UNIVERSITY OF Southampton

UK Electronics
Skills Foundation

Logic and Arithmetic Electronic Engineering Kit

## TRAINING HANDBOOK

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

Electronic Engineering Kits

A-Level Computer Science: Logic and Arithmetic Kit

This Logic and Arithmetic kit enables students to explore digital logic and binary arithmetic. The printed circuit board (PCB) includes light emitting diodes (LEDs) to indicate the logic values throughout, and has been designed to provide a practical, hands-on tool to deliver some of the fundamental electronics parts of the A-Level Computer Science curriculum. The board can be used to explore aspects of Boolean operations, logic gates, and base-2 number systems.

Updated versions of these notes will be made available at www.ecs.soton.ac.uk/kits in the future.
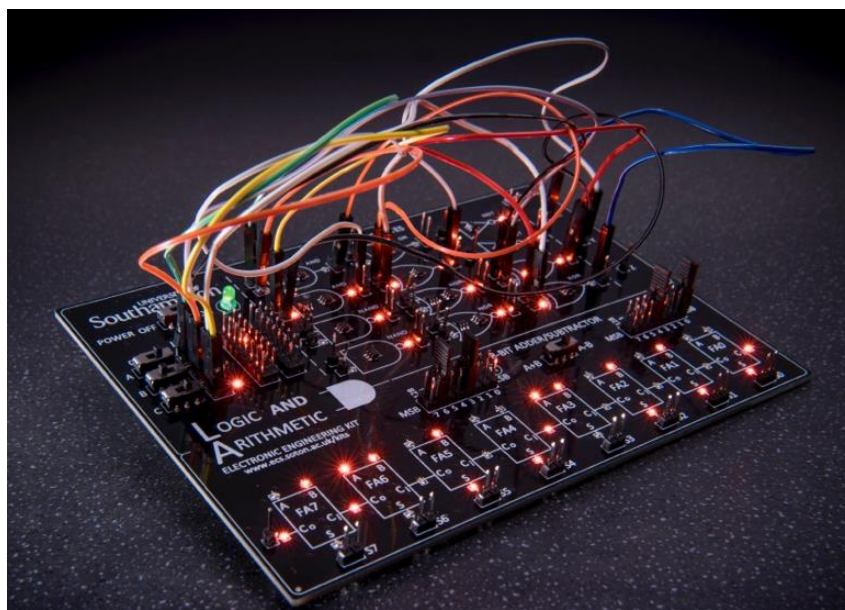
Schedule

Preparation time   :   10 minute introductory talk

Lab time   :   2-3 hours

Items provided

Tools   :   n/a

Components   :   n/a

Equipment   :   A-Level CS Logic and Arithmetic Electronic Engineering Kit

Software   :   n/a

Items to bring

n/a

You will undertake this laboratory exercise in a pair. Before starting, we recommend that you have a quick read through all sections of these notes so that you have an idea of what you are going to be doing in the lab.

Revision History

July 17, 2017	Geoff Merrett (gvm)	Version 1
July 16, 2018	Geoff Merrett (gvm)	Version 2
July 17, 2018	Geoff Merrett (gvm)	Version 2.1 (update to Q2)

1   Aims, Learning Outcomes and Outline

This laboratory exercise aims to:

- Give you an overview of the uses and capabilities of the Logic and Arithmetic kit

Having successfully completed the lab, you will be able to experimentally:

- Complete the truth tables of typical logic functions/gates
- Implement and test a simple logic function
- Understand binary addition
- Understand unsigned binary overflow
- Understand and twos complement addition/subtraction
- Implement and test a Full Adder circuit

---

2   The A-Level Computer Science Logic and Arithmetic Electronic Engineering Kit

This kit has been designed to aid the teaching of particular parts of the A-Level Computer Science syllabi, specifically:

- Boolean Operations and Logic Gates
    - Boolean operations and logic gates (AND, OR, NAND, NOR, XOR, NOT)
    - Logic gate symbols and circuit diagrams
    - Truth tables
    - Boolean expressions and equivalent logic gate circuits
    - The logic associated with half and full adders
- Base 2 number systems
    - Unsigned and signed (two's complement) binary
    - Addition of two unsigned binary integers
    - Addition and subtraction of two signed (two's complement) binary integers
    - Overflow and underflow

This section will walk you through the various areas of the kit, before you use them in the experiments presented in Section 3.

Before starting, ensure that a single AA battery has been inserted into the holder on the rear of the kit. After checking this, turn on the kit using the 'POWER' switch at the top left of the kit, see Figure 1.When this is positioned to the right ('ON'), the kit will be turned on and the orange power LED (Light Emitting Diode) dimly lit. Some orange LEDs will also be lit on the kit when the power is on; if they are not, there is either no battery, a dead battery, a short circuit, or a fault with the kit. When it is positioned to the left ('OFF'), the kit is completely turned off.
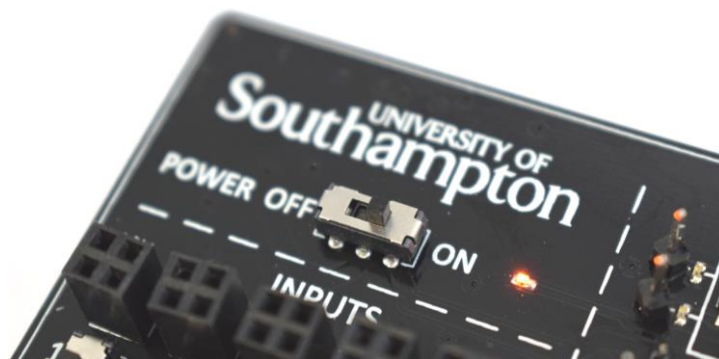
FIGURE 1: Power switch

Across the kits, orange LEDs are used to indicate the logic value of inputs and outputs, i.e. either logic '1' (orange LED lit) or logic '0' (orange LED not lit). Connections can be made to inputs and outputs using the wires provided, by connecting the plastic socket/metal pin onto the plastic socket/metal pin (shown in Figure 2).
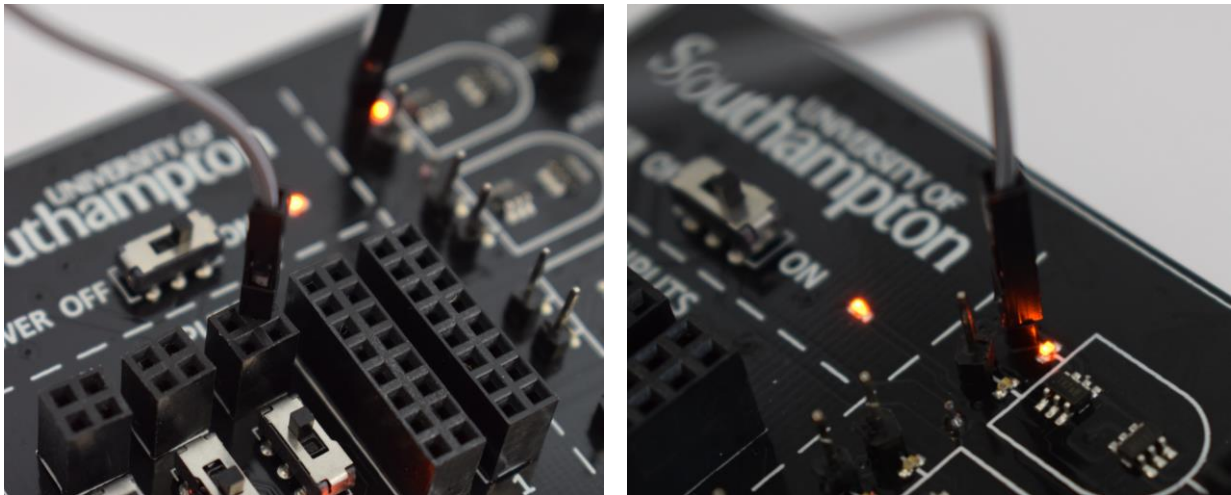
FIGURE 2: Connecting wires to inputs (right) and outputs (left)

IMPORTANT: We have designed the kit to try and prevent students 'short-circuiting' the board, and have deliberately provided wires with a socket at one end and a pin at the other.

The kit consists of two areas (identified in Figure 3):

- 2.1: LOGIC (upper) area;
- 2.2: ARITHMETIC (lower) area.
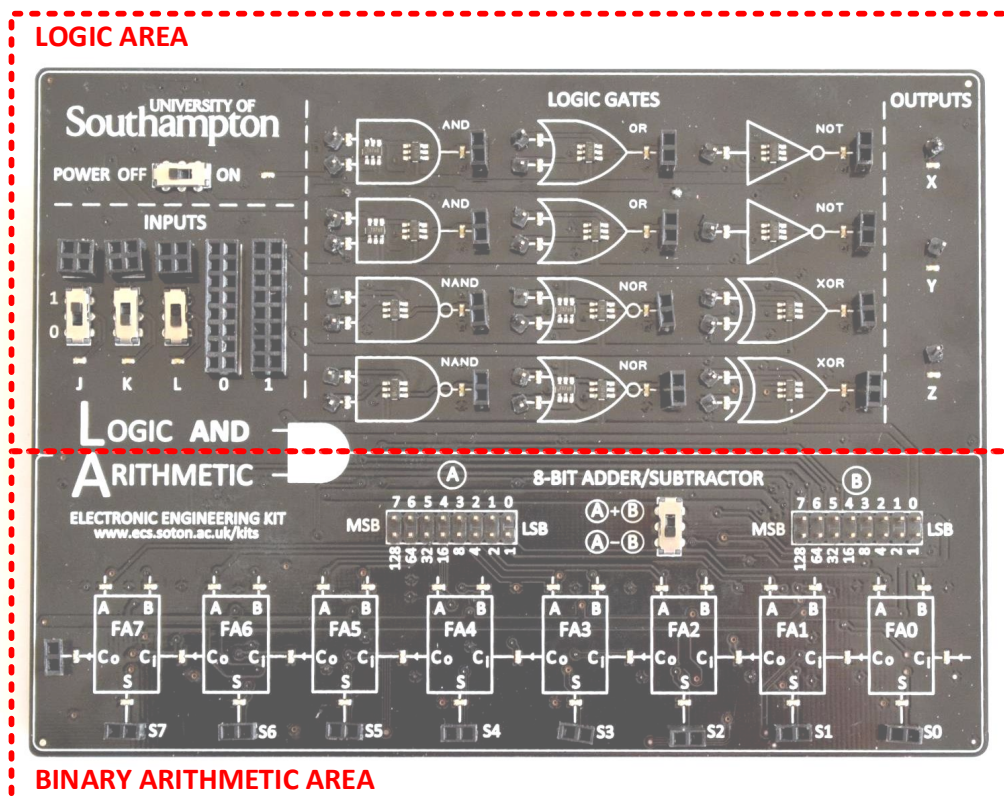
FIGURE 3: The two areas of the kit

These areas are discussed in turn in the following subsections.

2.1   LOGIC (upper) Area

The LOGIC area (the top half of the kit) can be used to explore logic gates, as well as to implement simple logic functions and circuits. The LOGIC area is split into a further three sections:

- **INPUTS** section (left);
- **LOGIC GATES** section (middle);
- **OUTPUTS** section (right).

These three sections of the LOGIC area are explored in more detail in the following subsections of these lab notes.

2.1.1   INPUTS section (left)

The INPUTS section provides two different types of inputs (shown in Figure 4):

- **Fixed inputs:** two vertical 2x8 pin headers on the right-hand side of the INPUTS section (marked '0' and '1') provide logic '0' and logic '1' signals.

- **Switchable inputs:** three 2x2 pin headers on the left-hand side of the INPUTS section (marked 'J', 'K' and 'L') provide switchable logic '0' or logic '1' signals. The value can be changed using the switch underneath each of the three 2x2 pin headers, and the orange LED underneath indicates the current value.
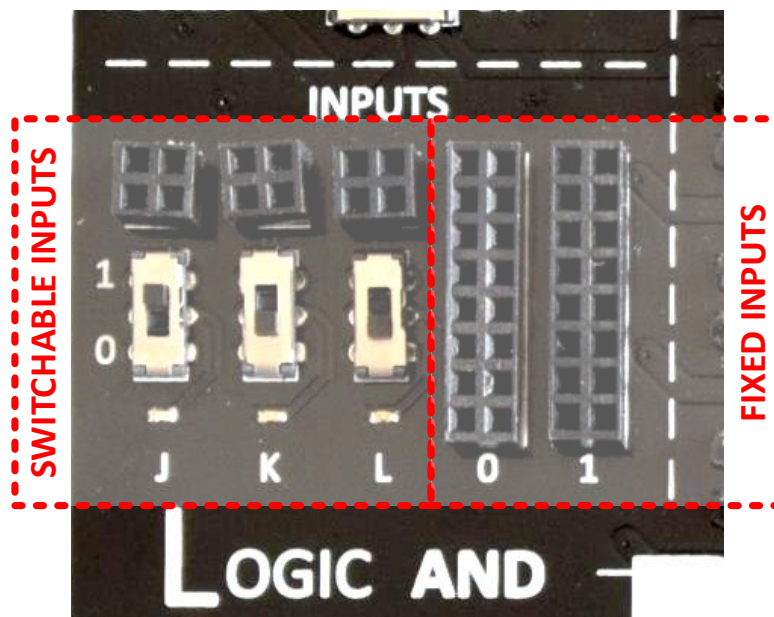
FIGURE 4: The INPUTS section

2.1.2   LOGIC GATES section (middle)

The LOGIC GATES section provides a pair of 6 different logic gates (AND, OR, NAND, NOR, XOR, and NOT). Connections can be made to the inputs (on the left of the gate) and output (on the right of the gate) using the pin headers and wires. Orange LEDs display the logic state of every input and output while the kit's power is turned on.

2.1.3   OUTPUTS section (right)

For some experiments, it can be useful to have a specific output – for example when implementing a logic function of the form 'y ='. For this reason, three outputs (X, Y, and Z) are provided in the OUTPUTS section of the kit. An LED next to the pin indicates the logic state of the outputs at all time.

2.2 ARITHMETIC (lower) Area

The ARITHMETIC area of the kit provides an 8-bit two's complement adder/subtractor circuit. This allows experimentation with both unsigned and signed Boolean arithmetic, as well as a different way of observing and understanding binary number systems. The ARITHMETIC area of the kit is split into a further three sections:

- INPUTS AND FUNCTION SELECTION section (top);
- ADDER/SUBTRACTOR section (middle);
- OUTPUTS section (bottom).

These three sections of the ARITHMETIC area are explored in more detail in the following subsections of these lab notes.

2.2.1 INPUTS AND FUNCTION SELECTION section (top)

The two 8x2 pin headers allow selection of the two 8-bit binary numbers, 'A' and 'B'. The Least Significant Bit (LSB) and Most Significant Bit (MSB) are illustrated on each, along with the position and weighting (i.e. $2^{position}$). The numbers are selected by using the jumpers included in the kit – where a jumper is present, that bit is logic '1'; where no jumper is present, that bit is logic '0'. Figure 5 shows an example where bits 0 and 4 of number 'A' have been set by jumpers, i.e. $A = 00010001_2$.
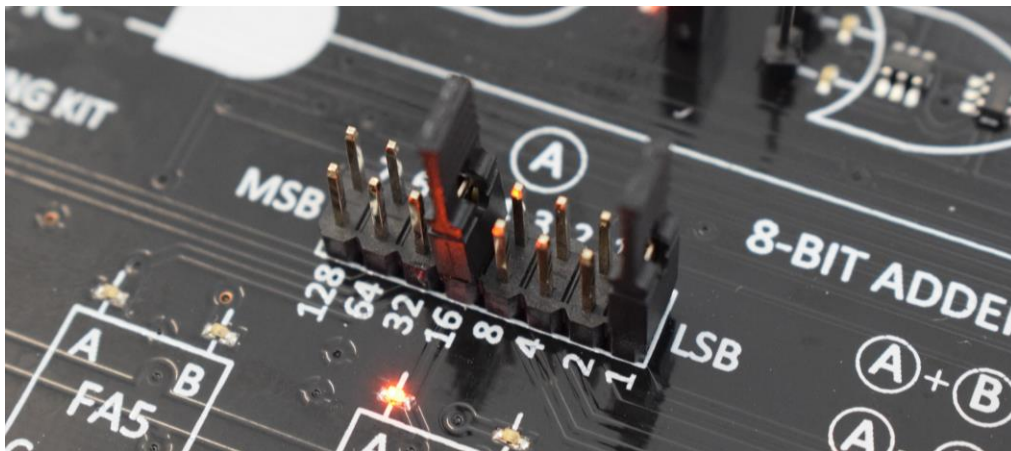
FIGURE 5: Bits 0 and 4 of number 'A' have been set by jumpers, i.e. $A = 00010001_2$.

The switch in-between A and B is used to selection the operation: either A+B or A-B:

- **A+B**: this can be used to illustrate both unsigned and signed (two's complement) addition, as the process is identical for both.
- **A-B**: this performs two's complement subtraction. To perform this, the fact that $A - B = A + (-B)$ is exploited. In two's complement, B can be negated by "flipping the bits and adding one". Therefore, when the switch is in the A-B position, all of the bits on B will be inverted ("flip the bits"), and a logic '1' inserted into the LSB carry-in ("add one").

2.2.2 ADDER/SUBTRACTOR section (middle)

This section shows the schematic of an 8-bit two's complement adder/subtractor with ripple carry, where each rectangular box represents a single Full Adder (with inputs A, B and Carry-In, and outputs Sum and Carry-Out). LEDs on the inputs and outputs of each Full Adder illustrate the state of all signals.

The schematic of a Full Adder is shown in Figure 6 (NOTE: this may not be the same as that implemented on the kit, but is the simpler version formed from combining two Half Adders).

NOTE: the actual circuitry for the adder/subtractor is hidden on the rear of the kit.

FIGURE 6: A Full Adder formed from two Half Adders (marked in grey).

For those interested, an additional XOR gate is attached to the 'B' input of each Full Adder to allow subtraction. One input of this XOR gate is connected to the appropriate bit of number 'B', the other is connected to the A+B/A-B function switch (this is also connected to the Carry-In of the Least-Significant Full Adder FA0).

- When in A+B mode, the switch provides logic '0', which means that the additional XOR gate is effectively transparent.

- When in A-B mode, the switch provides logic '1'. This causes the XOR gate to invert the appropriate bit of number 'B', and also to inject a '1' into the Carry-In of FA0.

2.2.3   OUTPUTS section (bottom)

The 'Sum' output from each Full Adder is available to connect to via pin headers S0-S7. These can then be used as inputs in the LOGIC area, if desired.

3   Laboratory work

This section of the lab notes will walk you through a number of different exercises, and 20 questions (marked in red), to illustrate the uses and capabilities of the kit.

3.1   Logic Gates and Truth Tables

The first exercise allows students to practically explore the truth tables of logic gates.

Q1: Experimenting with both the Fixed Inputs and Switchable Inputs in the INPUTS section of the kit's LOGIC area, complete the truth tables for the following logic gates:

A	B	A AND B

A	B	A XOR B

3.2   Simple Logic Functions

Students can then understand Boolean expressions by breaking them down into the individual operations required, and explore the truth table of the overall function.

Q2: Using the LOGIC area of the kit, implement the function $Y = K \oplus \left(K \cdot (J + L) \right)$. What is the only combination of J, K and L that gives $Y = 1$?

J	K	L	Y
			1

NOTE: do not try to minimize this expression (at least in the first instance!).

3.3   Unsigned Binary Addition

The next exercise explores unsigned binary addition, using the board to illustrate how it is performed. In these notes, the base is denoted by a subscript, i.e. 10_{10} is ten in decimal, whereas 10_2 is a binary number (with a decimal value of 2_{10}).

Q3: Using the ARITHMETIC area of the kit, perform the following calculations and write down the answer provided on the S outputs:

- $0000\ 0000_2 + 0000\ 0001_2$ $(0_{10}\ +\ 1_{10})$ — — — — — — — —$_2$

- $0000\ 0001_2 + 0000\ 0001_2$ $(1_{10}\ +\ 1_{10})$ — — — — — — — —$_2$

Q4: Perform the following calculation by hand (without converting to decimal), and then check your arithmetic (including the intermediate carry values) using the kit:

- $1001\ 0110_2 + 0010\ 1111_2$ — — — — — — — —$_2$

Q5: Can you understand how the kit did this, using the Full Adder as a building block?

3.3.1 Overflow

The following questions explore overflow in unsigned Boolean arithmetic.

Q6: Using the ARITHMETIC area of the kit, perform the following calculation:

- $1111\ 0000_2 + 1111\ 0000_2$ $(240_{10} + 240_{10})$ __ __ __ __ __ __ __ __ $_2$

Q7: Is this the correct answer? What happened? How could this be detected?

3.4 Two's Complement Binary Subtraction

Questions 8-10 explore two's complement binary subtraction, using the kit to illustrate what is happening.

Q8: Perform the following calculation by hand (without converting to decimal), and then check your arithmetic (including the intermediate carry values) using the ARITHMETIC area of the kit:

- $0000\ 0001_2$ - $0000\ 0001_2$ (1_{10} - 1_{10})   __ __ __ __ __ __ __ __ $_2$

Q9: Is the correct answer produced?

Q10: Can you understand what is happening here? Why do all the LEDs representing number 'B' change when the switch is put in the A-B position?

3.4.1   Two's Complement Overflow

The following four questions allow students to explore overflow in twos complement number systems.

Q11: While you still have the kit configured for $0000\ 0001_2$ - $0000\ 0001_2$, take a look at the LEDs. The Carry-Out of FA7 is lit. Has overflow occurred? Is the result valid?

Q12: Using the ARITHMETIC area of the kit, perform the following two's complement calculation:

- $1111\ 0000_2$ + $1111\ 0000_2$ (-16_{10} + -16_{10})   __ __ __ __ __ __ __ __ $_2$

Q13: Is this the correct answer? Has overflow occurred? Does it conflict with Q7?

Q14: Using the ARITHMETIC area of the kit, perform the following two's complement calculation:

- $1000\ 0000_2$ (-128_{10}) $+ 1000\ 0000_2$ (-128_{10} + -128_{10})   __ __ __ __ __ __ __ __ $_2$

Q15: Is this the correct answer? Has overflow occurred? How could you detect this?

3.5   The Full Adder

Q16: Using the LOGIC area of the kit, implement the circuit for a Full Adder (see Figure 6) and complete its truth table:

A	B	C (Carry-In)	X (Sum)	Y (Carry-Out)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3.5.1   Creating a 9-bit Adder

The remaining four questions bring together all parts of the kit, to extend the 8-bit adder into a 9-bit adder.

Q17: Combine the LOGIC and ARITHMETIC areas of the kit to create a 9-bit Adder. To do this, you'll need to connect the 'C' input of your Full Adder circuit to the Carry-Out of FA7.

Q18: Check its operation by performing the calculation:

- $0\ 1111\ 0000_2 + 0\ 1111\ 0001_2$ __ __ __ __ __ __ __ __ __ $_2$

Q19: The first time you performed this calculation (Q6), the result was not valid. Is it now?

Q20: If you try to do 9-bit subtraction using your circuit, it will not (always) work. Why is this?

3.6   What else?

If you think of any comments or feedback on the kit while you are using it, please jot them down in the box below, and hand this sheet to one of the organisers at the end of the session. If you forget to, please email it to kits@ecs.soton.ac.uk.

3.6   What else?

If you think of any comments or feedback on the kit while you are using it, please jot them down in the box below, and hand this sheet to one of the organisers at the end of the session. If you forget to, please email it to kits@ecs.soton.ac.uk.