# Digital Circuit Design

Queen Mary University of London

School of Electronic Engineering and Computer Science

Course Code: EBU4202

| | |
|---|---|
| Lab Sheet 2: Binary Adder Circuits | Date: _____ |
| Student's Surname, First Name (in English): _____ | Student's BUPT Number, Class Number: _____ |
| Email Username (102x*xxxx*): _____ | Total Mark (out of **35**): _____ |

**IMPORTANT**:

(a) In advance of the lab session: Print this Lab Sheet, read it and complete all the indicated "Preparatory Work".
(b) On the day of the lab session: Ensure that you have your notes to hand (taken during **Lab Session 1**) about how to set up the *Breadboard*.
(c) Write all your answers on this Lab Sheet, where indicated.
(d) Use additional A4 sheets of paper if you require more space to write your answers, ensuring that the question numbers are indicated clearly.
(e) Before handing in your Lab Sheet, make sure that you fill in the Table above with your personal details, ***and*** staple any additional answer sheets (with your name written on them) together with this Lab Sheet.
(f) Any question marked with the symbol **(***)** requires that you demonstrate your work to one of the TAs.

## 1. Learning Objectives

This Lab Session builds on the work done in **Lab Session 1** and from it, you should learn how to construct more complex circuits by implementing binary adder circuits (i.e., *half* and *full adders*).

Please note that all the circuits you build during this experiment will require the configured *breadboard* previously described in **Lab Sheet 1**; you need to assemble this before you start your experiments.

## 2. Preparatory Work

Read the Lab Sheet thoroughly, do all the background design work for **Experiments 1** and **2** and produce properly numbered circuit diagrams where indicated. You should do all the **Preparatory Work** before the Lab Session takes place.

**Question 2.1** [**4 Marks** ] – Analyse the circuits shown in **Figure 4** and **Figure 5** , and produce Boolean statements for the various outputs in terms of the inputs.

**Answer to Question 2.1**

## 3. Logic Gates

For these experiments, you will require the following TTL integrated circuits:

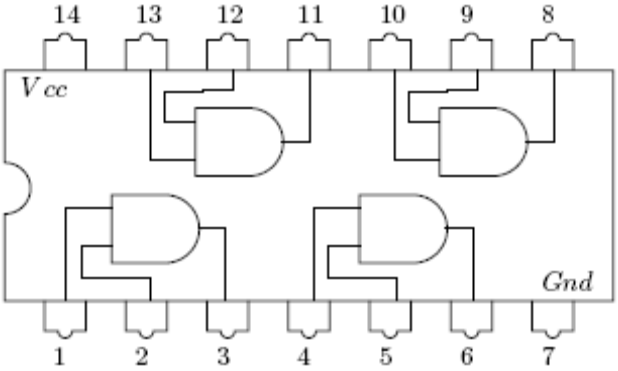



**Figure 1** – 7408 (labelled HD74LS08P) quadruple 2-input AND gates.

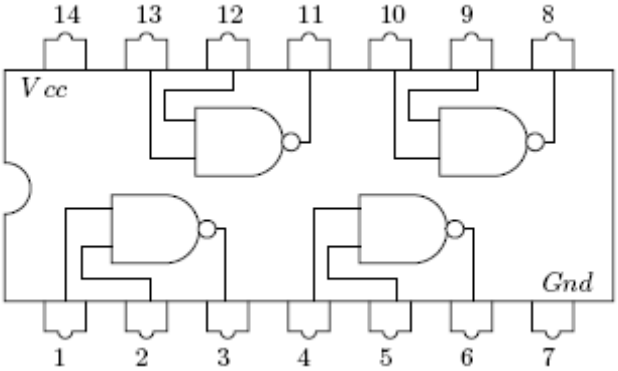



**Figure 2** – 7400 (labelled HD74LS00P) quadruple 2-input NAND gates.

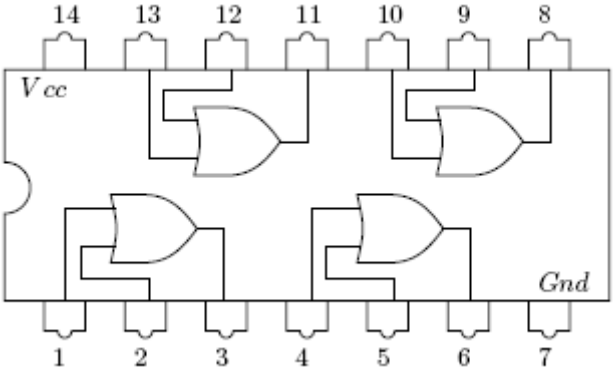


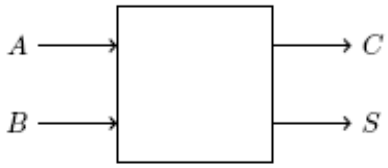

**Figure 3** – 7432 (labelled HD74LS32P) quadruple 2-input OR gates.

## 4. Experiments

**EXPERIMENT 1** – Half Adder

A Half Adder is shown in block form in *part 1)* of **Figure 4**; **A** and **B** are the 1-bit inputs, and the outputs are the sum bit (**S**) and the carry bit (**C**). Thus e.g., **1+0** gives **S=1** and no carry (i.e., **C=0**), and **1+1** gives **S=0** and the carry bit is **C=1**.

***1)***




***2)***

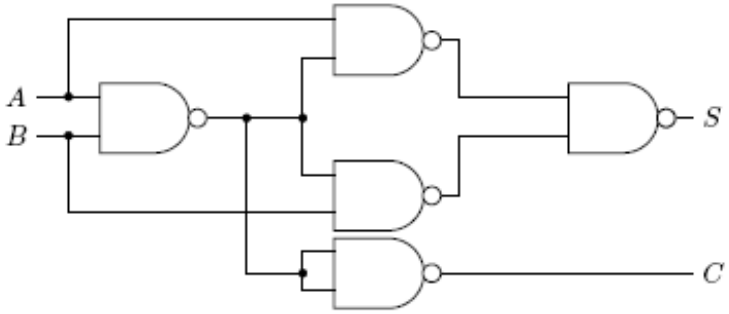



**Figure 4** – Half Adder: Block diagram and NAND implementation.

**(***) Question 4.1.1** [**6 Marks**] – Construct the NAND implementation of the Half Adder which is given in *part 2)* of **Figure 4** (using 7400 ICs). By checking its operation against the Truth Table (please include this in your answer), verify that your circuit is functioning correctly as a Half Adder. Your answer should include a fully labelled chip diagram, where all the pin numbers and gate connections are indicated.

**Answer to Question 4.1.1**

**(***) Question 4.1.2** [**7 Marks**] – Design (**Preparatory work**) and build an alternative Half Adder using 7408 and 7432 ICs (i.e., AND and OR gates, re spectively), and inverters constructed from 7400 (i.e., NAND) gates. Verify its functionality. Your answer should include a fully labelled chip diagram, where all the pin numbers and gate connections are indicated.

**Answer to Question 4.1.2**

**Question 4.1.3** [**3 Marks**] – (**Preparatory work**) If an XOR gate were available to you, how could you use it in a Half Adder? Your answer should also include a circuit diagram illustrating how you would do this.

**Answer to Question 4.1.3**

**Question 4.1.4** [**3 Marks**] – (**Preparatory work**) Prove, by using Boolean algebra (indicating any theorems applied in each step), that the NAND Half Adder implementation is valid.

**Answer to Question 4.1.4**

**EXPERIMENT 2** – Full Adder

The Half Adder is so-called this because it cannot accept a "carry-in" input from a previous stage. The Half Adder is quite satisfactory for adding 1-bit numbers or for the LSD (least significant digit) stage of an Adder for multiple bit numbers. However, in order to construct multiple bit Adders, we need to extend the Half Adder to accept a carry input for all digits beyond the LSD.

A Full Adder in block form is shown in *part 1)* of **Figure 5**. It can be built from two Half Adders, as shown in *part 2)* of Figure 5, for Half Adders with a **C'** output.

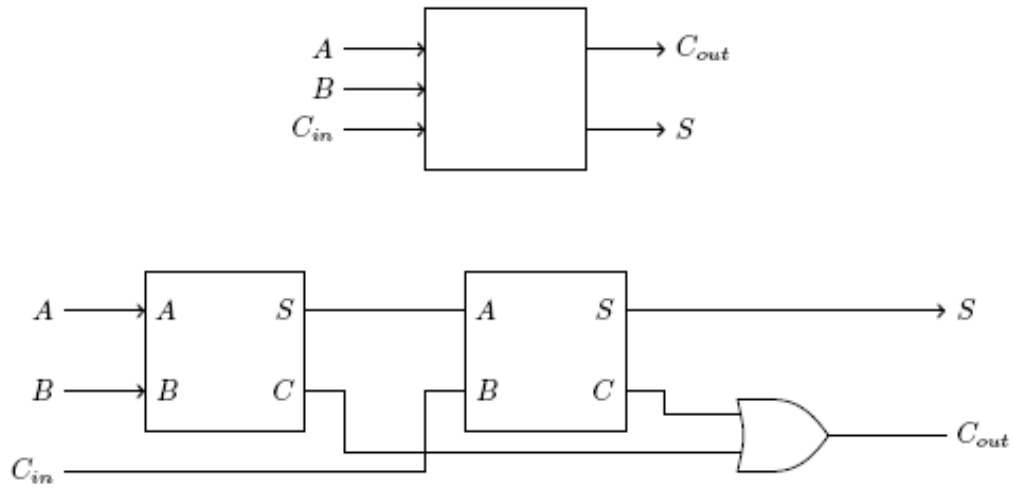



**Figure 5** – Full Adder built from 2 Half Adders: Block diagram and Circuit diagram.

**(***) Question 4.2.1** [**7 Marks**] – Build a Full Adder from 2 NAND gate Half Adders and check its operation against the Truth Table (please incl ude this in your answer). Your answer should include a fully labelled chip diagram, where all the pin numbers and gate connections are indicated.

**HINT**: Build and test two separate Half Adders before connecting them together as a Full Adder.

**Answer to Question 4.2.1**

(***) Question 4.2.2 [5 Marks] - In the space below, draw the diagram of two Full Adders connected together to allow the addition of two, 2-bit numbers. Draw a table showing the sum of two, 2-bit numbers and verify the circuit's operation.

Answer to question 4.2.2

## END OF Lab Session 2: Binary Adder Circuits