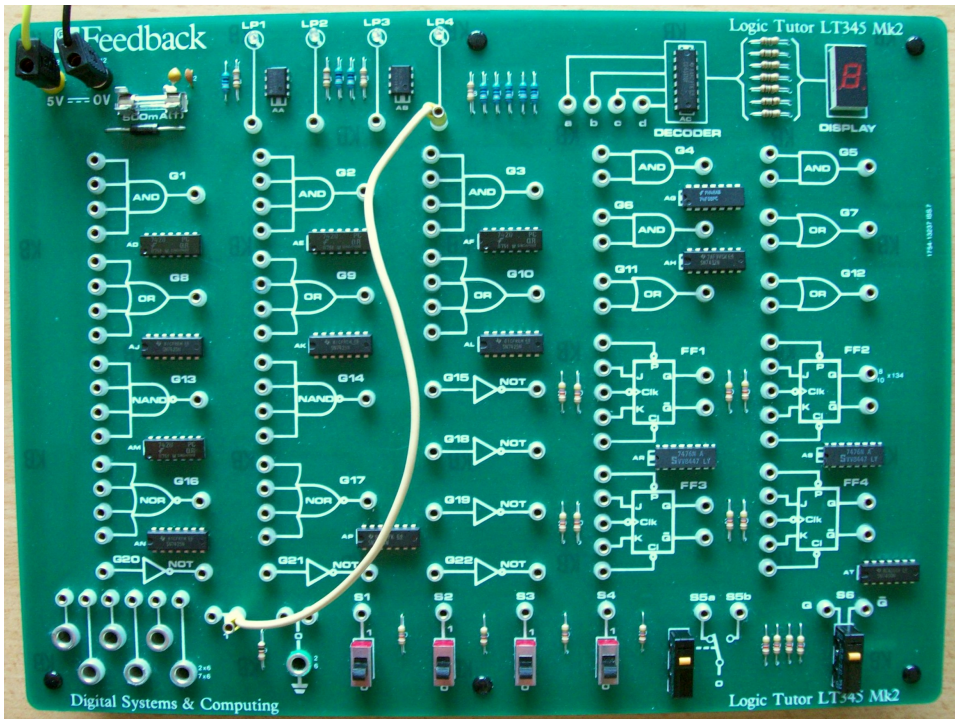


Figure 6: Initial test (top), Fixed Logic 0 / 1 (bottom left), LED (bottom right)

To test if the Logic Tutor fuse is working connect the fixed logic '1' socket to the light emitting diode (LED) `LP4` shown in figure 5, using the supplied wire links. The required sockets are shown in figure 6. Turn on the power supply by pressing the orange power button, if the fuse is working the `LP4` and `PWR` LED on the trap will be illuminated, otherwise the fuse will need to be replaced. If you are not sure how to use any items of equipment do ask for assistance.

Task 2

Using the wires provided connect the `Q` output of micro-switch `S6` (FIRE button) on the Logic Tutor board to the SOLENOID enable socket `SOLI` on the trap. Connect the same signal to the '`a`' input of the display decoder, as shown in figure 7.

Tip, the position and name of each socket on the trap are shown in figure 3.

Question: Turn on the power supply, press the red button on mico-switch `S6`, the logical state of the output will be shown on the display decoder. What logic state corresponds to the OPEN and CLOSED traps states?

Note, wires may be stacked on top of each other, or chained if they are connected to the same point e.g. in figure 7 the red and yellow wires are the same signal (FIRE).

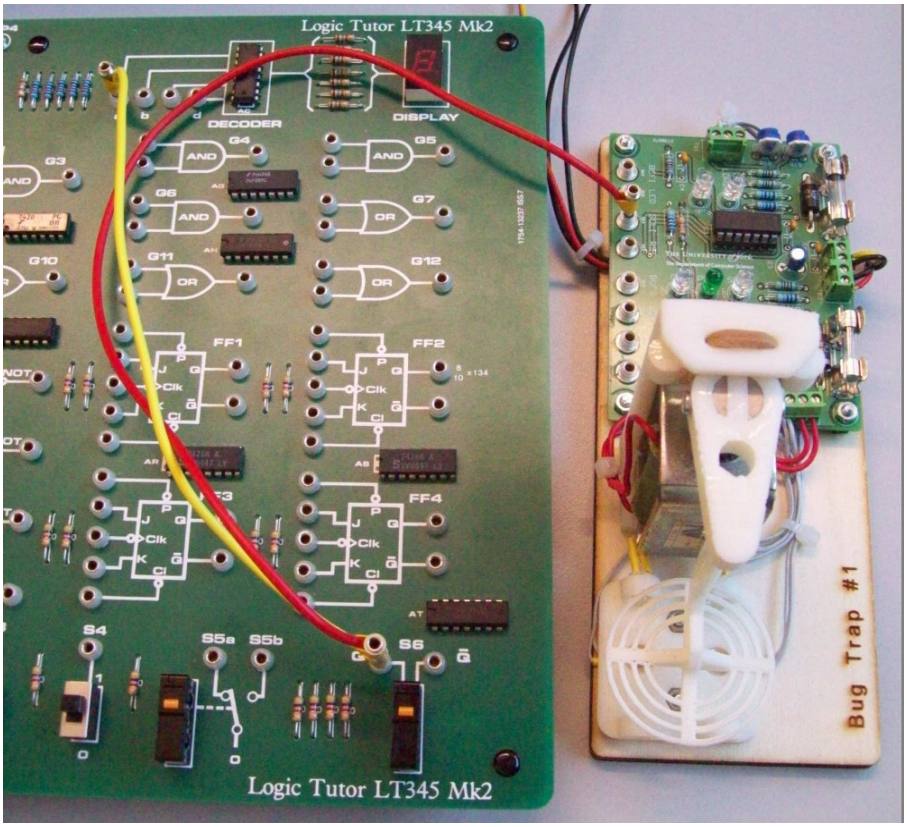


Figure 7: Solenoid test circuit

IMPORTANT: wire links are easily damaged, do not pull the wires to remove them as this will snap the solder joint connecting them to the plug. Rather hold, gently twist and lift the plug to remove it.

GOLDEN RULE: logic gates have inputs (on the left) and outputs (on the right) as shown in figure 8. When connecting these logic gates together remember these following rules.

- Input connected to an Input –> Ok, but nothing is producing a signal
- Output connected to an Input –> Good, signals processed by logic gates
- Output connected to an Output –> **VERY** bad, two outputs try to drive different signals onto the same wire.

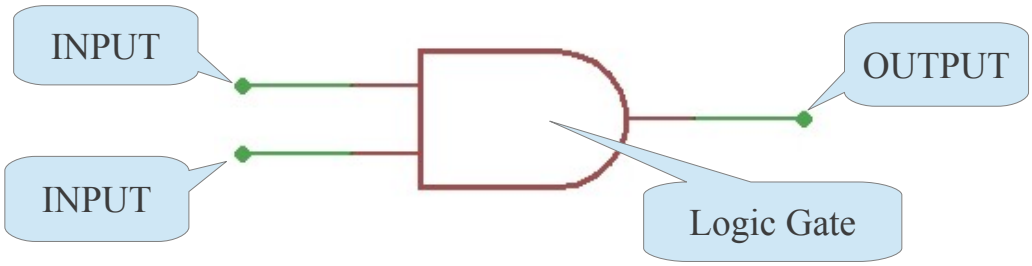


Figure 8: AND gate, inputs and outputs

Task 3

To avoid the trap from accidentally being triggered when bait is placed on the base an enable / disable trap switch is required, toggle switch `S4` (now the ENABLE button). This switch is located on the bottom right of the board, next to switch `S6` (figure 7)

Using the two input `AND` gate **`G5`** construct a logic circuit to implement the control rules shown in figure 9. These rules can also be represented graphically using a circuit diagram as shown in figure 10, or as a truth table, as shown in figure 11. **Remove** the previous wire connected to the `SOLI` input and implement this circuit.

```
LOGIC
SOLI = S6 AND S4

DESCRIPTION
IF ENABLE AND FIRE
THEN
     CLOSE TRAP
ELSE
     OPEN TRAP
```

WARNING : Do not connect the outputs of switches `S6` and `S4` to the same wire, as this will damage them. Signals must be combined using logic gates.

S6 = FIRE, S4 = ENABLE

Figure 9 : control rules – version 1

Figure 10 : circuit diagram

INPUTS		*OUTPUT*
ENABLE	FIRE	SOLENOID
0	0	0
0	1	0
1	0	0
1	1	1

Figure 11 : AND gate truth table

Question: Turn on the power supply, what combination of switch positions will cause the Solenoid to be triggered?

Task 4

To allow the trap to work at night i.e. without human assistance, the trap LEDs and LDR sensors (on base) need to be incorporated into the control rules. The first step is to add a MODE switch, toggle switch `S3`, selecting either automatic or manual control.

- S3=0 : manual mode, use switch S6 (FIRE) to control trap
- S3=1 : automatic mode, turn on base LEDs, use LDR sensors to control trap

Question: The trap base LEDs are controlled via a single input `LED`, socket shown in figure 3, connect this input to toggle switch `S3`. Turn on the power supply, what logic state causes the base LEDs to be illuminated?

Question: The trap base has two sensors `LDR1` and `LDR2`, sockets shown in figure 3. Connect these outputs to LEDs `LP1` and `LP2` on the Logic Tutor board. Turn on the power supply. Using the stick attached to the cockroach cover up each LDR in turn. What do the sensor output states '1' and '0' correspond to i.e. covered / uncovered?

Note, `LP1` and `LP2` are illuminated when a logic '1' is applied on their inputs. You should observe that when uncovered the sensor signal goes to a logic '0'. If this signal does not change, or becomes stuck in a logic '1' state you may require additional light to trigger the sensor. LDR sensor sensitivity can be adjusted using the **BLUE** pots, if you are not sure how to do this do ask for assistance.

```
LOGIC
SOLI = [(NOT S3) AND (S6 AND S4)]
       OR
       [S3 AND (LDR1 OR LDR2)]

LED = S3

DESCRIPTION
IF [MANUAL AND (ENABLE AND FIRE)]
    OR
   [AUTOMATIC AND (SENSOR1 OR SENSOR2)]
THEN
     CLOSE TRAP
ELSE
     OPEN TRAP

IF AUTOMATIC
THEN
     TURN ON BASE LEDS
ELSE
     TURN OFF BASE LEDS
```

NOTE
S3 = MODE
S4 = ENABLE
S6 = FIRE

Figure 12 : control rules – version 2

Using the above information and logic gates **`G4`**, **`G5`**, **`G6`**, **`G7`**, **`G11`** and **`G15`** **design** a

new circuit to implement the control rules shown in figure 12. Draw the circuit diagram needed to implement these rules, then construct and test this circuit.

Note, Implementation and design tips are given below. However, if you are not sure what the final logic circuit should look like its schematic is shown in Appendix A. Warning, you will be expected to understand simple logic circuits in the exam :)

Tips, break each logical expression down in a series of two input logical AND and logical OR gates. You will also need to use a NOT gate. Draw this representation out using the circuit diagram symbols shown in figure 13 to produce a wiring diagram i.e. showing what inputs (`S3`, `S4`, `S6`, `LDR1` and `LDR2`) are connected to what outputs (`SOLI` and `LED`) etc.

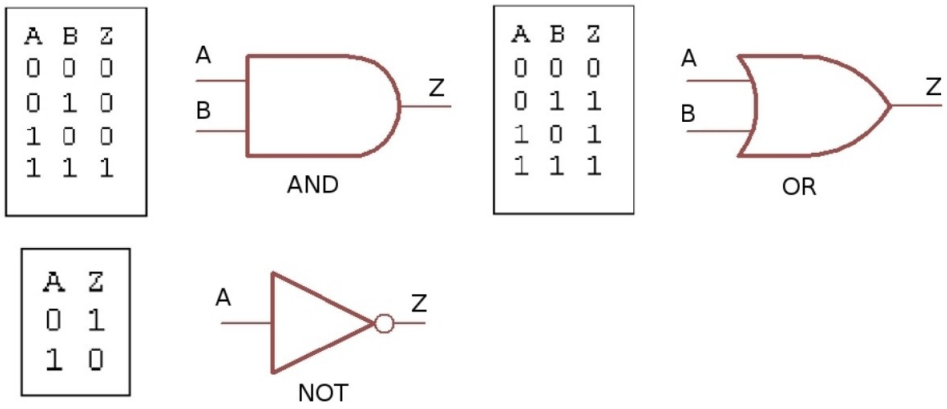


Figure 13 : Logic gate truth tables

Additional Task

The `OSC` output socket on the trap, as shown in figure 3 produces a square wave output i.e. a signal that repeatedly alternates between a logical '1' and a logical '0'. Using this output can you change the behavior of the trap so that when triggered it will repeatedly squash (stamp on) the bug instead of just holding it?

Hint, you need to add an additional `AND` gate.

Task 5

The processing speed of a logic circuit is dependent on the longest path a signal must travel from an input to an output i.e. the maximum number of gates it must pass through. This is called the critical path delay. To reduce this delay multiple logic gates can be combined into a single gate.

Question: Write out the truth tables for the circuits shown in figure 14. Which logic circuits implement a three input logical AND function? Which three input logical AND function has the shortest delay?

Tip, a logical AND gate with more than 2 inputs will only produce a logic '1' on its output when all of its inputs are a logic '1'. A logical OR gate with more than 2 inputs will only produce a logic '0' on its output when all of its inputs are a logic '0'.

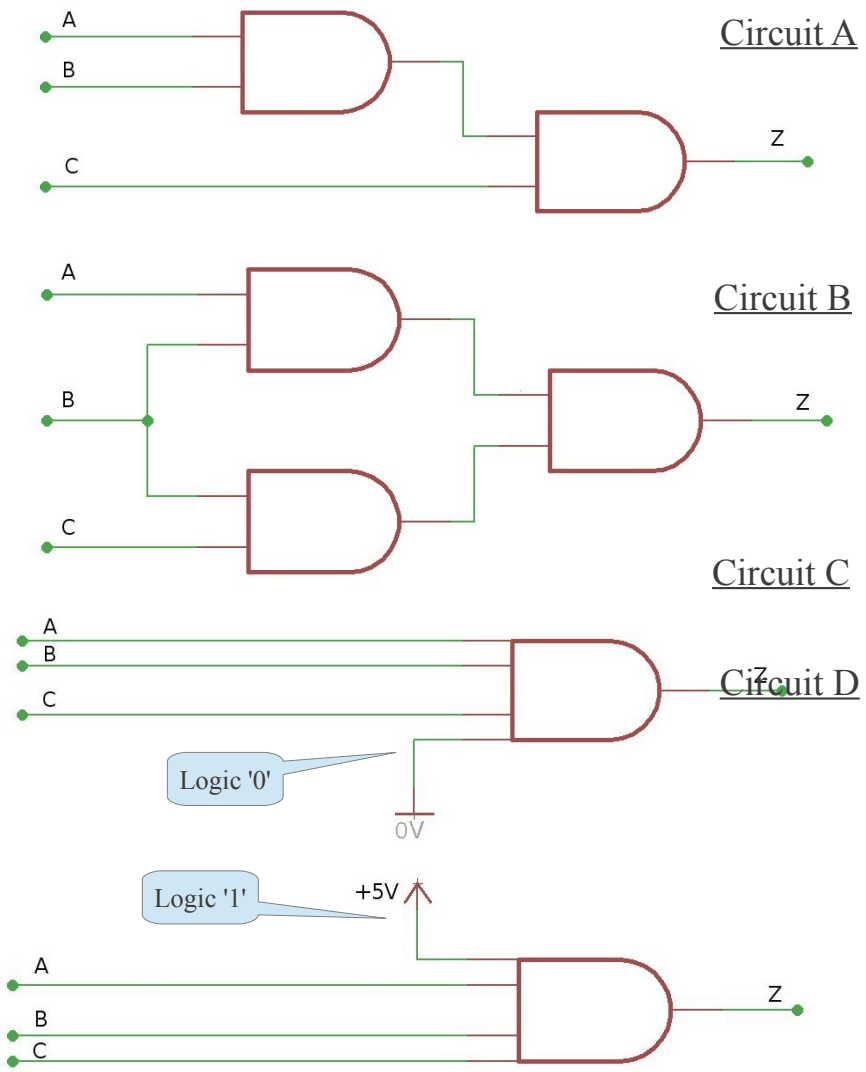


Figure 14: Logic circuits

	A	FUNC	Unused Input	Output
A AND TRUE = A	0	AND	1	0
	1	AND	1	1
A OR FALSE = A	0	OR	0	0
	1	OR	0	1

Figure 15 : Boolean Logic

Task 6

Using circuit D in figure 14 redesign your logic circuit to use logic gate **G3** i.e. a four input AND gate. Draw the circuit diagram needed to implement this circuit, then construct and test this circuit. Implementation tips are given below. However, if you are not sure what the final logic circuit should look like its schematic is shown in Appendix B.

Note, if you wish to use logic gates that have more than two inputs e.g. the four input AND gates, unused inputs must be tied to their inactive state. For an AND gate this is a logic '1', for an OR gate this is a logic '0'. This can be seen by considering the following Boolean equations and the truth tables shown in figure 15.

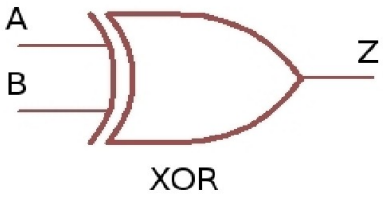
Task 7

A new breed of heavy weight cockroaches have been discovered. These are longer and more powerful than your typical bug. If caught these cockroaches can push their way through the trap i.e. they can not walk backwards. Therefore, you need to modify your circuit to implement the following additional rules:

- If the cockroach is small i.e. only covers one of the sensors, the net should hold the cockroach in the trap (to minimize the mess).
- If the cockroach is large i.e. covers both of the sensors, the net should repeatedly stamp on the bug to ensure it does not escape. You can use the `OSC` output on the trap, as shown in figure 3 to produce this behavior.

Tip: there are lots of different ways to solve this problem, one approach would be to use an XOR gate i.e. to detect that only one sensor is set. Unfortunately, the logic tutor board does not have an XOR gate, so you will need to implement it from other gates. The required truth table and Boolean equations (using AND, OR and NOT gates) is shown in figure 16.

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0




```
Z = (NOT A AND B) OR (A AND NOT B)
```

Figure 16: XOR gate

Additional Task

Can you implement the XOR gate function using only AND gates and NOT gates i.e. can you replace the OR gate using three NOT gates and an AND gate?

Summary

The logic circuit in task 6 uses one of the main building blocks found within a computer i.e. the multiplexer (MUX), an electronic switch used to route data from one point in a CPU to another. In this circuit it is used to select either the manual or automatic control logic, as shown in figure 17. To simplify circuit diagrams a multiplexer is more commonly represented using the component symbol shown in figure 18.

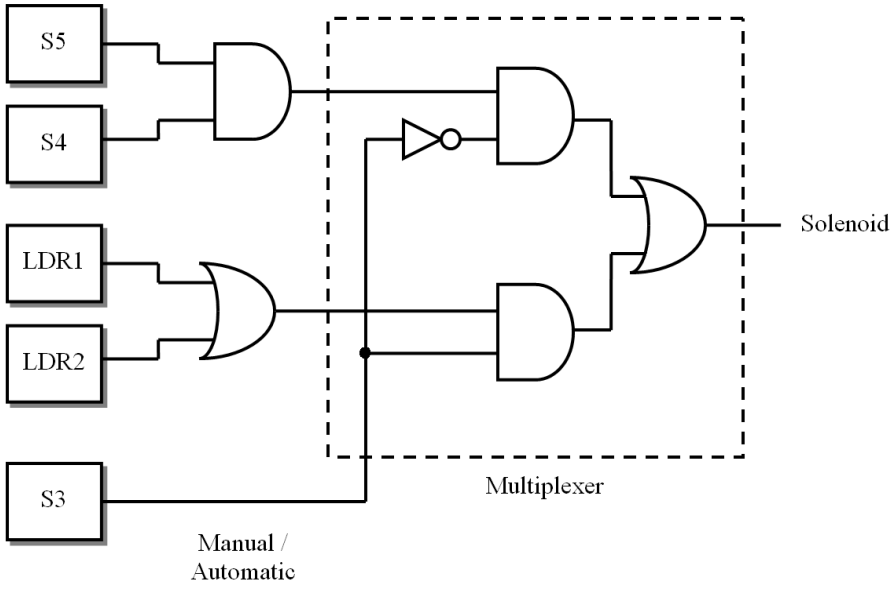


Figure 17: control logic multiplexer

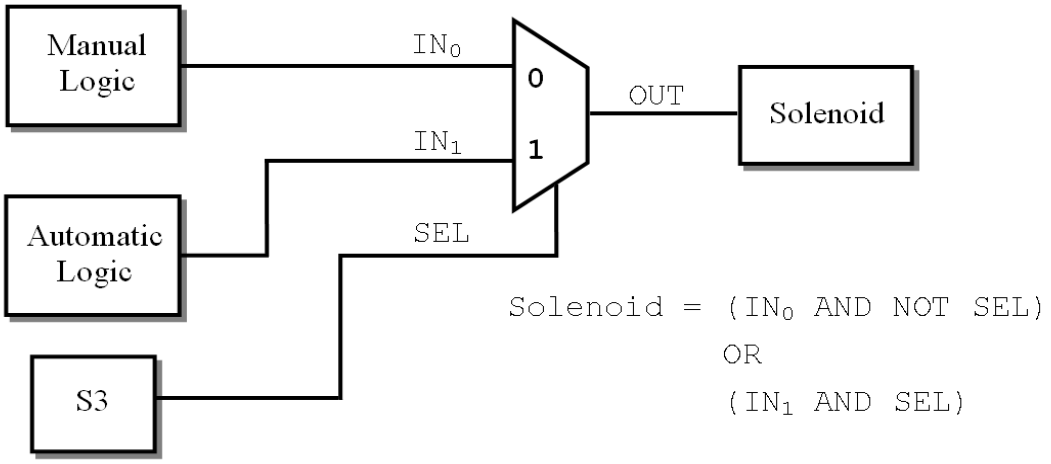


Figure 18: Multiplexer symbol

The operation of the multiplexer is controlled by the toggle switch `S3` i.e. the select (`SEL`) signal. This signal and its inverse are combined with `AND` gates, therefore, one `AND` gate will always have a logic 0 on its input allowing the `SEL` signal to mask one multiplexer input and select data from the other, as shown by these logical relationships:

```
A AND 0 = 0
A AND 1 = A
```

The final `OR` gate acts as a joining function merging the outputs of the two `AND` gates onto a single output, as only a single `AND` gate will produce a logical '1' at any time. The truth table of this two input multiplexer is shown in figure 19.

SEL	IN_0	IN_1	OUT	Description
0	0	0	0	Select IN_0
0	0	1	0	Select IN_0
0	1	0	1	Select IN_0
0	1	1	1	Select IN_0
1	0	0	0	Select IN_1
1	0	1	1	Select IN_1
1	1	0	0	Select IN_1
1	1	1	1	Select IN_1

Figure 19: Multiplexer truth table

The combination of controlling logic and multiplexers form the core of most computers i.e. hardware circuits that can work out what each instruction should do and then select the required data to be processed.

Appendix A

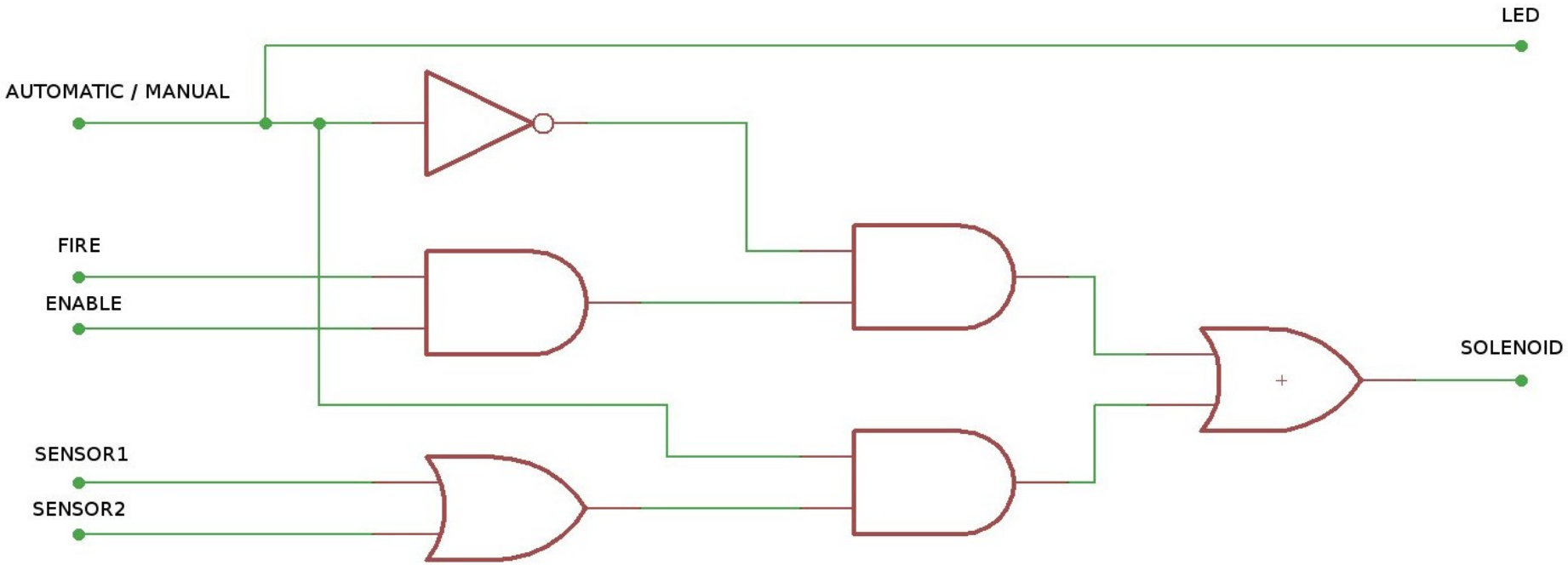


Figure A1: Task 4 circuit diagram using 2 input logic gates

Appendix B

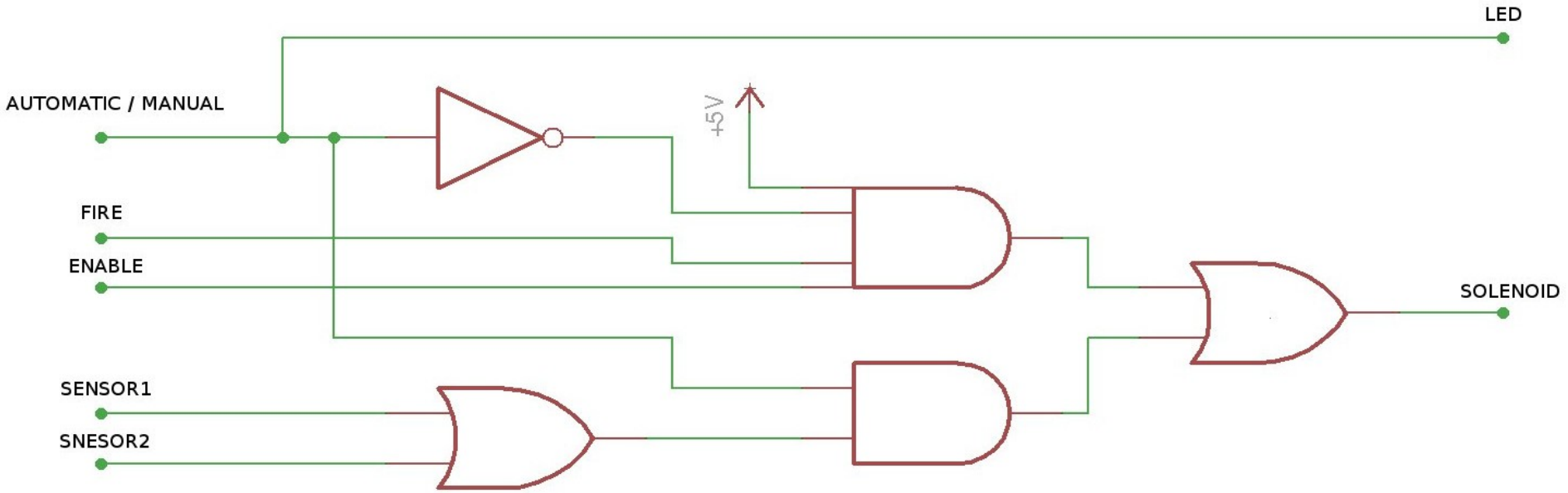


Figure B1: Task 5 circuit diagram using a 4 input logic gate