

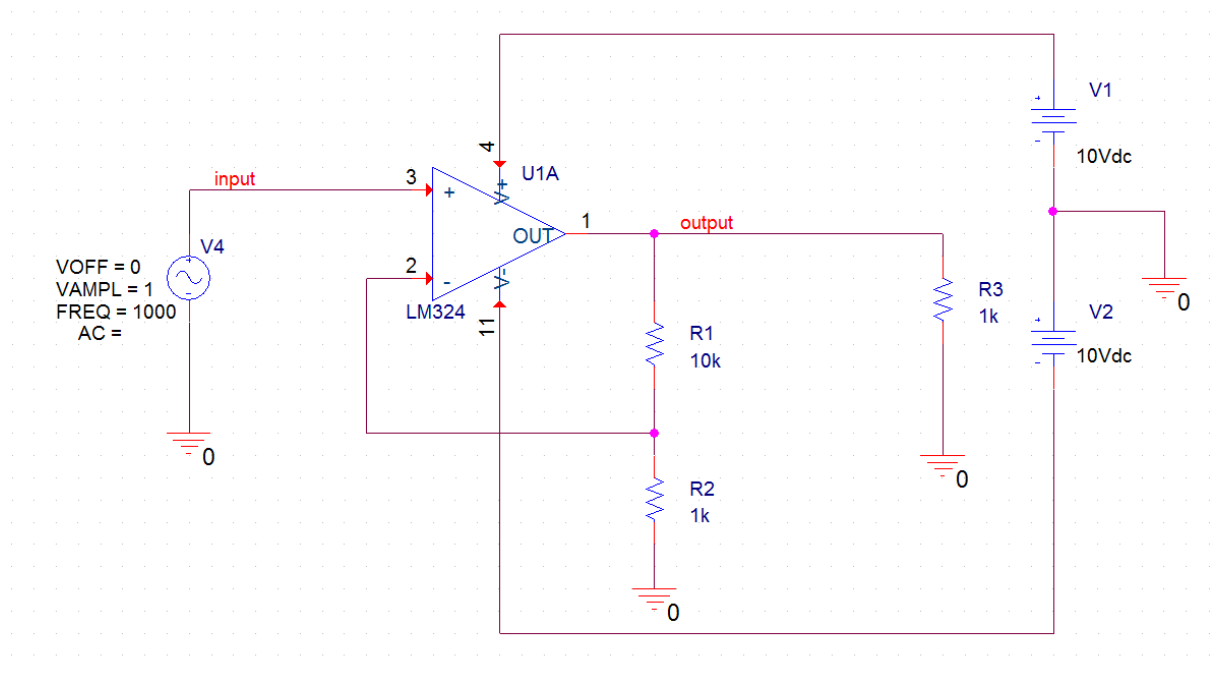
Imperial College of Science, Technology & Medicine

Department of Bioengineering

BEng/MEng in Biomedical Engineering

BE1.HEEL – Electrical Engineering 1<sup>st</sup> year Laboratory Sessions

## Op-Amp Applications Part I – Circuit emulation using SPICE



*Simulation Program with Integrated Circuit Emphasis*

**i. Laboratory Notice:**

Usual rules apply to the computer room as with any laboratory; no eating, drinking or personal stereos allowed.

For the computer part of this lab you will be working individually. You should keep a lab book for these simulations.

For the lab part of the course there will be a separate handbook. You will be working in groups of two or three. Every member of your group must keep a log book which will be assessed in one of the sessions towards the end of the module. There will also be a joint report to be submitted near the end of term (details will be given during *Op-Amp applications part II*).

*Note: If your log book is filled in as you do the exercises, the lab report will be straight forward and take little time to complete.*

## Table of Contents

Op-Amp Applications Part I – Circuit emulation using SPICE .....	1
i. Laboratory Notice: .....	2
1. Introduction .....	4
ii. What is SPICE? .....	4
iii. In this lab: .....	4
2. Using OrCAD: .....	5
3. Simulating circuits using Pspice: .....	8
iv. Transient Analysis.....	8
v. AC Sweep Analysis .....	9
4. Section One: Simple circuits using spice .....	10
vi. Exercise 1: .....	10
vii. Exercise 2: .....	10
viii. Exercise 3: Frequency Response. ....	11
5. Section 2: Op-Amp circuit simulations .....	13
ix. Exercise 4: .....	13
x. Exercise 5: .....	14
xi. Exercise 6: .....	15
6. Filters: .....	16
xii. Exercise 7: .....	17
xiii. Exercise 8: .....	18
7. Section 3: Electronic Stethoscope design: .....	19
xiv. Exercise 9: .....	19
8. Appendix A: Troubleshooting .....	21
9. Appendix B - Decibel.....	22
10. Appendix C Scaling Factors .....	23
11. Appendix D Potentiometer.....	24

## 1. Introduction

### ii. What is SPICE?

SPICE stands for *Simulation Program with Integrated Circuit Emphasis*. It is a type of software used to simulate electronic circuits. SPICE programs are able to perform many different analyses of electronic circuits such as time-domain response (giving results like you would see with an oscilloscope in the lab), small signal frequency response, the operating points of transistors and much more! The software has models for common circuit elements, both active and passive, and so is capable of simulating most electronic circuits. It is used widely in industry.

Much of the design of complex electronic circuits is performed using commercial versions of SPICE. In particular, chip design such as those for use in mobile phones and implantable devices would be virtually impossible without the use of such packages. It eliminates the cost of making prototype circuits, allowing the designer to make the final version with high confidence that the circuit will work. Analogue chip designers proficient in the use of such systems are some of the highest paid engineers in the world!

### iii. In this lab:

You will be using software called OrCAD Capture. It is produced by Cadence. It is one of the most popular eCAD (electronics Computer Aided Design) program for “Schematic Capture”. (A schematic is a “circuit diagram” – and “capture” suggests we are capturing the design intent). OrCAD is the graphical interface to the designer, the simulations will be done by a program called PSPICE (which is a windows variant of SPICE) which is called up from within OrCAD. You will be designing and simulating a variety of circuits very similar to those you will be building in part II of this module later this term.

Your main task in part I of the lab is to design a simple electronic stethoscope which you will then build during *OP-AMP Applications Part II*. The goal is to have it complete and working by the end of the term.

While this document is intended to be sufficiently detailed to allow you to complete all the exercises, there is a wealth of information available online about OrCAD and general PSPICE/ SPICE simulations. If you get stuck, try checking out the troubleshooting section at the end of this document.

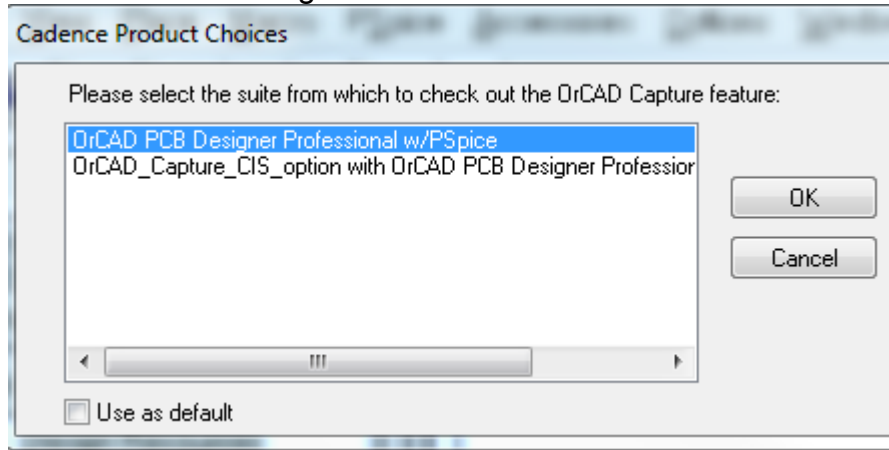
The full version of the software you will be using can be downloaded at home without charge from the Parallel Systems website and will be the same version (16.6) that you will be using in this lab.

Note that it will be useful if you have access to OrCAD/ PSPICE while building your electronic stethoscope pulse rate monitor in part II. You will often find that the initial values you have chosen for resistors and capacitors are not available, so a speedy recalculation and simulation can save a lot of time (the programs are available on the PCs in the lab B220).

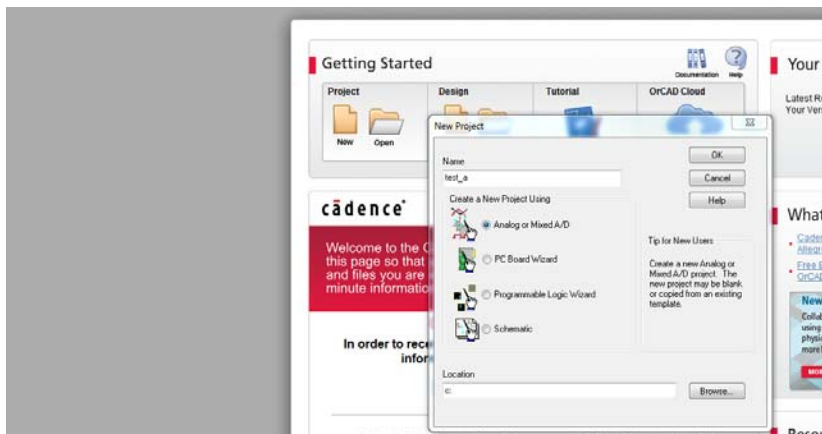
## 2. Using OrCAD:

### Step 1: Creating a circuit in schematic mode:

1. Open OrCAD Capture - Start/All Programs/Cadence/Release\_16.6/OrCAD Capture.
2. Select PCB Designer Professional w/PSPICE

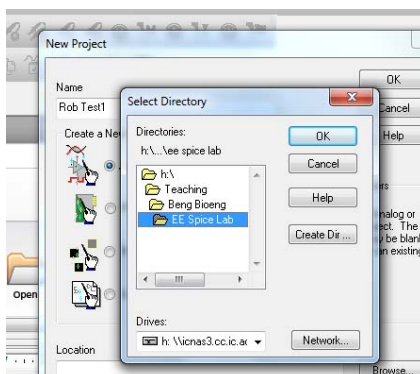


- I. Create a new Project document, and select it as an Analog or mixed signal A/D



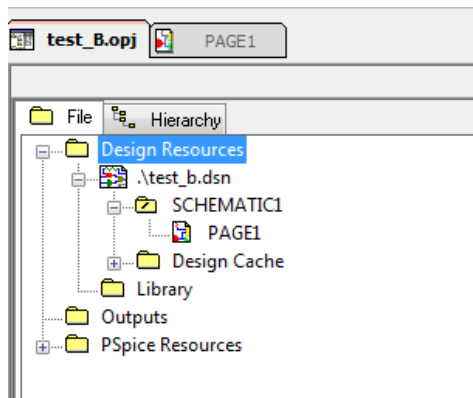
Give it a suitable name (no spaces in the name).

3. Create it as a blank project (not an existing project. You will also need to select a directory, using the Browse button, Choose a folder in your H: directory for this.



(You can now close the Start Page tab),


4. The folders tab should now look like:

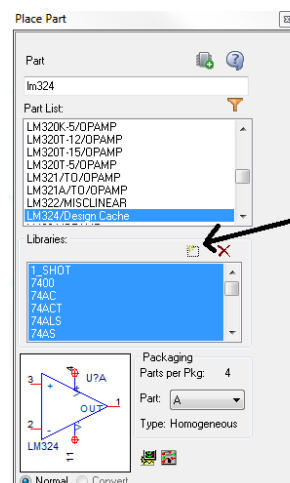


5. Click on the new PAGE1 icon to open a tab with that page (to start with it will be blank)
6. When there is a \*at the end of a tab or file name that denotes that it is a version that has not yet been save. Save your work regularly. You should have access to your college H: drive. Save files into the default directories chosen by OrCAD, **AND ALSO COPY TO YOUR H: drive (REGULARLY).**


### Step 2: Adding components:

- I. We can add components from the “Place>Pspice Component” menu on the task bar, and follow the dropdown menu. For basic things like resistors, capacitors and electrical sources, it is better to use this “PSPIICE Components” submenu,

2. For more advanced part selection use the icon  on the right hand side (can move menu if you want). In the “Place” menu select a component, either by finding the component from the library or by typing a prompt in the box above the component listing. Get used to doing this as it will save you a lot of time!). When using the “place parts” submenu for the first time, you may find that the correct libraries are not selected. There is a small icon to select which libraries to be used. It is probably best to highlight all the libraries in the PSpice sub-directories. You will find the op-amps in these libraries (In a complex project you may have separate library for all parts used, with version control and even an engineer with a “librarian” function).




- II. Double click on the component listing or click ‘ok’.

- III. Position the component on the schematic using the mouse and left-click to place. The cursor will keep the component until you hit the escape key or right-click, allowing you to place multiple copies of the same component.
- IV. To move a component, left click on it and then click on the hand symbol.
- V. To rotate a component left click on it, then right click in it and select rotate
- VI. Ensure that your circuit includes at least one ground terminal  or it won't simulate!

### Step 3: Assigning values to components:

- I. Right-click on the value next to the component of interest and select 'Edit Properties'
- II. Edit the numerical value to the relevant box including scaling factors (see Appendix D, e.g. k for kilo  $10^3$  ; MEG for  $10^6$ ) and WITHOUT units. (E.g.  $4 \times 10^{-3}$  V would be inputted as '4m'. Entering 4mV is likely to cause an error).
- III. Hit the enter key or click OK.

### Step 4: Wiring Up :

- I. Connect the components with the Wire tool  . (or "Place>wire").
- II. Simulate (will be explained below)

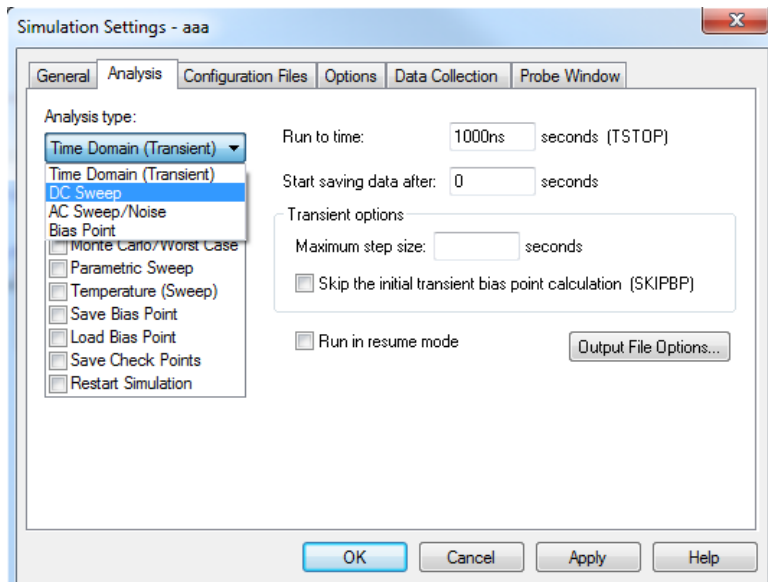
**NOTE:** files saved to the C:\ drive will be lost when you log off, if you want to keep it make sure you save to your network H:\ drive or flash memory while working in the computer lab.

### Step 5: Final Step :

Fill in the Title block, with things like date, name of schematic. Would be worth keeping version number, eg LP\_FILTER\_version\_a / b / c etc.

### 3. Simulating circuits using Pspice:

Before you can simulate your circuit, you have to choose what kind of simulation you want to run. In this lab you will use *transient* and *AC analysis*. With the schematic open, the first time, go to the PSPICE menu and choose NEW SIMULATION PROFILE. In the Name text box, type a descriptive name, e.g. "Scope". From the Inherit From List: select none and click "Create". This will open the simulation settings dialogue box:



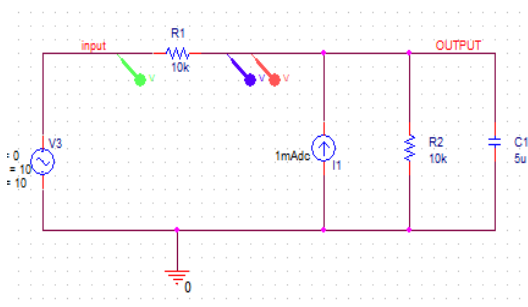
Use the analysis type drop-down menu to either select time domain (gives signals like those seen on a scope) or ac sweep (which gives signals like those seen on a spectrum analyser, or when the FFT button is pressed on the scopes – i.e. not time as x axis but frequency).


All future times you can use the "Edit Simulation Profile" item in the PSPICE menu.

#### iv. Transient Analysis

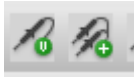
The main simulation method used will be *transient analysis*, which shows how the circuit will behave over a period of time and uses the Transient Tab above. The results of *transient analysis* are analogous to what you'd see if you had built the circuit and were testing with an oscilloscope probe. You must specify a stop time for the simulation, but you shouldn't need to worry about the other options for this type of analysis. If your curves look a bit jagged you may need to specify a maximum timestep but this won't usually be necessary. Note that choosing a stop-time of greater than a few seconds is likely to take a while to run and use a lot of processing power. Until you reach Exercise 9 (or unless otherwise instructed) a *stop time* (TSTOP) in the range of about 200ms should be sufficient to get detailed results.

Once you have entered a *stop time* (and maximum time-step, if required) click ok. Your circuit should now be ready to run.



Select *run*,  or use *Pspice/run* (or F12) from the top menu. This will open a new window in which the results will be displayed. Clicking back on the schematic page for the probe icons





will let you have a probe you can examine voltage (or current) If you click on a node (a wire connection the probe colour will change and a graph of that colour will appear in the results window), much like the trace you would see on an oscilloscope probe. You can click on multiple nodes to simultaneously display their values in the same results window. In the results window you can also “add trace” in the trace menu,

If you want to compare values that are orders of magnitude apart, it can be useful to add a further window of results. To do this, click onto the results window and in Plot menu select *add plot to window*. To select which trace you want in each plot plane simply click on the plot plane of interest and select the node from the schematic like usual.

A useful tool is the *cursor* function. This allows you to get accurate numerical readings from the graphical result plots. To do this; left click on the top menu Trace>Cursor . Now when you hover over the selected waveform and right-click the cursor should become a red cross hair.. Clicking and dragging left or right will shift the cursor along the waveform allowing a point in time to be chosen and the numerical result displayed. A new window will open up – with location of the cursor and the values at that point.

#### v. AC Sweep Analysis

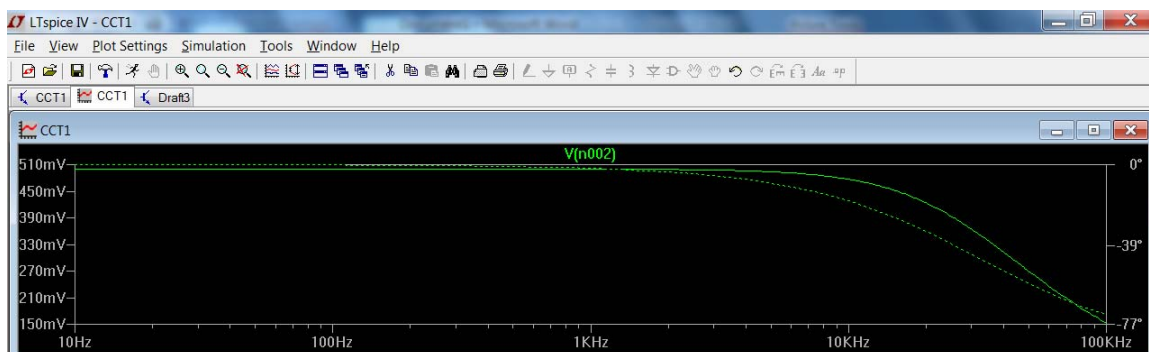
The AC analysis will apply a sinusoidal voltage whose frequency is swept over a specified range. Use the AC Analysis Tab on the Edit Simulation Command. The simulation calculates the corresponding voltage and current amplitude and phases for each frequency. When the input amplitude is set to 1V, then the output voltage is basically the transfer function.

Unless otherwise stated, use the following as standard values:

Type of Sweep	Decade
Points per decade	15
Start frequency	0.1 (Hz)
Stop frequency	10k (Hz)


In order to run an AC sweep analysis of your circuit you need to include in your circuit a voltage/current source set to generate AC. The *signal* component can be used for this purpose by ntering an AC amplitude (use 1V unless otherwise stated).

You can also add a generic AC input from  
Place>PSpice\_Component>source>Voltage\_source>AC.



#### 4. Section One: Simple circuits using spice

##### vi. Exercise 1:

Open a new schematic document in LTspice and enter the circuit shown in fig 1 below, you will need to add a DC (source) battery from the component list. What current do you expect through R1. You can check this by clicking the  button. You can also display the DC voltage.

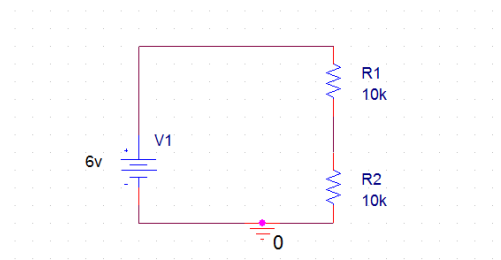


Fig 1 - Circuit for exercise 1

##### vii. Exercise 2:

Replace the battery in Fig 1 with the component called PSICE Component>source>Voltage Sources>Sine. . Right click to edit and enter values to generate a sinusoidal signal with an amplitude of 10V, a frequency of 10Hz and a 0V DC offset (VOFF=0, AC=0). Insert Probes as shown. Run a transient analysis with a stop time of 200ms and record the results.

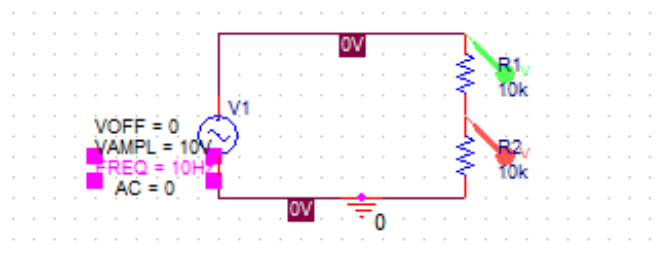


Fig 2a – Circuit for exercise 2

The combination of R1 and R2 halves the voltage from V1, and is called a potential divider. Change R1 to 30k. What do expect the voltage amplitude across R2 to be? Run a simulation and check if you were correct.

Often we want to quickly adjust the values of the potential divider, this is done with a component called a potentiometer which has a shaft that can be rotated.. For more details on this component see Appendix D

Create the following circuit with a potentiometer (selected from Passive components). When placing the potentiometer you will need to rotate and then mirror it to ensure that its pin 1 is at the top.

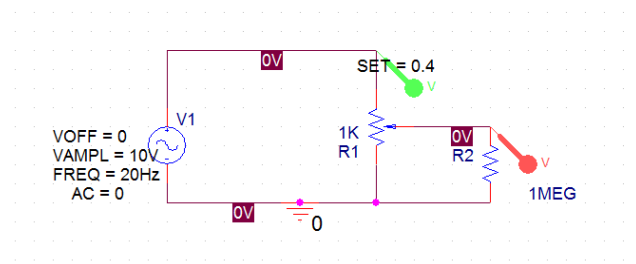


Fig 3b – Circuit for exercise 2

The potentiometer has a parameter ‘SET’ that determines the split of the two resistances either side of the central wiper, these two resistors always add up the potentiometer value, in this case 1k $\Omega$ . Run some simulations and see how the amplitude of the voltage across R2 varies with the SET value.

#### viii. Exercise 3: Frequency Response.

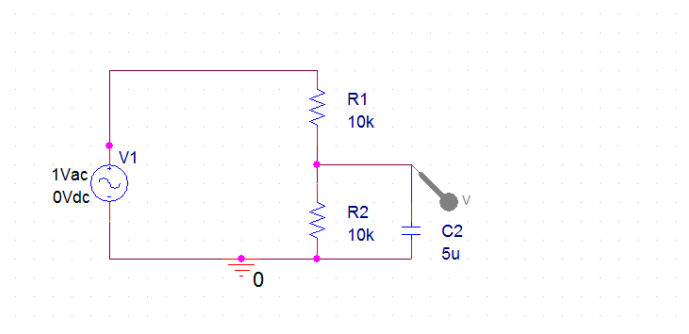
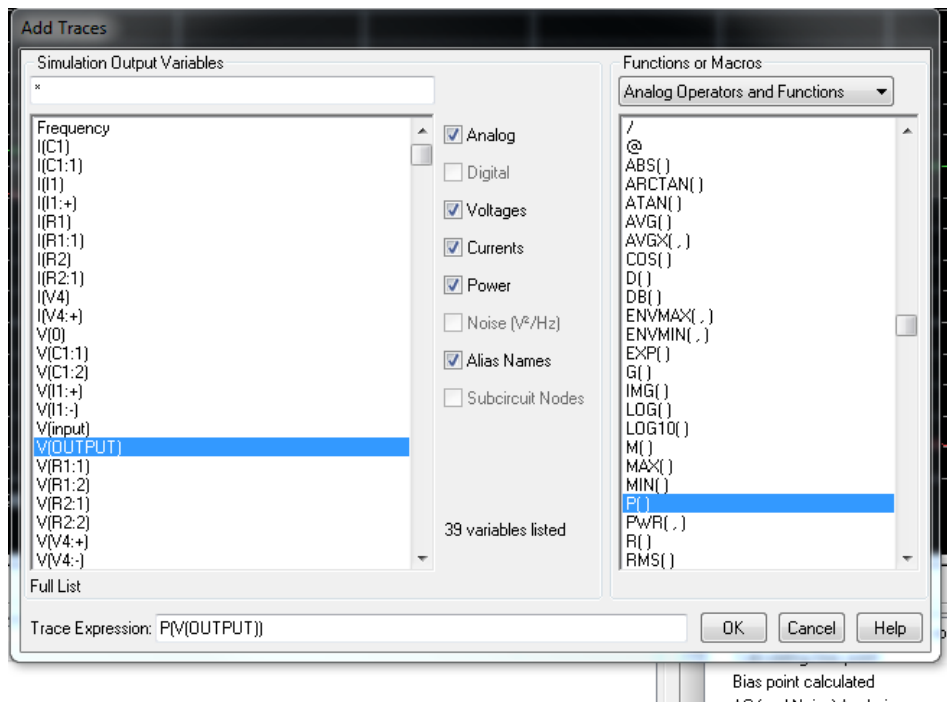


Fig 3 – Circuit for exercise 3

Simulate the frequency response of the circuit shown in fig 3. To use “ac sweep” the source needs to be called PSPICE Component>source>Voltage Sources>AC. This is an AC generator with no given frequency – it will be swept. Typically there will only be one such generator in a circuit (perhaps on the input, and the behaviour of the circuit as the input frequency is swept is measured).

Run an AC analysis using the standard values (15 points per decade, range 0.1-10,000 Hz). Record the resulting output amplitudes and phase shifts taken at node A.

By default PSPICE only shows the magnitude of the voltage. To select showing the phase as well (i.e. as a BODE plot) you need to select function and P[ ] as below to show the phase of selected voltage.



You can also display the magnitude of the Voltage as a logarithmic function by editing the Plot>Axis Setting menu on the plot window. Try this and notice how the frequency response is a straight line on the resultant log-log plot.

## 5. Section 2: Op-Amp circuit simulations

In this section you will be simulating circuits containing operational amplifiers (op-amps). They play a key role in many medical devices and will form the backbone of your heart-rate monitor.

For many general purposes, an op-amp modelled as a linear amplifier can give reasonable 'ball park' figures during component design. However, realistic simulation of the actual behaviour is required for most applications to accurately measure the response of the circuit. PSPICE does come with an idealised linear op-amp macromodel but you'll be using the component LM324 which is the op-amp that you will be using during the build stages - so you should find your results correlate closely to what you see during part II of this lab.

The op-amps you'll be using require a  $\pm 15\text{V}$  DC voltage supply, and this must be included in your SPICE schematics. The PSpice Component>Source>VoltageSource>DC component is probably the best to use as it takes up the least space. Make sure that you have the polarity of the op-amp matched with your voltage supplies. (I.e. +15V goes to the positive supply terminal, and -15V is connected to the -15V supply terminal). A common mistake is to mirror the op-amp vertically but forget to switch the voltage supplies to match the transformation. If you get them round the wrong way in SPICE you'll get incorrect output waveforms, get it wrong when you're building them for real and you risk destroying your integrated chips containing the op-amps!

### ix. Exercise 4:

Implement the non-inverting amplifier circuit shown in fig 4. Apply a sinusoidal voltage with a 1mV amplitude. Record the output waveform. The amplification factor is the ratio of amplitudes of output/input. Change the values of resistance and verify that your output

follows an amplification factor of  $\frac{V_{out}}{V_{in}} = 1 + \frac{R_1}{R_2}$ .

Can you think of a way of adding a potential divider of making the gain less than 1?  
(Hint : think of the three terminals of a potential divider and the voltages across them)

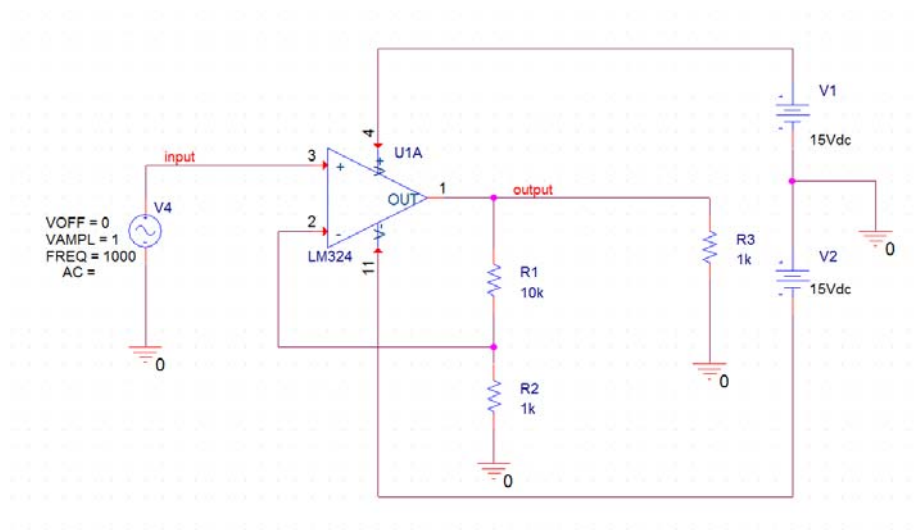


Fig 4 - circuit for exercise 4

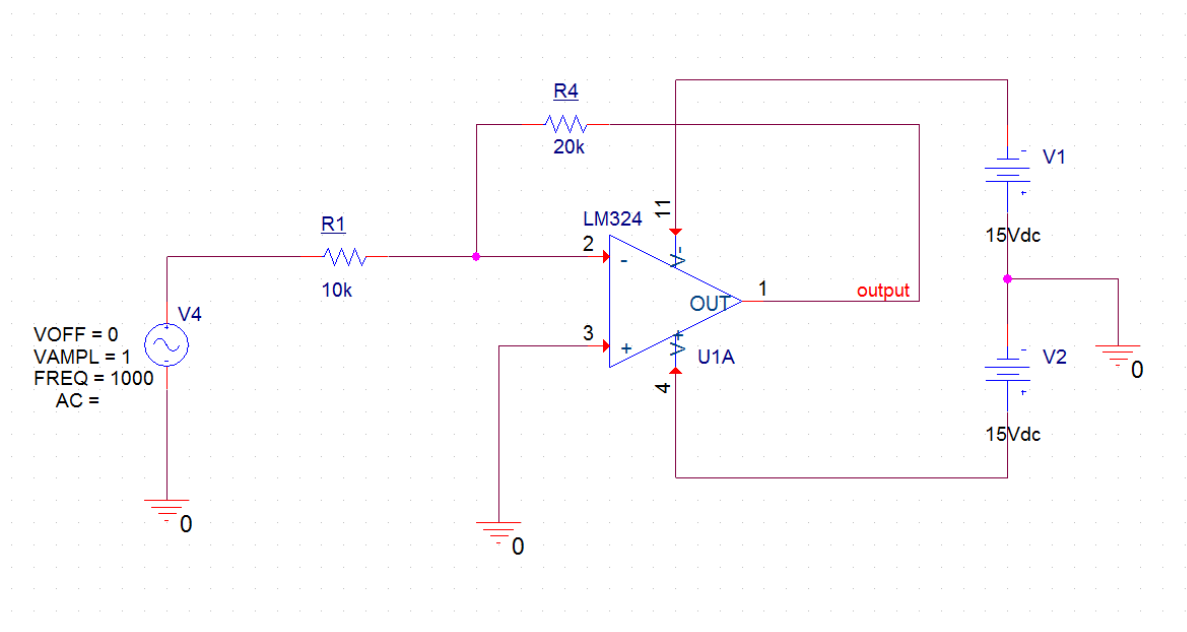
*NOTE: The component “battery” has been used to supply the  $\pm 15\text{V}$  DC supply needed for the op-amp to operate. There are many possible ways of implementing this, use whichever you prefer as long as you have  $+15\text{V}$  supplied to the positive supply input, and  $-15\text{V}$  supplied to the negative supply input. Make sure that you don't have a connection between the grounded supply voltage node and the output of the op-amp.*

*NOTE: Wires laid over the terminals of a component will always form a connection.*

#### x. Exercise 5:

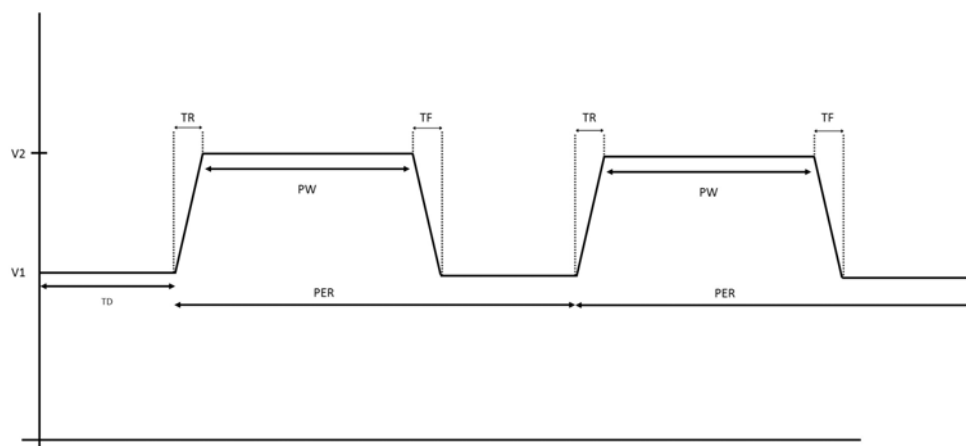
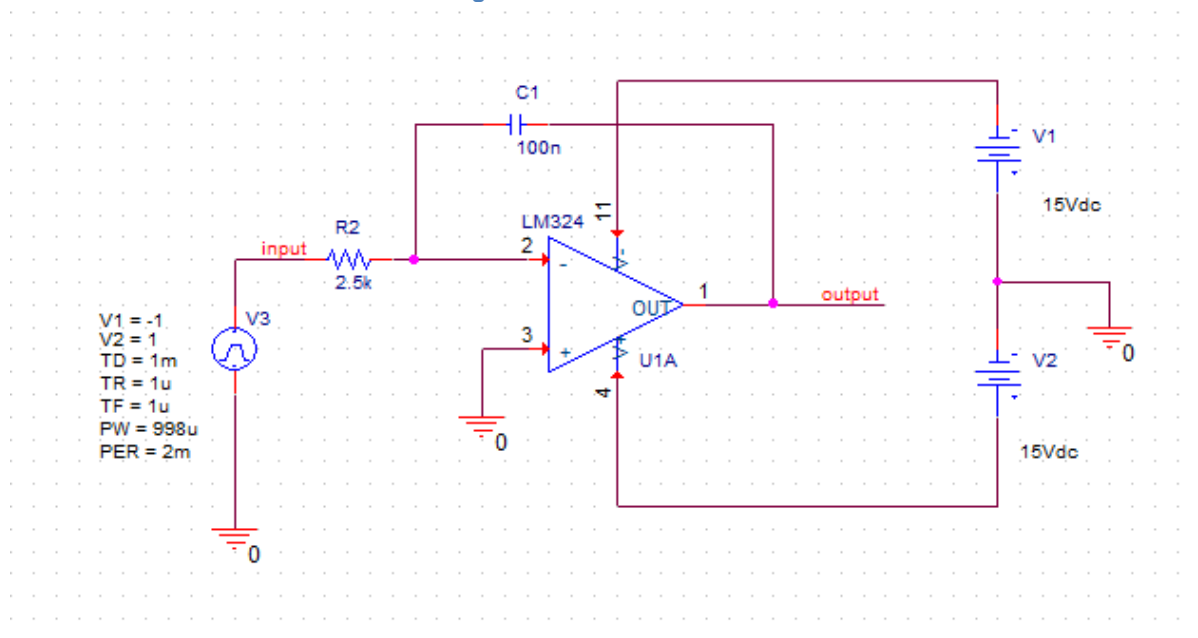
Implement the inverting amplifier shown in fig 5. Apply a  $1\text{mV}$  sinusoidal input and record the output waveform. Vary the values of the resistors to deduce the relationship governing the amplification factor (gain). Can you vary the gain by varying just one resistor? Can you make the (modulus of the) gain less than 1 so the output amplitude is less than the input amplitude?

Fig 5 - circuit for exercise 5



**xi. Exercise 6:**

Implement the integrator circuit shown in fig 6. For this circuit you're going to use *signal* to generate a square wave input. Instead of *sine* you'll need to select *PULSE* under the 'functions' section. Now choose values for V1 (the initial voltage), V2 (the pulse amplitude), Tdelay, Trise, Tfall, PW (pulse width) and Tperiod (see figure below) so that the input is a 500Hz, +/- 1V square waveform. To do this sketch out a square wave, and identify which values of Figure 6b correspond to the parameters of the square wave. You should also have a short period of zero Voltage before the square wave starts. The rise and fall times should be small but finite (e.g. 1 $\mu$ s). Try to get a regular triangular output waveform by adjusting your input signal. Record your results.

**Fig 6a - circuit for exercise 6****Figure 6B: Defining a Pulse**

## 6. Filters:

A *filter* is usually a circuit that is designed to pass a specified band of frequencies while attenuating all signals outside this band. Filters are commonly constructed using op-amps. The filters that we'll be exploring are low and high pass Butterworth filters with a roll-off of 40dB/decade. The circuits described below will be met in your course material, both this year and in later years. They are very widely used in all forms of medical instrumentation involving the recording of physiological signals, as well as medical imaging instrumentation, mobile phones, hi-fi circuitry, MP3 player etc.

*Note: Roll off means that once you reach the cut-off frequency of the filter, the output voltage drops by a factor of 100 (40 dB) for every tenfold increase (decrease) in frequency. The roll-off can be thought of in terms of the gradient (or slope) of the cut-off region in the filter's frequency response curve.*



**xii. Exercise 7:**

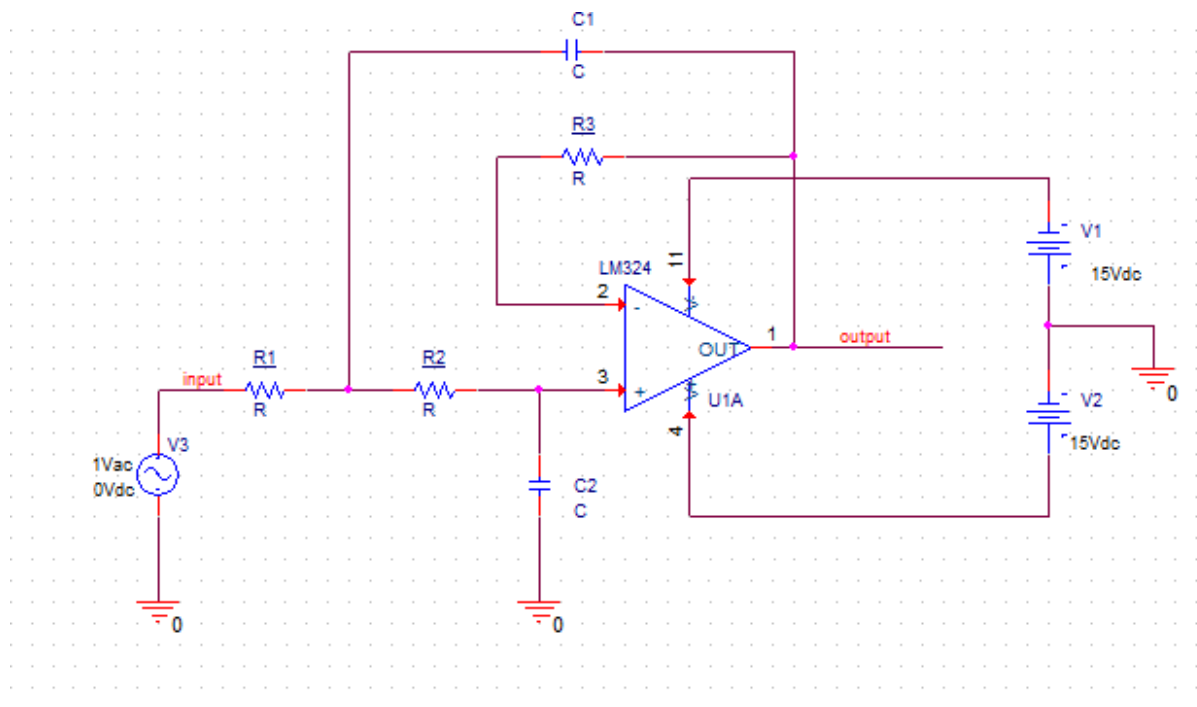
Implement the low pass Butterworth filter shown in fig 7 so that it has a 1 kHz cut-off frequency. The values of some of the components have been omitted so you'll have to calculate their values using the formulas given below. This circuit uses the signal input set to an *AC amplitude* of 1V, a phase of zero and (*none*) selected under *Functions*. Simulate an AC sweep and record the resulting frequency response.

$$R_1 = R_2 = \frac{1}{2\pi \cdot f_c \cdot C_1}$$

Choose a value of  $C_1$  in the range 100 pF – 0.1  $\mu$ F.

Make  $C_2 = C_1$  and set  $R_3 = 2R_1$

Fig 7 - Low pass filter for exercise 7



**xiii. Exercise 8:**

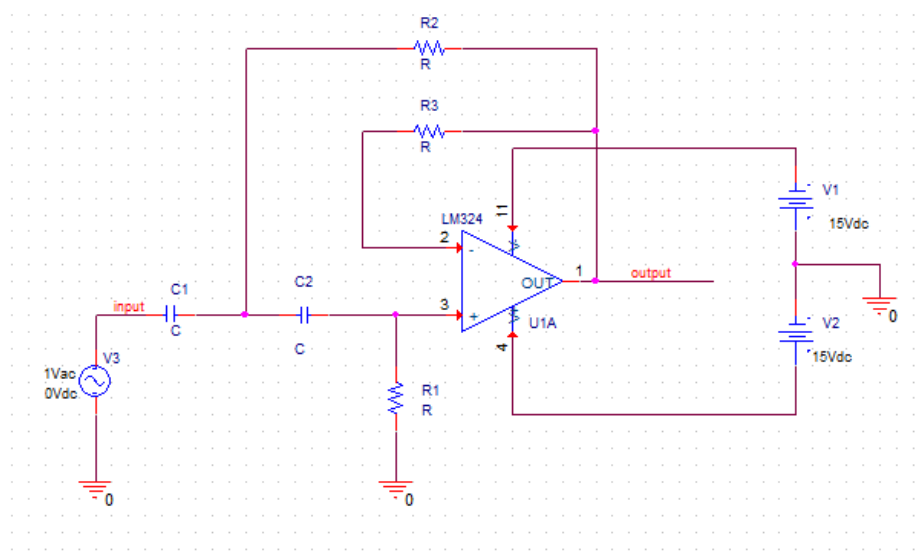
Implement the high pass filter detailed in fig 8 with a 1 kHz cut-off frequency, and simulate in the same way as the low pass filter. Plot the frequency response. Component values should be calculated using the following rules:

$$R_1 = R_2 = \frac{1}{2\pi \cdot f_c \cdot C_1}$$

$$R_3 = \frac{1}{2} R_1$$

Choose a value of  $C_1$  in the range 100 pF – 0.1  $\mu$ F. Make  $C_2 = C_1$ .

**Fig 8 - High pass filter**



## 7. Section 3: Electronic Stethoscope design:

### xiv. Exercise 9:

Op-amps are commonly used to amplify physiological signals such as the electrocardiogram, or ECG, or in this lab you will be using the heart sounds from a microphone. Your task is to design a circuit that first acts as an electronic stethoscope, taking an input heart sound signal from a microphone and producing a signal that can drive a set of headphones. (The signal can then be processed to produce a clean digitised 2V P-P output signal that can be passed into a PIC microprocessor for display on a dual 7-segment LED output to measure the heart-rate.)

*NOTE: The report for this module is focussed on the electronic stethoscope, both design and build. Your design stages will form a key part of this so make sure you record what you're doing for future reference.*

Once your design is complete you will be heading over to the lab to build these for real. Be warned, implementing these circuits on breadboards is surprisingly fiddly, so try to avoid overcomplicating your designs where possible.

*NOTE: Your build time in the lab is limited, making sure that you've tested each part of your design before you start building will save a lot of time and frustration. Remember that only certain component values are available (called the E12 or E24 series [http://baec.tripod.com/resistor\\_prefered\\_values.htm](http://baec.tripod.com/resistor_prefered_values.htm) )*

**The requirements are as follows.**

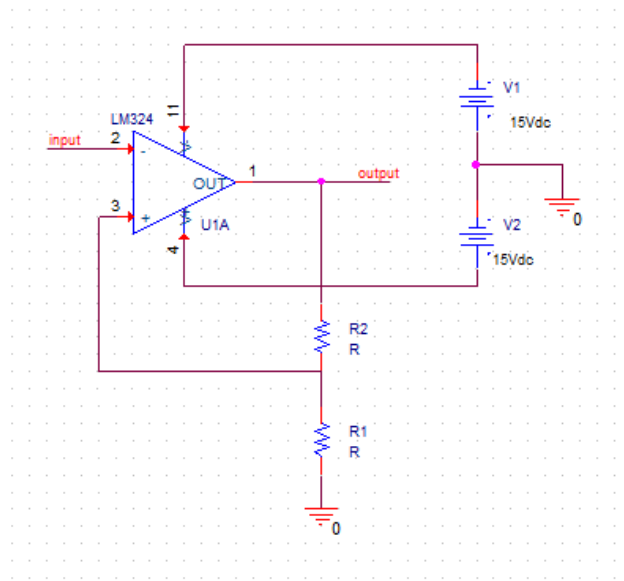
- 1) The microphone signal has a voltage amplitude of approximately 1-5 mV, and to drive a headphone an amplified signal of 2-3 V is required. Use this to specify the gain required from amplification stages.
- 2) The maximum gain for an individual op-amp stage is 200, higher gains can be unstable, and so to have a larger gain will require more than one stage (i.e. more than one op-amp).
- 3) As the signal is small, it will be necessary to filter out noise. What frequencies will the noise be at? How does this compare to the signal frequency (which is an acoustic signal)? How can you filter out the noise? (HINT: you measured one source of noise in last terms oscilloscope practical).
- 4) The high gain required means that a small DC offset towards the input of the circuit can be amplified to saturate the signal, and must be taken into account. Think of ways you can remove a dc offset and see how easy it would be to implement a circuit to do this.
- 5) You will use the output to drive a set of headphones to listen to the signal but you do not need to simulate this bit.
- 6) You can also use the output to eventually trigger a rate counter or flash an LED. For this you will need a positive going pulse that goes from 0v (<0.2V) to 5V (>3.5V). If you have time you could also simulate this using the circuit on the next page..

A straight forward method of converting your signal into a clean square wave is to include a Schmitt trigger in your circuit. NOTE: You can adjust the sensitivity of the Schmitt trigger easily by changing the values of the two main resistors. The resistors form a potential divider giving rise to the following relationship for the threshold voltage: (See Fig 9 for resistor numbering)

$$V_{\text{threshold}} = \left( \frac{R_2}{R_1 + R_2} \right) \cdot V_{\text{saturation}}$$

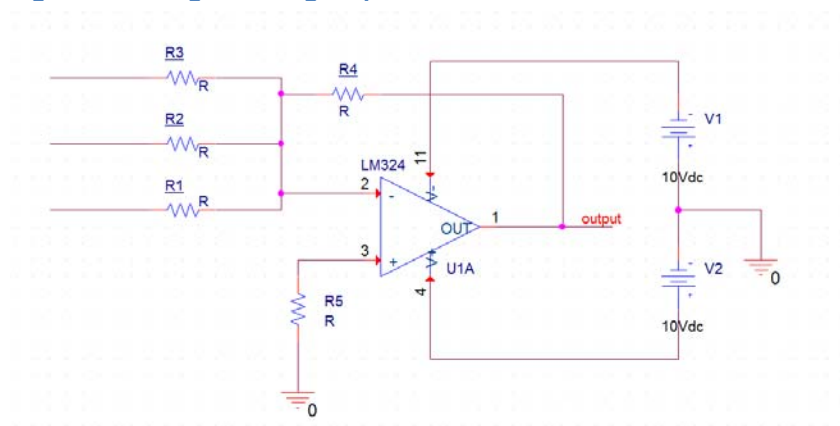
Where  $V_{\text{saturation}}$  is the saturation of the op-amp, typically in the region of around 13V for LM324/LT1464 with a  $\pm 15V$  DC voltage supply.

Fig 9 - Schmitt Trigger



Another useful addition to your circuit is the inverting summing-amplifier which can be easily used to add a DC bias to your signal. This works with any number of input voltages, the example shown is for 3 inputs.

Fig 10 - Inverting Summing Amplifier



The inverting summing amplifier follows the following relationship:

$$V_{\text{out}} = -R_4 \cdot \left( \frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3} \right) \quad R_5 = R_4$$

## 8. Appendix A: Troubleshooting

- Check that you've included a **ground** somewhere in your circuit
- Double check that you've specified a sensible **stop time** (in the order of microseconds or milliseconds for most circuits, no more than around 5 seconds when testing your pulse monitor)
- Check that all components have been **assigned a value** by right clicking and filling in attributes
- Ensure that you **have not included units** when you've entered component attributes. E.g. 5 $\mu$ F for a capacitor would be entered as '5u' or '5U'
- If the above have all been checked, there is likely to be an error with the **wiring** arrangement of your circuit. Check for unwanted connections where wires cross, and that all components have been connected to the circuit. It is possible to overshoot the wire across components leading to simulation problems; if in doubt zoom in a bit and check for any extra wiring encroaching on your component symbols. You can set your wiring to a more contrasting colour using drafting options detailed in A1.2 below.
- Resistor values used with Op-amps should be in the region 300 $\Omega$ - 300k $\Omega$ , any lower or higher and problems can arise.

## 9. Appendix B - Decibel

The decibel (dB) is a relative measure of power and is defined by the following equation

$$dB = 10 \log_{10} \left( \frac{P_1}{P_2} \right)$$

The following conversion applies.

dB	Amplitude Ratio	Power Ratio
20	10	100
10	3.2	10
6	2	4
3	1.4	2
0	1	1
-10	0.32	0.1

It is used in both engineering and bio-engineering as:

- It can conveniently describe a wide range of values
- When used to characterise the gain of an individual amplifier stage the respective gains (in dB's) can be added to give the overall gain.
- The physiological response of many senses has a logarithmic nature, such as hearing, where an increase in acoustic power from 10 to 100 units, sounds the same as an increase from 1000 to 10000.

Although dB is strictly a relative measure based on a ratio, in some applications it is used relative to a reference power to give an absolute unit of power. Eg in some electronic terminology  $P_{ref}$  is defined arbitrarily as 1mW, and all other power are defined in dB's relative to this. Hence:

$$dB = 10 \log_{10} \left( \frac{P_1}{P_{ref}} \right)$$

$$dB_{mW} = 10 \log_{10} \left( \frac{P_1}{1mW} \right)$$

In acoustics the reference power is defined as the threshold of human hearing which is agreed to be 20 microPascal. Hence 50 dB is quiet conversation, 90 dB traffic, 130 dB is threshold of pain.

Although the decibel is defined in terms of Power it is often used with an amplitude (A) measure such as pressure, voltage, etc. Then we assume that  $P \propto A^2$

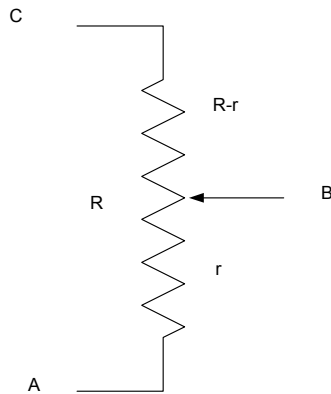
$$dB = 20 \log_{10} \left( \frac{A_1}{A_2} \right)$$

## 10. Appendix C Scaling Factors

Scaling factor	Power of ten ( $10^{\#}$ )	LTSPICE Suffix
<b>Terra</b>	12	<b>T or t</b>
<b>Giga</b>	9	<b>G or g</b>
<b>Mega</b>	6	<b>MEG or meg</b>
<b>Kilo</b>	3	<b>K or k</b>
<b>Milli</b>	-3	<b>M or m</b>
<b>Micro</b>	-6	<b>U or u</b>
<b>Nano</b>	-9	<b>N or n</b>
<b>Pico</b>	-12	<b>P or p</b>
<b>Femto</b>	-15	<b>F or f</b>

## 11. Appendix D Potentiometer

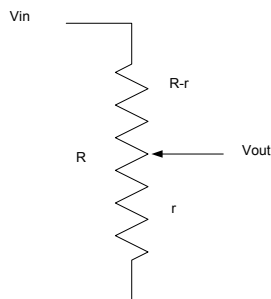
A potentiometer is a variable resistor where a moveable contact can give a resistance value between the 0 and the full value of the resistor. The contact can be moved by turning a shaft or knob, or a screw head.



The resistance  $R$  between A and C is constant and depends on the Part. The resistance  $r$  between B and A can be varied from 0 to  $R$ , The resistance between B and C is  $R-r$

This component can be used as a variable resistor by connecting between A and B.

Another use is as a variable potential divider as in the circuit below.



Here if  $r$  is set to be  $f.R$  then  $V_{out} = f.V_{in}$  (where  $f$  is fraction between 0 and 1). This arrangement is often used for volume controls on audio equipment, radios etc.

For applications that require adjustment during assembly but not afterwards we use a trimmer potentiometer as shown on the picture bottom right, this must be adjusted

with a screwdriver.

Examples of potentiometers

