电子科技大学
格拉斯哥学院
Glasgow College, UESTC

UESTC3020 – Digital Circuit Design

COURSEWORK

PRESENTATION

v 1.0

Frederic Surre **2024/25**

BEng in Electrical and Electronic Engineering (Communications)

Description of Coursework & Tasks

1- Introduction

This coursework built on the first three labs and lectures of the course.

The objective of this coursework is to design a system that combine both combinational and sequential circuits and simulate the final design using VHDL descriptions.

The system you have to design include an adder and some sequential circuits and has to satisfy given specifications.

This is an open problem as there are more than one possible solution. Therefore, you have to think carefully and **justify your decisions** in the final report.

This project must be done in group of 2 (both partners must be in the same lab group as assigned at the start of the course)**.** No group of 3. Only when all students have been paired, the coordinator can arrange a group of 3. If there is an issue, Fred has the last word to organise the groups of 2.

You are expected to submit a report showing your work for both parts of the coursework. More information on the report format and how to submit it will be available on Moodle.

2- Preliminary remarks

- **Objectives:**
  - To design a complex system combining combinational and sequential.
  - To simulate a complex system.
  - To resolved design issues.
  - To investigate timing issues.
  - To properly justify design decision.

- **Equipment required**
  - Logisim (for combination circuit testing in Part 1).
  - Xillinx Vivado Webpack 2018.3 or 2019.1 (for sequential circuit and overall design validation in Parts 2 and 3) – a more recent version may be acceptable but may be an issue if your code needs to be tested during marking.

3- Introduction

Artificial Neural Networks (ANN)

An ANN is a network made of artificial neurons organised in layers. The input layer is the interface with the data that needs to be processed and the output layer is the layer where the final calculations are done. However, the key idea is to have at least one hidden layer that links the input layer to the output layer. Historically, ANN were using only one hidden layer. In recent time and with the advent of Deep Learning, ANNs are regularly using multiple hidden layers connected to each other.

Another key characteristic of an ANN is all the neurons of one layer are connected to all the neurons of the next layer with defined weights (these weights can be varied, especially during the training phase). Figure 1 is a high-level representation of a small-scale ANN with two input neurons, three neurons in the hidden layer and one neuron in the output layer.

Artificial Neuron

An artificial neuron is, in theory, very simple. It is connected to a number of other neurons and its input receives the output value of each of these neurons. Each connection having an assigned specific weight. The first stage of the neuron does the weighted sum of the contribution of the

preceding neurons. Then, to have the output value, this sum is fed into a non-linear function. Various functions have been tested over the year and you do not need to know about them in this project. Figure 2 presents a schematic view of an artificial neuron (for simplicity, in this project, we do not consider the bias input to the sum).

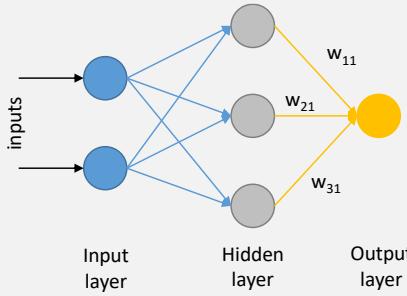

Figure 1: High-level block diagram of an ANN.

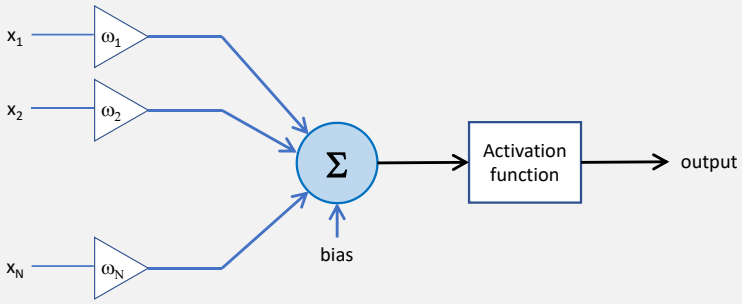

Figure 2: Schematic of an artificial neuron.

As mentioned, the theoretical principle is not complicated, however, implementing artificial neurons using digital functions (which is currently an active field of research) presents some difficulties. For example, one such complication (but not the only one) is how to implement an adder than may have to add a large number of values. It is a problem as due to the potential large number of values to add a cascade of 2-input adders is not practical (too large footprint). This problem will of course be an important part of your design later in the project.

As a consequence, in this project, you are tasked with the design of a 'generalised' adder, which can add multiple samples using a single adder circuit (we will only focus on this function and not the weights and activation function in figure 2). The specifications your design must satisfy are given in the following sections.

4- Tasks to be completed

PART I: Adder (this is identical to the description of CW1)

Question 1: 2-bit adder.

In this part, you have to design a combinational circuit to meet set specifications and constraint. The circuit to design and how to implement it is specific to each lab group.

The specification of the 2-bit adder are: the circuit to design is a 2-bit adder (i.e. that implement arithmetic addition) with two inputs A and B, which are 2-bit signal, i.e. $A = A_1A_0$ and $B = B_1B_0$, and two outputs: one 2-bit output $S = S_1S_0$ (sum) and one 1-bit output C (carry out).

In help you in the design, the following tasks are strongly suggested.

(a) Task 1

Design the circuit following the procedure studied in lecture.

(b) Task 2

Implement your design using 2-input AND and OR gates.

(c) Task 3

Optimise the circuit to minimize the cost. To evaluate the cost, you must use the cost per gate provided in appendix 1 (be careful to use the correct table as each class has a different cost table).

(d) Task 4

Simulate your final design of the adder using Logisim and verify it satisfies specifications the truth table to verify that your implementation meets the specifications.

Question 2: Design Reuse - 4-bit adder.

In this question, you have to design a new circuit and reuse your design from question The specifications of the circuit to design are: The specification of the 2-bit adder are: the circuit to design is a 2-bit adder (i.e. that implement arithmetic addition) with two inputs A and B, which are 4-bit signal, i.e. $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$, and two outputs: one 4-bit output $S = S_3S_2S_1S_0$ (sum) and one 1-bit output *C* (carry out).

Your design in this Part must reuse the 2-bit adder you have validated in Task 4 (hint: you may need to use two copies of this design and design some small circuit to connect them). You cannot modify the design from Part I, it must be used as designed.

(e) Task 5

Propose a design for the 4-bit adder reusing the 2-bit adder designed in Part I. (Reminder: you can not modify your design in Part I. You must use exactly the same circuit as the one validated in Task 4)

(f) Task 6

- Using Logisim, validate your proposed design.

PART II: Validation of combinational design

In this part, you are expected to validate your design from Part I using VHDL. Consider the circuit you have proposed in Part I, complete the following tasks:

(g) Task 7

Write a VHDL behavioural description of the combinational circuit you had to design in Part I.

(h) Task 8

Simulate the behavioural description written in task 7.

(i) Task 9

Write a VHDL structural description of your optimised design derived in Part I.

(j) Task 10

Simulate the VHDL description from Task 9 and verify that the circuit works as expected.

PART III: Generalised adder

In this part, we are looking to use the adder you have validated in the previous part to add the weighted contributions of neurons in the previous layer.

The specifications of the 'generalised' adder are as follow:

- The samples are inputted one after another in the adder and each of them are 4-bit in parallel (be careful, each sample has 4-bit in parallel, so bits are inputted at the same time for a sample, but samples are inputted at different times).
- The summing should stop when a predetermined number of inputs, noted *N*, has been reached (this number is chosen by the user so you cannot assume a specific number in your design). You can assume that this number is at maximum 16 (it can be any number between 2 and 16.
- The clock period is 10ns and a new value is inputted at each tick of the clock until *N* values have been inputted.
- The output of the 'generalised' adder should only be outputted to the next circuit when the *N* values have been added (and not before).

(k) Task 11

Propose a block diagram of the design to be achieved (this step is also called functional design and include the different basic functions your design need and how they are connected to each other)

(l) Task 12

Deduce the specifications for each block in the block diagram from Task 11.

(m) Task 13

Write a behavioural description of each block in the block diagram from Task 11.

(n) Task 14

Simulate each block and justify the results (you should be using the specifications from Task 12 to validate the design).

(o) Task 15

Write a VHDL description of the complete system (you should use a structural description reusing the behavioural description for each block in Task 13).

(p) Task 16

Simulate the VHDL description from Task 15. The input sequence should be the following:

- Input0 = '0011';
- Input1 = '0101'
- Input2 = 'the sum of the last two digits of student 1's UofG number (module 16) in binary'.
- Input3 = 'the sum of the last two digits of student 2's UofG number (module 16) in binary'.

(q) Task 17

Based on the simulation, analyse the results and verify if your design is working (if it does not, explain what the problem is and how to solve it).

PART IV: Submission

This project must be done in group of 2. If there is an issue, Fred has the last word to organise the groups. The course coordinator will deal with single students (who may be assigned to a group).

More details on the submission will be available on Moodle during the week. At this point, the report is expected to be limited in length. A template will be provided on Moodle with detailed marking scheme and page limits for each section.
The expected submission date is early January (but it may change based on other deadlines you may have and exam dates). The exact date will be announced on Moodle.

For the design part, it is an open problem where more than one solution is possible. And it is important that you think deeply about the design as it is a good learning exercise. Neither the GTAs nor Ahmad/Fred will give you the solution for the design questions. Deciding when under uncertainty is one of the characteristics of an engineer. This project will contribute to your training as an engineer.

END OF COURSWEWORK