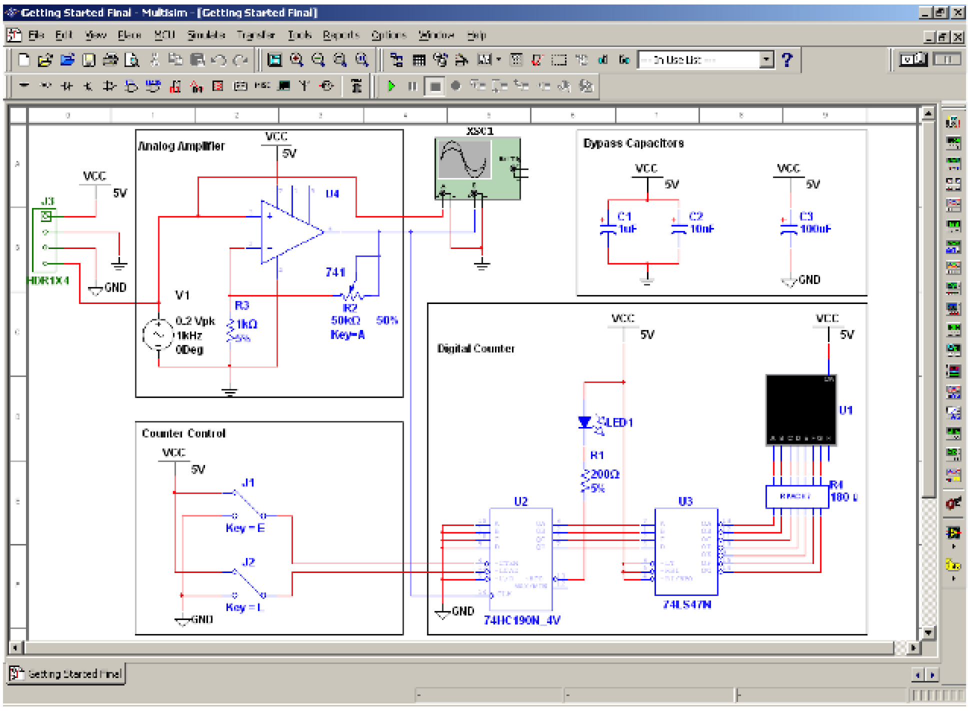
X6

Circuit Simulation

This exercise is an introduction to working with mixed-signal circuit simulation. Simulation is an important tool which can be used to identify design problems before committing to circuit construction. Nevertheless, it is only as good as the models in the simulator and the user driving the simulator. Hence, a circuit that appears to simulate successfully could operate differently when built. In the laboratory, you will simulate the operation of a mixed analogue and digital circuit using National Instruments Multisim software. In subsequent laboratories, you will design, build and test, a PCB based on the circuits in this exercise.



Schedule

Preparation Time	:	3 hours
Lab Time	:	3 hours

Items provided

Tools	:	
Components	:	
Equipment	:	
Software	:	National Instruments Multisim

Items to bring

Identity card

Laboratory logbook

Version: February 6, 2014

Steve R. Gunn and Alex S. Weddell
Electronics and Computer Science
University of Southampton

Before entering the laboratory you should read through this document and complete the preparatory tasks detailed in section 2.

> **Academic Integrity** – *If you wish you may undertake the preparation jointly with other students. If you do so it is important that you acknowledge this fact in your logbook. Similarly, you will probably want to use sources from the internet to help answer some of the questions. Again, record any sources in your logbook.*

You will undertake the exercise working with your laboratory partner. During the exercise you should use your logbook to record your observations, which you can refer to in the future – perhaps to write a formal report on the exercise, or to remind you about the procedures. As such it should be legible, and observations should be clearly referenced to the appropriate part of the exercise. As a guide the ✎ symbol has been used to indicate a mandatory entry in your logbook. However, you should always record additional observations whenever something unexpected occurs, or when you discover something of interest.

Notation

This document uses the following conventions:

✎	An entry should be made in your logbook

1 Introduction

This laboratory exercise aims to:

- Introduce you to the simulation of electronic circuits
- Give you experience in using NI Multisim software for circuit simulation
- Demonstrate how to simulate mixed analogue and digital systems

1.1 Outcomes

At the end of the exercise you should be able to:

- Capture schematics for analogue and digital circuits
- Simulate analogue and digital circuits using transient and AC analysis

Previous labs have introduced analogue and digital circuits, which you have built on a breadboard and tested. In this exercise, you will learn how to simulate circuit behaviour (for both analogue and digital circuits) using National Instruments Multisim software. The simulation capabilities of Multisim are based on the widely-used SPICE simulation system. Multisim is available for you to download, is installed on `roo` and the laboratory machines.

2 Preparation

2.1 Simulation software

Watch the online lecture L7. Answer the following questions:

1. What is meant by *schematic capture* and *circuit simulation*? ✎
2. What is a *net*, and (in the context of schematic capture) what is the relationship between a wire and a net? ✎
3. How can you tell that two crossing wires are connected? ✎
4. What is a SPICE model, and why is it needed for simulation? ✎
5. What is the difference between *transient analysis* and *AC analysis*? ✎
6. What are the x-axes on graphs output by each of these types of simulation? ✎
7. What can be plotted on the y-axis of the graphs output by each of these types of simulations? ✎

Familiarise yourself with the content of chapter 2 of the NI Circuit Design Suite - Getting Started guide[]. You do not need to read chapter 3 – you will be using a better piece of software for your PCB design in future laboratory exercises.

For the following preparatory sections you are required to use the Multisim software. You can do this by either downloading Multisim, logging on to `roo` or by using one of the laboratory machines.

2.2 Analogue simulation

8. In the components toolbar, click the *Place analog* button. Ensure that `<All families>` is selected. Search for the MCP601T-I/OT component by typing *mcp601* into the box below *Component*. Click *View model*. What information is there in the SPICE model of this component?
9. What does the Q factor of a resonant circuit measure? How is it related to the bandwidth of the circuit?
10. Compute the Q factor and the bandwidth of the circuit in figure 2.
11. Compute the resonant frequency of the LCR circuit in figure 2.
12. Calculate the R_1C_1 time constant of the AM demodulator in figure 3.
13. Explain the concept of a *3db point* in filter design.

2.3 Digital simulation

As part of the lab, you will be simulating the operation of sequential digital circuits using D-type flip-flops on 74HC74 integrated circuits (ICs). You may find it helpful to refer back to your notes from Lab T3, and from your ELEC1202 lectures on sequential circuits.

14. Look at the datasheet for the 74HC74 (dual D-type flip-flop). What is its typical propagation delay at 5V?
15. In the components toolbar, click the *Place CMOS* button and select the 74HC74 family. Select the 74HC74D_4V component and click *View model*. How is this model different from the SPICE model of the MCP601T-I/OT?
16. What is a *Linear Feedback Shift Register* (LFSR)?
17. What is the sequence length of the LFSR in figure 5?
18. Why is the second input to the XOR gate connected to $\overline{Q3}$ and not $Q3$?
19. What is *Manchester encoding*?
20. Give one advantage and one disadvantage of a Manchester encoding scheme?
21. For both the LFSR circuit in figure 5 and the Manchester encoding circuit in figure 6, compute the maximum propagation delay through the circuit. To do this consider the time it takes for all parts of the circuit to stabilise after a clock edge is applied.

3 Laboratory Work

3.1 Analogue simulation

In this section you will analyse the behaviour of an RFID detector circuit which is shown in figure 1. The input v_i will be driven by a 125kHz square wave from a microcontroller. The circuit operates by inductive coupling of L1 with the RFID tag or card and this amplitude modulates the 125kHz carrier frequency with a Manchester encoded signal at ~ 2kHz. D1, C2 and R1 demodulate the signal, C3 effectively high pass filters the signal and the remaining components form an active low pass filter to remove remnants of the carrier and amplify the weak data signal so that it is in a suitable form to be fed into the digital input of a microcontroller.

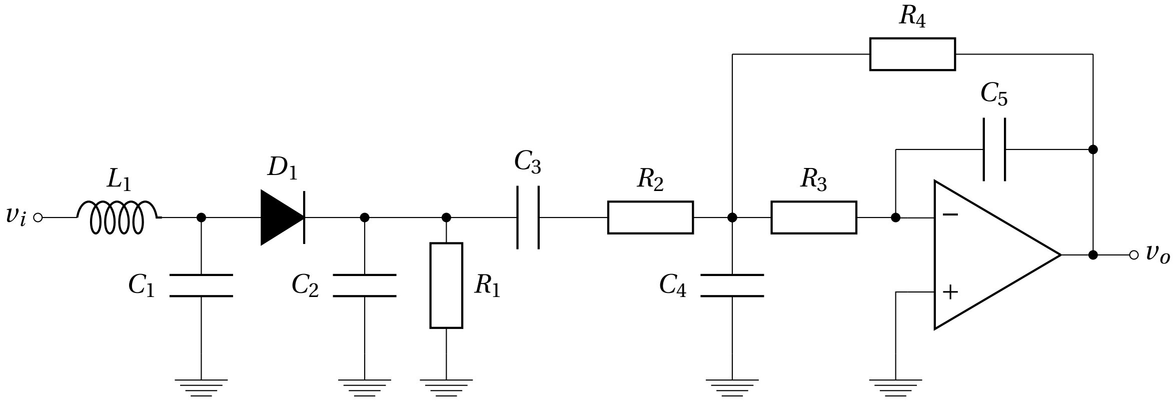


FIGURE 1: RFID detector circuit

Initially you will work on simulating sub-circuits of the overall detector in isolation. It is important to remember that this neglects any loading effects of subsequent stages. To provide a simple approximation of subsequent loading we can add a resistive load to the output, but it should be remembered that rarely is an input stage purely resistive.

3.1.1 LCR circuit

In Multisim, create a new blank design and draw the schematic of the LCR circuit shown in figure 2. To add a component, right click and select *Place Component* (or use the shortcut `[ctrl]+W`). The components (INDUCTOR, CAPACITOR, RESISTOR, AC_VOLTAGE) you require can be found under the *Sources* and *Basic* groups. Use the `[ctrl]+R` shortcut to rotate the component before placement. Then join the components together by moving the mouse over the end of a component and click to create a wire to another part of the circuit. Finally, add a net name label to the wire joining the inductor to the voltage source by double-clicking on it and setting it to VIN and checking the box to display the name. Repeat for the wire joining the inductor to the capacitor and label it VOUT. Adding net names like this makes it easy to identify which nodes you wish to plot in the simulation. Save your design. You are now ready to perform your first simulation.

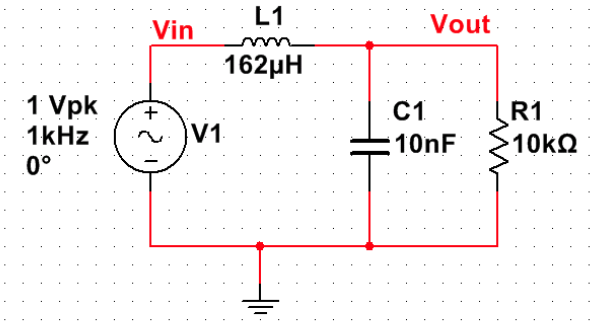


Figure 2: LCR schematic

AC analysis

Our first goal is to understand how the circuit will respond to different frequencies of a sine wave voltage source. The AC analysis will sweep the voltage source over a range that you specify and plot circuit measurements at these different frequencies[1]. To configure the simulation select *Simulate* > *Analyses* > *AC Analysis*. You should change the frequency range to start at 10 Hz and to end at 10 MHz. Increase the points per decade to 100.

The *Output* tab lets you select which output signal(s) you are interested in. Here, you should select Vin and Vout, adding them to the selected variables for analysis. Use defaults for the other parameters and click *Simulate*. You should see a graph of magnitude against frequency, and phase against frequency.

To make it easier to find the 3dB point, you can change the y-axis from a *linear* scale to a *dB* scale. To do this, right-click on the Magnitude axis and click *Axis Properties*, then alter the scale to *Decibels*.

You can also use cursors to allow you to look at the exact values. Click the *Show cursors* button on the toolbar. A vertical cursor will now appear. Try dragging it sideways around the graph window. The *3dB point* is the frequency where the magnitude drops to -3 dB. ✎

Transient analysis

The circuit in figure 2 forms part of the detector stage of the RFID reader that you will build. It will be driven by a square wave from a microcontroller to excite it as a resonant circuit. Our goal is to understand how this square wave will excite the circuit. To do this you should replace the AC voltage source with a PULSE_VOLTAGE. This is easily done by double-clicking on the voltage source and selecting *Replace*. Configure the new source to have - Initial value: 0V, Pulsed value: 3.0V and leave the other parameters at their default values.

The transient analysis will simulate how the voltages vary over time. To configure the simulation select *Simulate* > *Analyses* > *Transient Analysis*. You should change the End time to 1.0s. Confirm that Vin and Vout are selected in the output tab as before and click *Simulate*. You should see a graph of voltage against time. Record your observations. ✎

[1] Do not be confused about the 1kHz label next to the source – when doing AC analysis this frequency is overridden.

This simulation used a pulse frequency of 1kHz. Change this to 125kHz which is the frequency at which your circuit will be driven. Perform a new simulation and comment on any differences.

3.1.2 AM demodulator circuit

Create a new blank design and draw the schematic of the AM demodulator circuit shown in figure 3. Save your design.

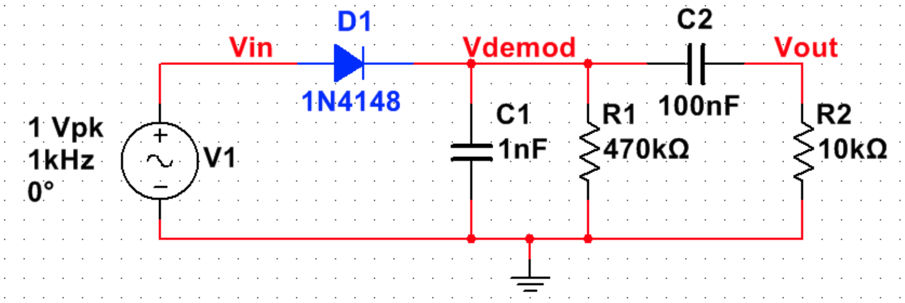


Figure 3: AM demodulator schematic

AC analysis

Use the same parameters as in section 3.1.1, add the three nets Vin, Vdemod and Vout to the output and run the simulation. What does the AC analysis tell you about the behaviour of the circuit? What is the purpose of C2?

Transient analysis

The detector will produce a signal which has a 125kHz carrier frequency whose amplitude is modulated by the data being transmitted. The RFID tags actually modulate the carrier with a Manchester encoded square wave, but we will simulate here with a sine wave modulation. To do this replace the AC voltage source with a AM_VOLTAGE. Configure the parameters - Carrier amplitude: 3.0V, Carrier frequency: 125kHz, Modulation index: 0.1, Intelligence frequency 2kHz. To speed up the simulation time change the value of capacitor C_2 to 10nF[2]. Set the simulation time to 5ms, add the three nets Vin, Vdemod and Vout to the output and run the simulation. Discuss what has happened to the carrier and intelligence frequencies of the original modulated signal.

3.1.3 MFB low pass active filter circuit

Our goal is to recover the data at the intelligence frequency of the modulated signal. From your transient analysis of the previous section, you should have observed that the carrier frequency is still evident in the output signal and the data at the intelligence frequency is of a small amplitude. We wish to feed the resulting signal to the digital input of a microcontroller so that the data can be recovered. As such we need to reduce the carrier remnants and amplify the data at the intelligence frequency

[2] This just gives slightly poorer low frequency response, but gives a quicker settling time.

to digital logic levels. This can be achieved by using an active filter which attenuates signals at the carrier frequency and amplifies signals at the intelligence frequency. There are a number of active filter configurations that one can choose from, but the Multiple Feedback (MFB) filter chosen here is advantageous in that it can be constructed from a single opamp circuit whilst providing suitable gain. Further details about the circuit and how to determine its component values can be found in Appendix A.

Create a new blank design and draw the schematic of the low pass active filter shown in figure 4. It is important to note that the voltage source must be configured to have a DC offset equal to the virtual ground, 1.5V; set the voltage (pk) to 0.1V. Save your design.

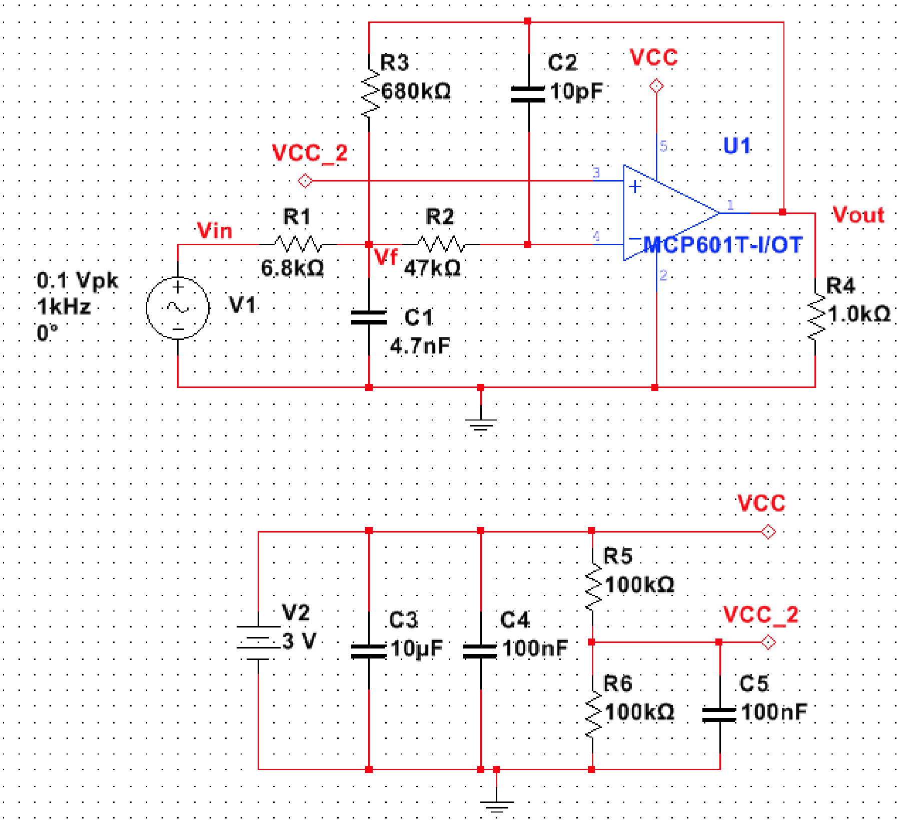


FIGURE 4: Second order all pole low pass multiple feedback active filter

It is important to understand the biasing of the opamp. There are two ways to bias an opamp: a) you can provide separate positive and negative supplies; b) you can create a virtual ground from a single supply at half of the supply voltage. The latter has advantages in that it reduces power supply circuitry, but great care must be taken to ensure that the output impedance of the circuitry matches the circuit's requirements. Here we use approach b) and since we are using the opamp in an MFB inverting configuration we are able to meet the requirements with a simple resistive voltage divider because the current through the non-inverting input is very small. However, in other opamp configurations you will require a second opamp configured as a voltage follower to buffer the output from the voltage divider to provide a suitable virtual ground.

AC analysis

Use the same parameters as in section 3.1.1, add the net VOUT to the output and run the simulation. What does the AC analysis tell you about the behaviour of the circuit? The data is located in the 2kHz part of the spectrum and the carrier is located at 125kHz. Determine the gain and attenuation for these two parts of the spectrum. What is the -3dB cutoff point for the filter? ✎

Transient analysis

Things are a little more complicated than simply looking at the magnitude response since we require to capture the timing information of the data signal which is contained in the edges of the signal. From Fourier analysis one can show that a square wave is made up from an infinite number of sine waves whose amplitude decreases with frequency - hence we should be careful. Furthermore, you should have observed that the phase information from the AC analysis showed that it varied with frequency. Potentially this is a problem. Another way to verify whether the filter is appropriate is to use a transient analysis to determine whether the rising and falling edges of the output signal capture the timing information of the original data signal. Remember a delay is not a problem, provided all edges are delayed by the same amount.

To do this you should replace the AC voltage source with a PULSE_VOLTAGE. Configure the new source to have - Initial value: 0V, Pulsed value: 3.0V and leave the other parameters at their default values. Run the simulation and verify that that the timing information is retained. Estimate any loss in accuracy. ✎

3.2 Digital simulation

Multisim is capable of simulating both analogue and digital circuits. In this part of the exercise, you will be simulating the behaviour of a linear feedback shift register which is configured to generate a binary sequence of maximum length, $2^n - 1$ where n is the number of D-types. This will be used to simulate the 64-bit data stream that an RFID tag transmits. The tag data is then encoded using Manchester encoding. The second circuit you will investigate is a Manchester encoder. When simulating digital circuits one of the most important aspects is to understand whether the propagation delays of the elements will produce unwanted behaviour.

3.3 Linear feedback shift register (LFSR)

A LFSR is an interesting circuit that produces a binary sequence using D-type flip-flops. The D-types are configured as a shift register with the input to the first given by a combinatorial function of the D-type outputs. Here the LFSR is configured to generate its maximum sequence length by using a combinatorial function that XORs the output of the final two registers. This function generates the maximum sequence length for any LFSR.

Create a new blank design and draw the schematic of the LFSR shown in figure 5. You will use 74HC-series ICs, which can be found in the CMOS group of components. In the simulator these ICs come with the power supply already connected, but you can only choose from a fixed set of supply voltages (2V, 4V or 6V) - use the 4V versions (74HC_4V). You will also need to add some digital sources to the schematic: these are found in the DIGITAL_SOURCES family of the *Sources* group of components. The digital constant value can be set by double-clicking on it, as can the clock frequency.

Many 74-series components have more than one logic element. For example, the 74HC74D chips you are using in this exercise actually have two flip-flops. Multisim knows this, and will ask you which one of the elements (A or B) you want to use. When you come to place additional elements, if there are any unused elements available it will offer you the option of using one of these or adding a new component. It is good practice to reuse all elements from one package before using a new one, as you would typically do in construction to minimise component count and cost.

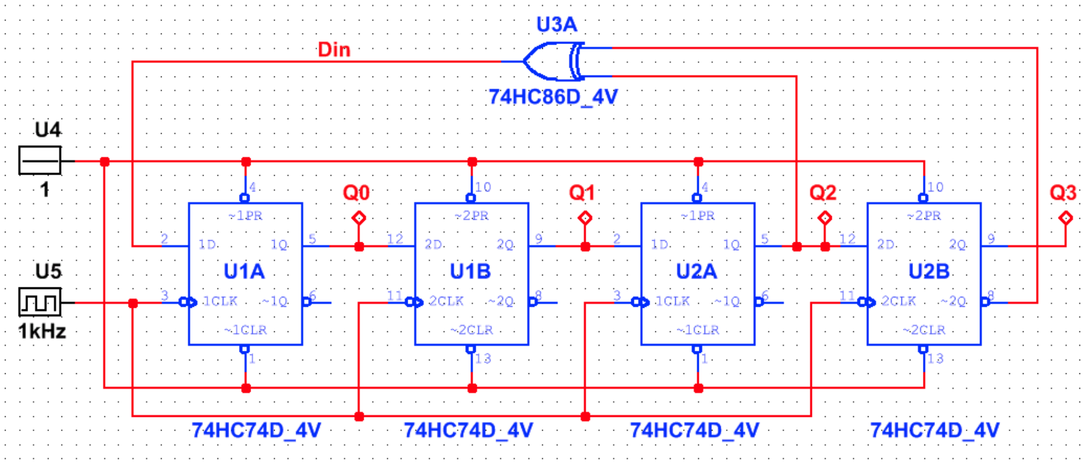


FIGURE 5: Maximum length linear feedback shift register

Transient analysis

Perform transient analysis on the system and note down the generated sequence at $Q3$. How well do the propagation delays that you observed from the data sheet in the preparation get modelled by the simulator?

3.4 Manchester encoder

Manchester encoding is attractive in that by encoding a '0' as ‾|_ and a '1' as _|‾ it is easy to recover the clock signal from the data stream at the receiver. However, it should be remembered that it is not the most optimal encoding scheme in terms of bandwidth utilisation. Create a copy of your LFSR design, by saving as a different name, and add the Manchester encoder to the schematic as shown in figure 6. Here you should use *on-page connectors* to avoid having lots of wires crossing each other. You can place such a connector by right clicking and selecting from the *place on schematic* list (or use `[ctrl]+[alt]+O`). Finally, use an off-page connector for the output of the encoder, in anticipation to use this block as part of a larger multi-sheet system simulation in section 4.

The new circuit now has two clocks. U7A is configured as a simple divide-by-2 counter which drives the pseudo-random sequence generator at half of the clock frequency. U7B and U8A form a similar toggling flip-flop which is controlled by the output of U10A. When this output is high the MAN_DATA output will simply toggle every time a clock pulse is received. Manchester encoding requires that the output is always toggled in the middle of a data bit, and also toggled at the end if the subsequent bit is the same as the last, which is implemented by the combinatorial logic of U9A and U10A.

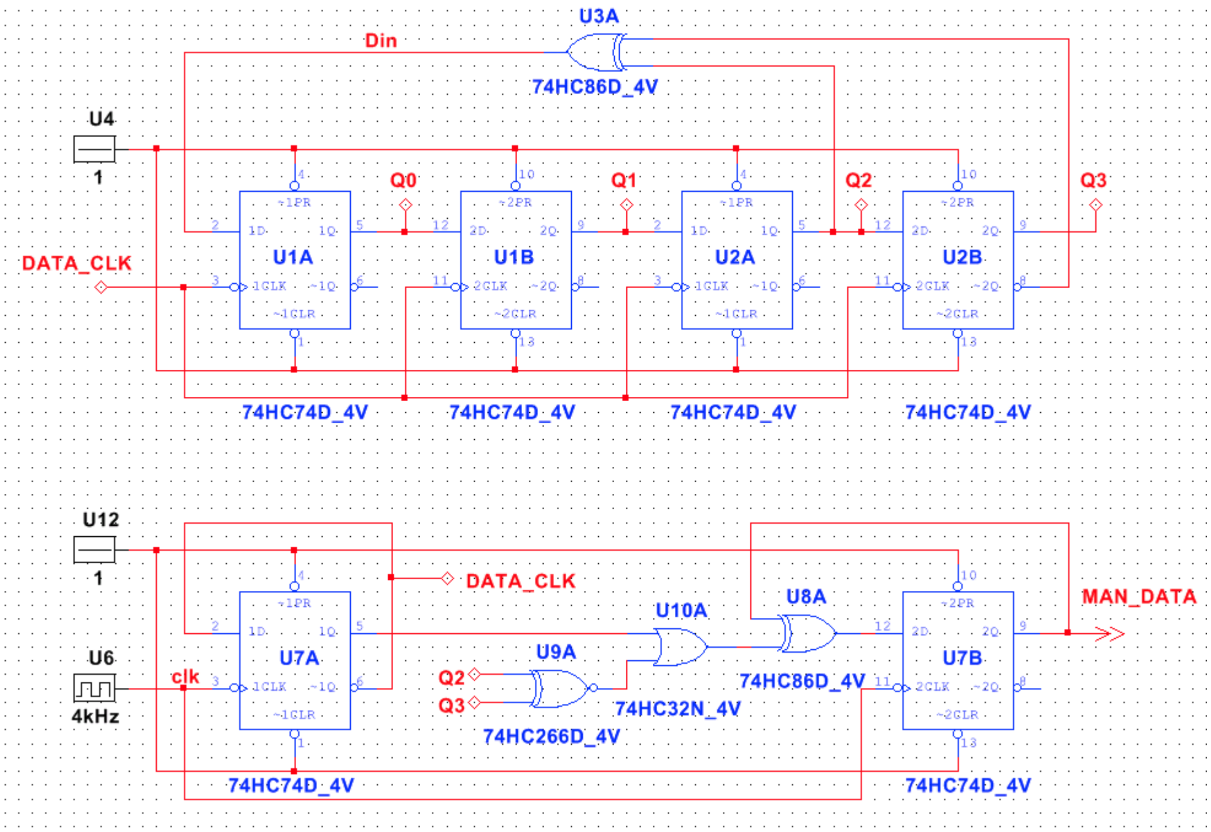


FIGURE 6: Manchester encoded LFSR sequence

Transient analysis

Perform transient analysis on the system and confirm that the output is in Manchester encoded form. Is there a clock frequency above which the system fails?

4 Optional Additional Work

Multisim uses two different simulators: a SPICE-based simulator for analogue circuits and an XSPICE-based event-driven simulator for digital circuits. The digital devices provided, such as 74HC-series logic, do not allow you to vary their supply voltage like you can on normal analogue components.

In fact, you do not even have to connect a power supply to these devices in simulation (although of course you do have to on the actual ICs when you prototype and build the circuits). You do not really have to worry about the way that Multisim simulates these devices, but you should remember that these simulations are carried out with the assumption that the supply voltage is a fixed value, and that it only simulates the logic states (e.g. 0 or 1) rather than actual voltages. This is particularly important to remember when you are simulating mixed analogue and digital circuits as it means that digital subcircuits can output voltages that are different from the supply voltage of the rest of the system. In this part of the lab, you are going to combine the analogue and digital sub-circuits that you have designed to simulate the complete RFID detector stage with a simulated RFID tag. In this instance, although the digital circuits have a supply of 4V and the analogue a supply of 3V, they have been designed to be compatible.

4.1 Mixed-signal simulation

The lab notes for this part will be updated before Monday 10th Feb.

A Second Order Multiple Feedback Low Pass Filter Analysis

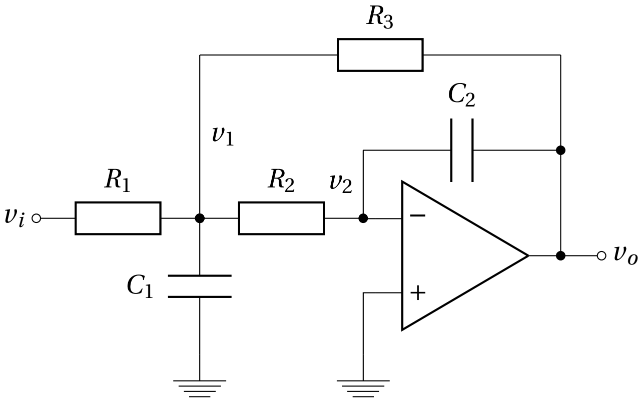


FIGURE 7: Second Order Multiple Feedback Low Pass Filter

The transfer function can be derived using the usual ideal op-amp approximation of infinite open loop gain, infinite input impedance and zero potential across its input terminals. Performing nodal analysis at v_2,

$$\frac{v_1 - v_2}{R_2} = \frac{v_2 - v_o}{\frac{1}{sC_2}}$$

and noting that $v_2 = 0$,

$$v_1 = -v_o sC_2 R_2 \tag{1}$$

Performing nodal analysis at v_1,

$$\frac{v_i - v_1}{R_1} = \frac{v_1}{R_2} + \frac{v_1 - v_o}{R_3} + v_1 sC_1$$

rearranging for v_i,

$$v_i = v_1 \left(1 + \frac{R_1}{R_2} + \frac{R_1}{R_3} + sC_1 R_1\right) - v_o \frac{R_1}{R_3}$$

and substituting v_1 from Equation 1,

$$v_i = -v_o sC_2 R_2 \left(1 + \frac{R_1}{R_2} + \frac{R_1}{R_3} + sC_1 R_1\right) - v_o \frac{R_1}{R_3}$$

Hence, the transfer function can be written,

$$\begin{aligned}
\frac{v_o}{v_i} &= -\frac{1}{sC_2 R_2 \left(1 + \frac{R_1}{R_2} + \frac{R_1}{R_3} + sC_1 R_1\right) + \frac{R_1}{R_3}} \\
&= -\frac{\frac{R_3}{R_1}}{s^2 C_2 R_2 C_1 R_3 + sC_2 \left(\frac{R_2 R_3}{R_1} + R_3 + R_2\right) + 1} \\
&= -\frac{\frac{R_3}{R_1} \frac{1}{R_2 R_3 C_1 C_2}}{s^2 + s\frac{C_2}{R_1} \frac{R_2 R_3 + R_1 R_3 + R_1 R_2}{R_2 R_3 C_1 C_2} + \frac{1}{R_2 R_3 C_1 C_2}}
\end{aligned}$$

Writing this second order all pole low pass filter in standard form,

$$\frac{v_o}{v_i} = \frac{A_0\omega_0^2}{s^2+2\zeta\omega_0 s+\omega_0^2}$$

we see that

$$\begin{aligned} A_0 &= -\frac{R_3}{R_1} \\ \omega_0 &= \frac{1}{\sqrt{R_2R_3C_1C_2}} \\ \zeta &= \frac{C_2}{2R_1}\frac{R_2R_3+R_1R_3+R_1R_2}{\sqrt{R_2R_3C_1C_2}} \end{aligned}$$

Now we have three equations and five unknowns, so our system is underdetermined. One could just choose the values for the two capacitors and then solve for the resistors using the three equations:

$$R_1 = \frac{\zeta C_1 - \sqrt{\zeta^2C_1^2-(1-A_0)\,C_1C_2}}{-A_0C_1C_2\omega_0} = -\frac{1}{A_0}(c-d) \tag{2}$$

$$R_2 = \frac{\zeta C_1 + \sqrt{\zeta^2C_1^2-(1-A_0)\,C_1C_2}}{(1-A_0)\,C_1C_2\omega_0} = \frac{1}{1-A_0}(c+d) \tag{3}$$

$$R_3 = \frac{\zeta C_1 - \sqrt{\zeta^2C_1^2-(1-A_0)\,C_1C_2}}{C_1C_2\omega_0} = (c-d) \tag{4}$$

where

$$\begin{aligned} c &= \frac{\zeta}{C_2\omega_0} \\ d &= \frac{\sqrt{\zeta^2-(1-A_0)\,\frac{C_2}{C_1}}}{C_2\omega_0} \end{aligned}$$

Clearly c is a real quantity. Hence, in order that the resistor values are real, it is necessary to ensure that d is real and $c > d$ which, noting $A_0 < 0$, is satisfied if

$$C_1 \geq \frac{1-A_0}{\zeta^2}C_2 \tag{5}$$

One can argue that the circuit is optimised when the capacitance ratio is minimised and hence when the equality holds in the above equation. In practice, capacitor values are only readily available in discrete values and these typically have tolerances of ±10%; therefore obtaining this equality is infeasible. Furthermore, it may be desirable to keep the input impedance as high as possible which is achieved by choosing a small value for C_2.

Hence, the following method can be used to determine the component values:

1. Specify A_0, ω_0 and ζ.

2. Choose C_2 to be a small value.

3. Choose C_1 to conservatively satisfy the inequality in equation 5.

4. Solve for R_1, R_2 and R_3 using equations 2, 3 and 4.

A.1 Example

1. $A_0 = -100$, $f_0 = 4000$ and $\zeta = \frac{1}{\sqrt{2}}$

2. Choose $C_2 = 10\text{pF}$

3. Choose $C_1 = 4.7\text{nF} \geq 202 \times 10\text{pF}$

4. $R_1 \approx 6.8\text{k}\Omega$, $R_2 \approx 47\text{k}\Omega$ and $R_3 \approx 680\text{k}\Omega$