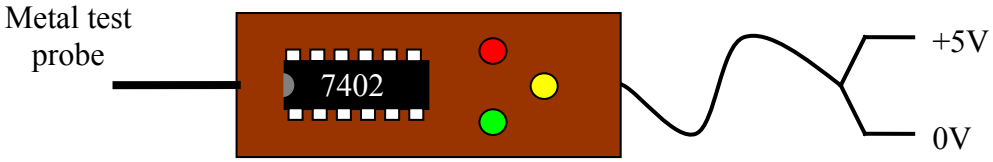
DCD: Logic Probe Laboratory 06

The aim of this lab is to demonstrate how logic gates can be combined with analogue components to produce more complex functional blocks. These circuits mix both digital and analogue devices, therefore the SPICE simulation tool will be used to design and test each circuit. These circuits are then combined to produce a logic probe, an item of test equipment that that can be used to trace down faults within a digital circuit. A logic probe allows you to quickly test the state of a wire using three LEDs i.e. test if it is a logic '1' (red LED), '0' (green LED) or is pulsing (yellow LED), as shown in figure 1. To construct this device a circuit board must be designed to physically mount (hold) these components and provide the required wired links. At the end of this practical you will understand how to:

- Simulate analogue and digital circuits using SPICE
- Design a Vero-board circuit board layout
- Solder components onto a circuit board.



Logic probe

LED display			
Red	Green	Yellow	State
○	●	○	Logic '0'
●	○	○	Logic '1'
○	○	○	Invalid state / open circuit
●	●	●	Pulsing < 1MHz (approx)
●	○	●	Pulsing > 1MHz (approx)

Figure 1: Logic probe

Each circuit will be simulated using LTspice to determine what analogue component values are required. This software can be downloaded for **home** use from:

http://www.linear.com/designtools/software/

To start this practical click on the start button and select LTspice.

```
Start -> All Programs -> Hardware development -> LTspice
```

The logic probe we will be making is a modified version of one originally designed by Andy Collinson:

http://www.zen22142.zen.co.uk/Circuits/Testgear/lprobe.html

When designing this logic probe one of the main aims was to minimise the number of components used, reducing costs and the physical size. Therefore, all functionality must be implemented using a single 7402 NOR gate.

The complete circuit diagram of this logic probe is shown in Appendix A, within this there are three functional blocks:

- Edge detector: generates a short pulse on each falling edge i.e. an input transition from a logic '1' to a logic '0'.
- Astable multivibrator: a square wave generator i.e. an oscillator.
- Monostable multivibrator: a one shot pulse generator i.e. when triggered this circuit produces an output pulse of a specific width (on time).

Edge Detector

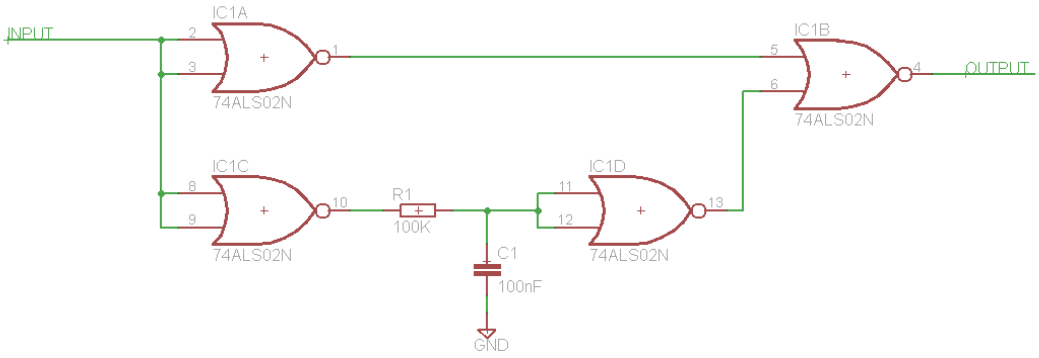


Figure 2: Edge detector

For the OUTPUT signal i.e. the output of IC1B in figure 2, to produce a logic '1' both inputs must be zero. From a logic point of view this can never occur as the lower input (pin 6) has two NOT gates (cancel each other out) and the top input (pin 5) has one i.e. the inputs should always be different. However, the input to IC1D is driven by a CR circuit, this delays changes on INPUT (exponential voltage rise time), allowing the "00" state to occur on the inputs of IC1B, as shown in figure 3.

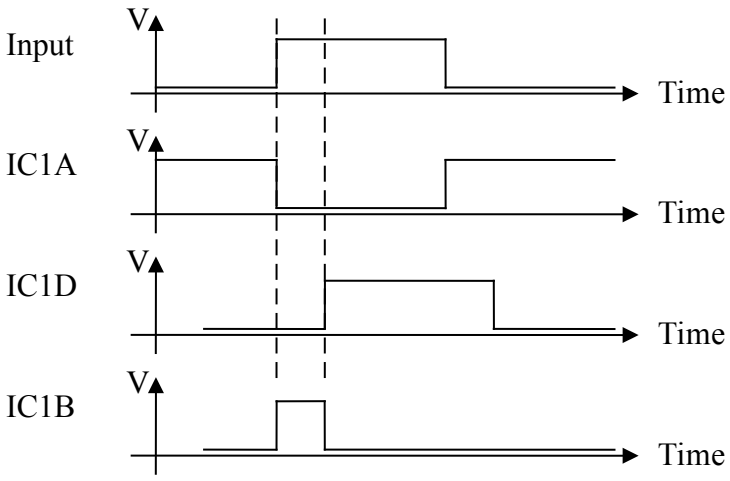


Figure 3: Edge detector input / output waveform diagram

To add a NOR gate to the LTSpice simulation model left click on the  icon, or click on the pull down:

```
Edit -> Component
```

This will open the Select Component Symbol window, select the OR component and place four OR gates as shown in figure 4. Repeat this process adding the capacitor (CAP), resistor (RES) and voltage source (VOLTAGE) components.

Note: these are in the top level symbol directory, to return to this you will need to double left click on the [..] symbol.

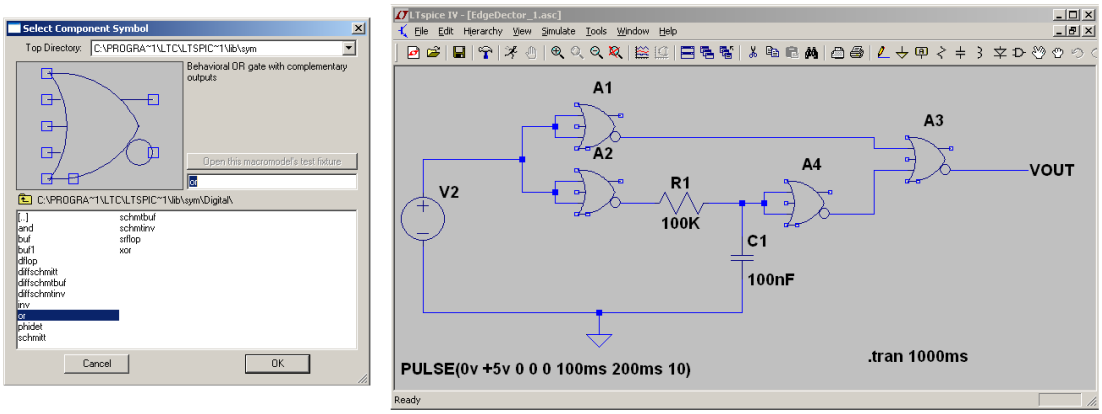


Figure 4: Edge detector circuit diagram

To wire up these components left click on the  icon, or click on the pull down:

```
Edit -> Draw wire
```

The cursor will be replaced with cross hairs. To start a wire left click on an input or output, 90 degree bends can be placed when routing the wire by left clicking. To finish a wire again left click on an input or output. A wire can be deleted by pressing the **DEL** key. The cursor will be replaced with a pair of scissors, left click on the wire segment you wish to delete. To return to the default cursor press the **ESC** key.

Each component's parameters can be entered by right clicking on its symbol. Update each component to match figures 4 and 5.

Note, for the voltage source you will need to click on the advanced options button.

To simulate this circuit i.e. to plot each voltage waveform left click on the pull down:

```
Simulate -> Edit Simulation Cmd
```

Next, enter the required simulation time i.e. Stop Time of 1000 ms (milli-seconds), as shown in figure 6. To generate the waveform diagrams click on the  icon, or click on the pull down:

```
Simulate -> Run
```

Right click on the blank waveform trace and select “Add Plot Plane”. This will allow the input and output Voltages to be displayed on separate traces.

Next, click on the schematic (circuit diagram). Move the cursor onto the VOUT wire, as you move the cursor across the components the cursor will change to a **red** test probe, click on the wire to add this waveform to the plot plane. Left click on the empty plot plane and repeat for the input signal. The final simulation plot is shown in figure 6

To magnify a section of the waveform left click on the  icon. Next left click, hold and drag a box around the area of the waveform you wish to view in more detail. To return to the original view left click on the  icon.

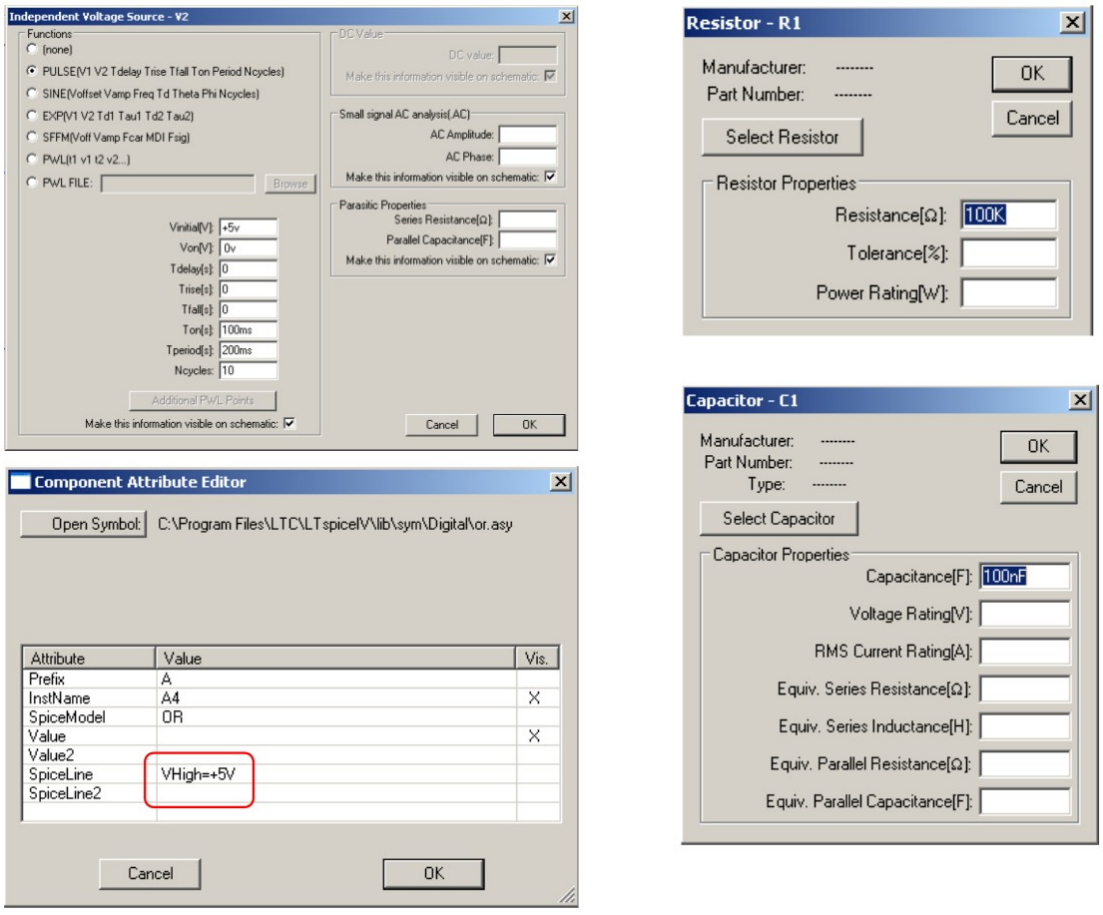


Figure 5: component attributes

Task 1

Using the simulator, measure the voltage across the capacitor. What are the voltage levels that define a logic ‘1’ and a logic ‘0’? What will happen to the output voltage if the value of the resistor is increased to 200K ohms?

Task 2

The edge detector shown in figure 4 requires four logic gates, therefore a simpler circuit has to be used, as shown in figure 7.

Note, +5V is a voltage source, as shown in figure 8, do **NOT** add the diode at this stage. Build a simulation model of the circuit shown in figure 7. Does it perform the same function?

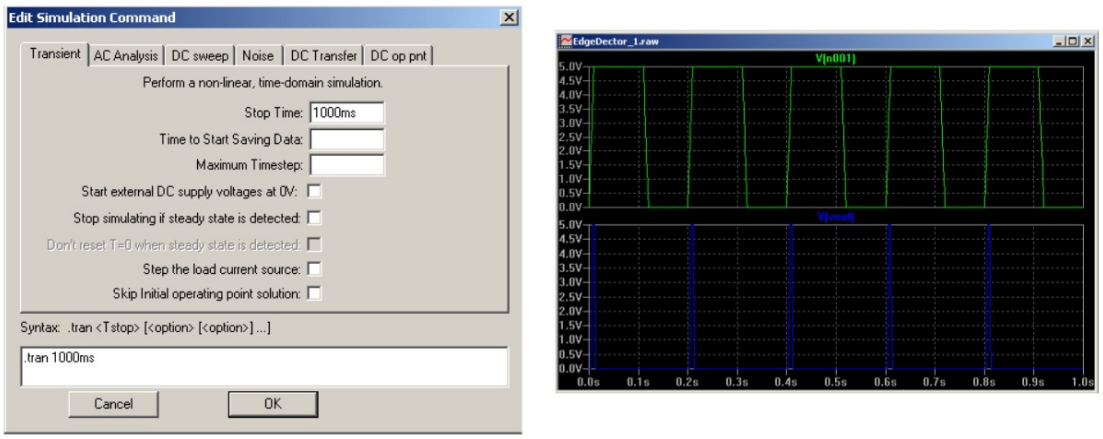


Figure 6: Simulation

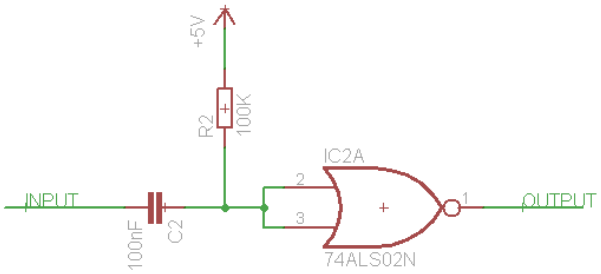


Figure 7: Simple edge detector

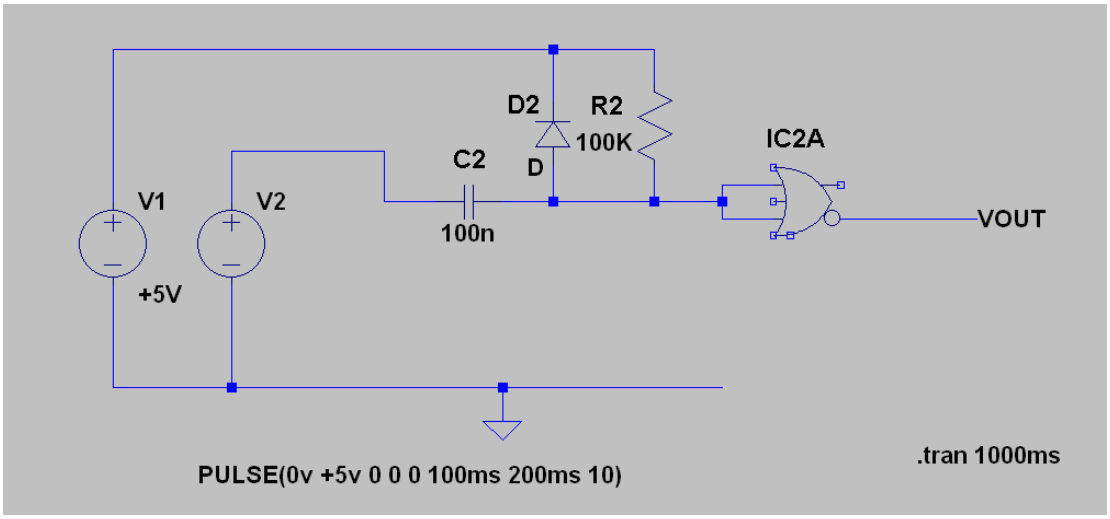


Figure 8: Simple edge detector

Task 3

Examine the voltage on the input to the NOR gate. Why could this damage the IC? To protect this logic gate connect a diode in parallel with the resistor i.e. Anode connected to the logic gate's input and Cathode to +5V, as shown in figure 8. Re-run the simulation, examine the voltage on the input to the NOR gate. How has it changed?

Task 4

What will happen to the output voltage if the value of the capacitor is reduced to 3.3nF? Re-run the simulation, examine the voltage on the input to the NOR gate. How has it changed? What will now happen if the input pulse period is reduced:

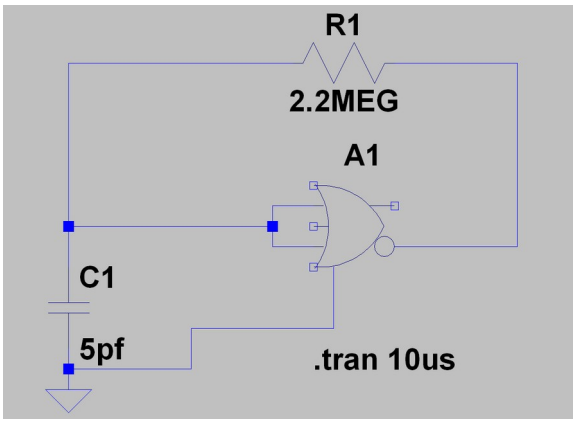
Ton = 100ns
Tperiod = 200ns

Re-run the simulation for **1us**, examine the voltage on the input to the NOR gate. How has it changed? What value of capacitor should be used in the logic probe?

Astable Multivibrator

Create a new schematic for figure 9. This circuit uses a NOR gate with both inputs wired together i.e. acting as an inverter. Feedback via a 2M2 resistor connects its output back to its input forming an oscillator. The frequency of the resulting 'square' wave is proportional to the RC time constant of R1 and C1 i.e. the time it takes for the capacitor to charge up (or discharge) to the voltage threshold that define the opposite logic state.

Note, in the final circuit the capacitor C1 is not a separate component, but the internal input capacitance of the NOR gate.



Attribute	Value	Vis.
Prefix	A	
InstName	A1	X
SpiceModel	OR	
Value		
Value2		
SpiceLine	VHigh=+5V	
SpiceLine2	Trise=7ns Td=11ns	

Figure 9: Astable multivibrator

A simulation is only as good as its model, therefore, the parameters of the NOR gate must be updated using the datasheet for the 74HCT02 NOR gate, as shown in figure 9 and 10 i.e. the propagation delay and rise / fall times.

Task 5

Enter the Trise and Td values and run the simulation, examine the voltage on the input of the NOR gate. As this model is oscillating at a high frequency the simulation model takes significantly longer to run. Therefore, press the **ESC** key at any point to stop the simulation. What is the frequency of the output waveform? How would this change if the value of R is decreased or increased?

Symbol	Parameter	Conditions	25 °C			−40 °C to +125 °C		Unit
			Min	Typ	Max	Max (85 °C)	Max (125 °C)	
74HCT02								
t_{pd}	propagation delay	nA, nB to nY;						
		V_{CC} = 4.5 V	-	11	19	24	29	ns
		V_{CC} = 5.0 V; C_L = 15 pF	-	9	-	-	-	ns
t_t	transition time	V_{CC} = 4.5 V;	-	7	15	19	22	ns

Figure 10: NOR gate parameters

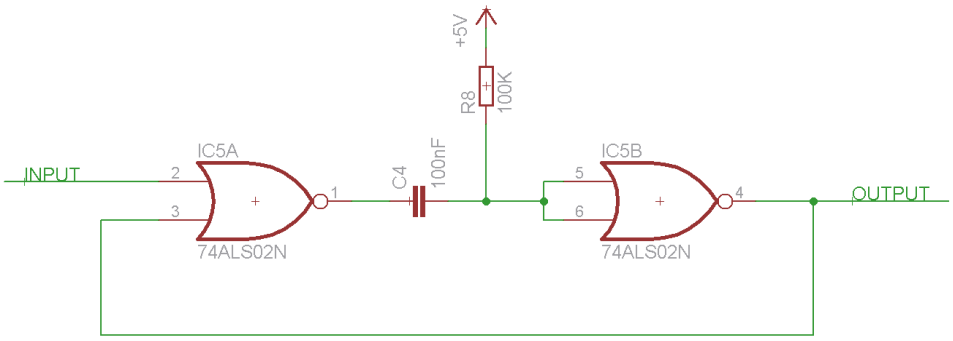


Figure 11: Monostable multivibrator

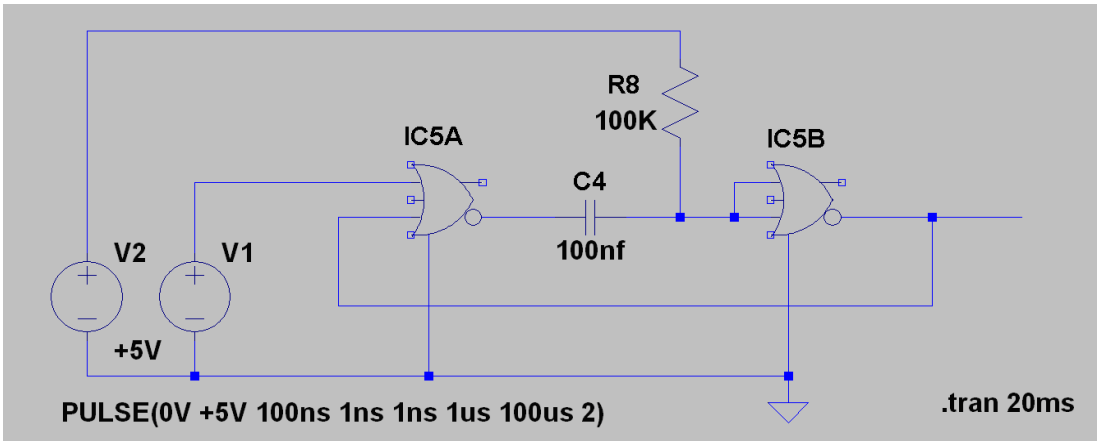


Figure 12: Monostable multivibrator SPICE simulation model

Monostable Multivibrator

This circuit acts as a pulse extender i.e. a short input pulse will generate a longer output pulse. Create a new schematic for figure 11. This circuit uses two NOR gates, IC5B has both its inputs wired together i.e. an inverter. Initially, INPUT and the output of IC5B are assumed to be a logic '0'. Therefore, there is zero volts across

capacitor C4 i.e. the pull up resistor R8 ensures that both sides are +5V. When a positive going pulse is applied to INPUT, the output of IC5A will go to a logic '0' discharging C4. This logic '0' causes the output of IC5B to go high, ensuring the output of IC5A remains in a logic '0' state i.e. the signal INPUT can now be set low. The voltage on capacitor C4 will now begin to charge to +5V via R8. When this voltage reaches the threshold that defines a logic '1' the output of IC5B will go to a logic '1', resetting the circuit ready for another input pulse.

Task 6

Configure voltage source V1 as a pulse generator, producing a 1us 'start' pulse, as shown in figure 12. Run this simulation, what is the duration of the output pulse? If multiple 'start' pulses occur during this time, will these re-start the monostable multivibrator? What value of resistor is required to produce an output pulse of a 150ms? Measure the voltage on the input of the second NOR gate i.e. the logic gate that produces the output pulse. What additional component should be used to protect this device?

Logic Probe

The three functional blocks designed i.e. edge detector, astable and monostable multivibrator can now be combined to produce a logic probe.

Task 7

The astable multibribrator is used to drive the logic '0' and logic '1' output LEDs on the logic probe as shown in figure 13. How does this circuit work? Why does it enable the invalid state / open circuit state to be displayed i.e. when the input is not connected to either a logical '1' or '0' (floating) neither LED will be illuminated?

Hint: if the oscillator output has a 50 – 50 duty cycle i.e. is on (+5V) and off (0V) the same period of time, what will the average output voltage be? How does this compare to the voltage created by the potential divider circuit, R6 and R7 shown in figure 13?

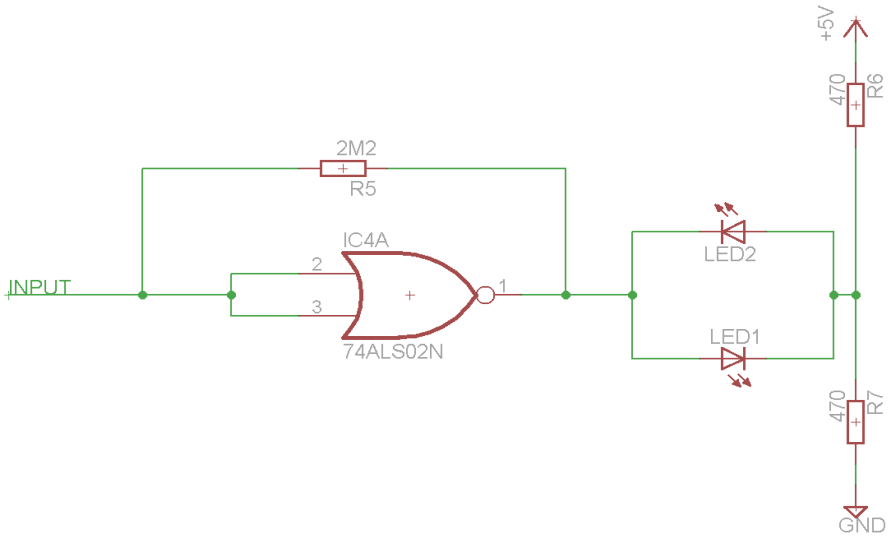


Figure 13: output LED circuit diagram

Create a new schematic of figure 14. To simulate the open circuit test condition an additional component has been added to the circuit i.e. a voltage controlled switch, component name SW. The operating parameters of this generic device are defined as a SPICE directive. This can be entered by left clicking on the pull down:

```
Edit -> SPICE directive
```

This will open the Edit text window, enter the following string and left click on OK.

```
.model SW SW(VON=1, VOFF=0, RON=1, ROFF=100MEG)
```

This defines the voltage that when applied to the inputs (+, –) will cause the switch to open (`VOFF`) or close (`VON`) its contacts. The resistance of these contacts when closed or open are defined by `RON` and `ROFF` respectively. Run this simulation to determine the LED status for a logical '0', '1' and open circuit input condition. You will need to define the voltage levels / pulses for V1 (power supply), V2 (logic signal from UUT) and V3 (simulating when the logic probe is attached / removed from UUT).

Note, you may need to run this simulation for a significant period of time.

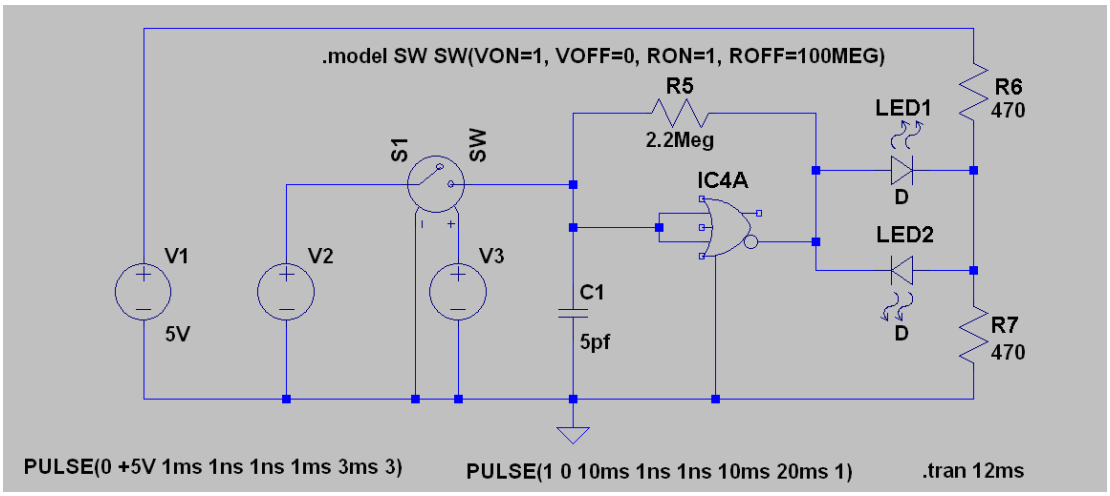


Figure 14: LED driver SPICE simulation model

Task 8

The edge detector and monstable multivibrator are used to drive the Pulsing LED, as shown in figure 15. This LED is used to indicate that the signal under test is changing. In general, if these test signals were used to drive a LED directly the flashing light would be changing too fast for the human eye to see. Therefore, the monstable multivibrator is used to 'slow' down these signals. Create a new schematic of this circuit and run the simulation. How does this circuit work? Why is the edge detector circuit required?

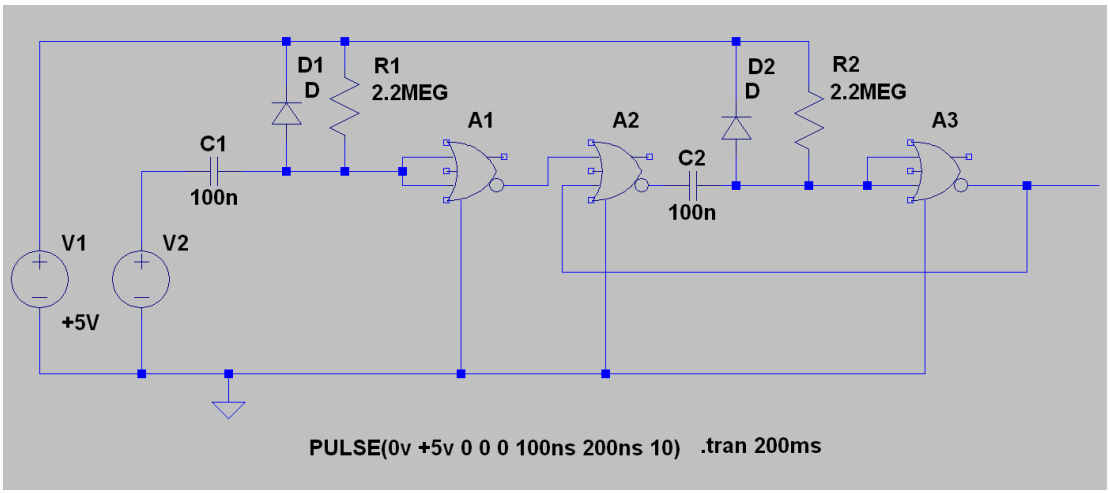


Figure 15: Pulse detector SPICE simulation mode

Implementation

The logic probe will be implemented on Veroboard (also called stripboard), a 0.1" grid of holes connected by parallel copper tracks, as shown in figure 16. The bottom layer has copper strips allowing components to be soldered onto the board, mounting and connecting the components to form the required circuit.

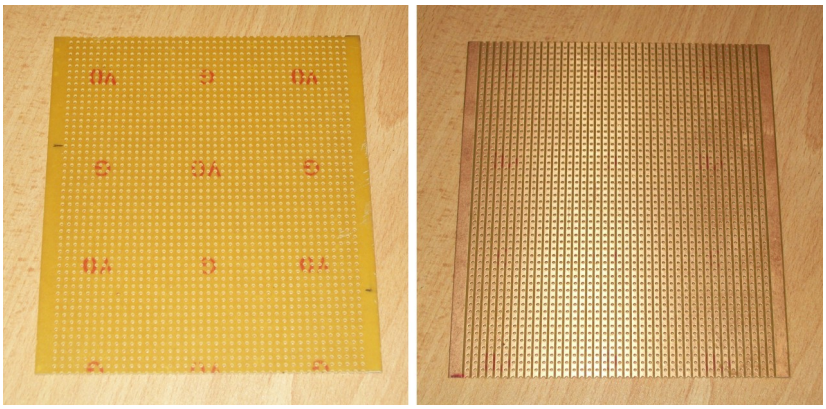


Figure 16: Veroboard, top layer (left), bottom layer (right)

There are three main methods of laying out a circuit on Veroboard: city-block, dead-bug, or hybrid as shown in figure 17. The city-block approach mirrors the layout of a modern city with wiring routed using 90 degree turns i.e. using horizontal copper strips and vertical wire links. A key advantage of this approach is that it is easy to debug and repair. The dead-bug approach, so called for the upside down ICs (the dead bugs) uses thin wires to connect components together. Now the shortest point-to-point route is taken when wiring a circuit. A key advantage of this approach is that it minimises size and interference i.e. parallel copper tracks can cause capacitive coupling between wires. The final method is a hybrid of the two approaches.

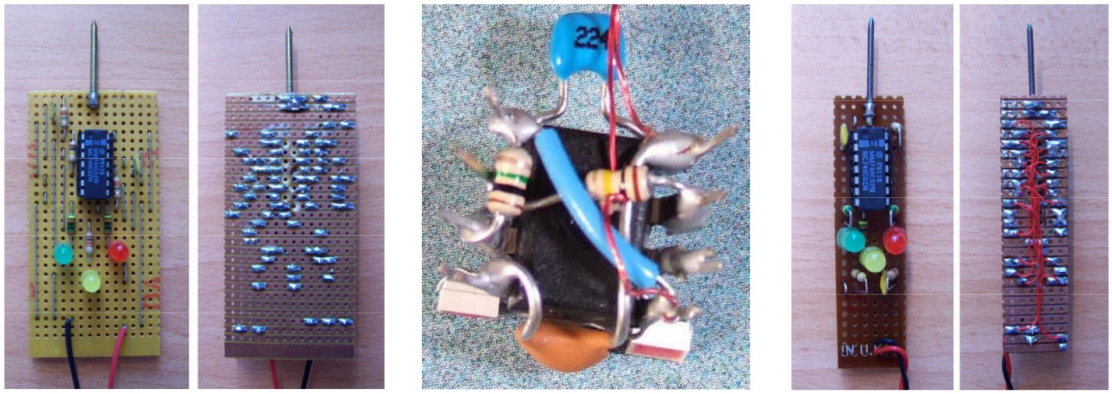


Figure 17: Veroboard circuit construction, city-block (left), Dead-bug (middle) and Hybrid (right)

The translation of the circuit diagram to layout can be done 'free hand', incrementally soldering components onto the board, allowing the layout to evolve. A better approach is to prototype the design using computer aided design (CAD) tools. These convert a schematic into a netlist, a textual description of the components used and their interconnects. This is then used in a layout tool to place and route these components on a virtual sheet of Veroboard.

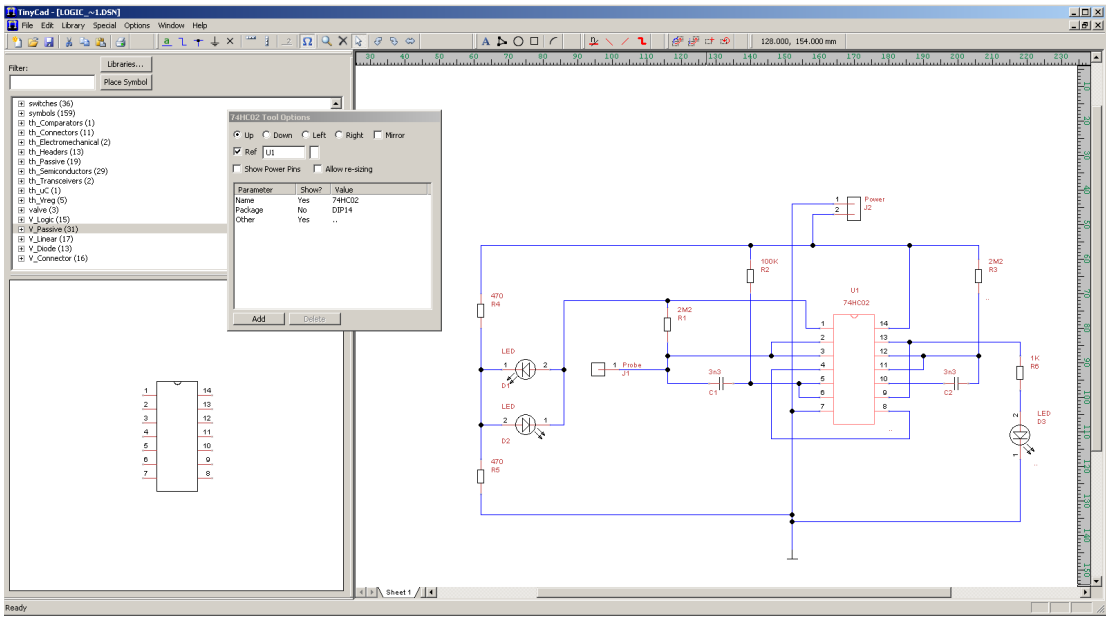


Figure 18: TinyCAD schematic

To enter this schematic we will be using TinyCAD. This software can be downloaded for **home** use from:

http://tinycad.sourceforge.net/

To start this schematic capture tool click on the start button and select TinyCAD.

```
Start -> All Programs -> Hardware development -> TinyCAD
```

This will start the main user interface. A 'basic' schematic of the logic probe has already been created and can be downloaded from the module web page.

This circuit diagram uses a generic 14pin IC component to represent the 7402 NOR gate. Additional components can be added to this schematic, however, you **MUST** use components from libraries starting with "V_". These can be added to the project by left clicking on the pull down:

```
Library -> Add
```

then browse to the directory:

```
G:\Apps\VeeCAD\Library\TinyCAD
```

The "V_" libraries, as shown in figure 18 can now be selected. For more information on using TinyCAD left click on the pull down:

```
Help -> Open TinyCAD user manual
```

Task 9

Update component values and add any additional components to match your design created in the SPICE simulations. You may also wish to consider rewiring the 7402 package to minimise routing complexity i.e. to minimise wiring distance between pins, or components. When complete save this file and left click on the pull down:

```
Special -> Create parts list
```

This will open the 'Save a parts list window', click on the Export button to save this text file to the project directory. Minimise the parts list to return to the schematic, then left click on the pull down:

```
Special -> Create PCB netlist
```

The 'PCB Export' window will open allowing you to select the format (Protel) and destination (home directory) of the netlist. Left click on the Export button.

Note, the 'Design Markers' window displays potential design errors, those referring to pin 8 can be safely ignored, left click on the Close button.

This netlist can now be loaded into the Veroboard layout tool: VeeCAD. To start this design tool click on the start button and select VeeCAD.

```
Start -> All Programs -> Hardware development -> VeeCAD
```

The Veroboard has been pre-cut 15 x 30 holes, if required this can be trimmed using side cutters to a smaller size. To update the VeeCAD board size left click on the pull down:

```
Board -> Size
```

To import the logic probes netlist left click on the pull down:

```
Netlist -> import
```

A component library must again be added to this project, left click on the `ADD` button and browse to:

```
G:\Apps\VeeCAD\Library\V_Standard.per
```

then click on `Open`. Ensure the netlist format is set to 'Protel', then click on the browse icon ... to select the netlist exported from TinyCAD i.e. `logic_probe.net`. Click on the Import button to finish. Each schematic component should now appear on the Veroboard, as shown in figure 19.

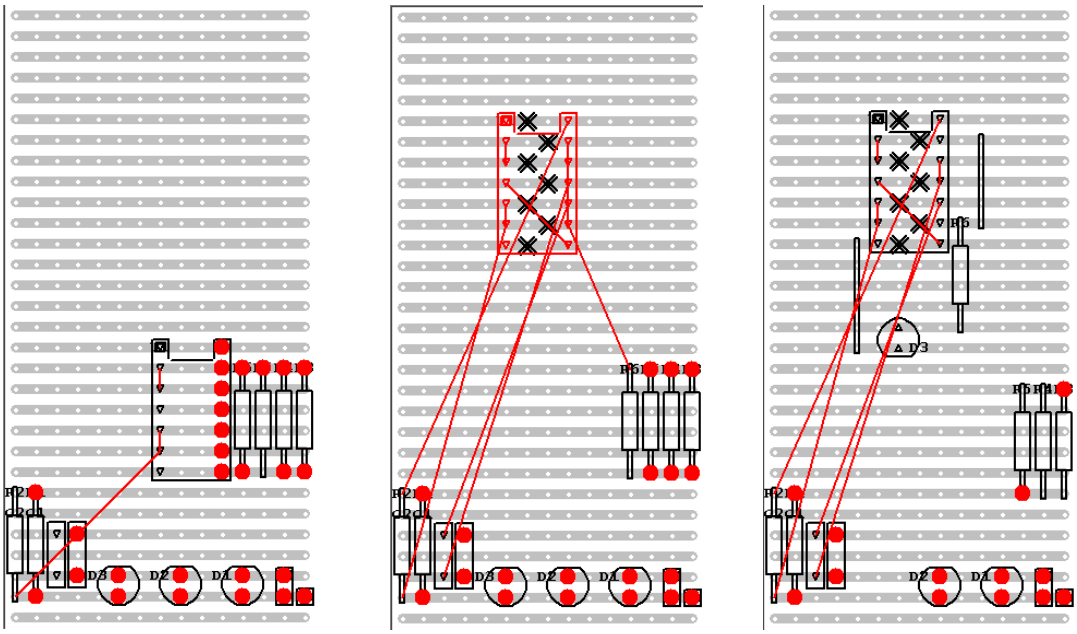


Figure 19: Veroboard layout, initial (left), adding track breaks (middle), adding links (right)

Task 10

The next task is to layout and route the circuit diagram on Veroboard. Left click, hold and drag to place a component. Short circuits i.e. components / wires incorrectly connected to the same track are illustrated by a read dot e.g. the right hand side of pins on the IC in figure 19. To isolate these wires onto different tracks, the copper track can be broken using a small hand drill. Within the simulator left click on the icon, next move the mouse to the correct position and left click to place a track **break**. This will remove the red dot, the simulator will now indicate what components should be connected to each pin using a red line. To illustrate the layout process resistor R6 and D3 have been placed and routed in figure 19.

Note, refer back to the TinyCAD schematic to determine how each component is connected.

To **move** a component left click on the  icon, then left click on the component, hold and drag it to its new position. A component can also be **rotated** by left clicking on the  icon.

When a component has been placed additional wires (links) may be required to complete a circuit. To place a link left click on the  icon, move the cursor to the start point, then left click, hold and drag the link to its required length. To finish drawing the link release the left mouse button.

When routing a design you may wish to make small modifications to the circuit diagram e.g. switch which logic gates are used to simplify routing. To achieve this, update the TinyCAD schematic, regenerate the netlist and import into VeeCAD to update your design. Place and route the logic probe schematic in VeeCAD.

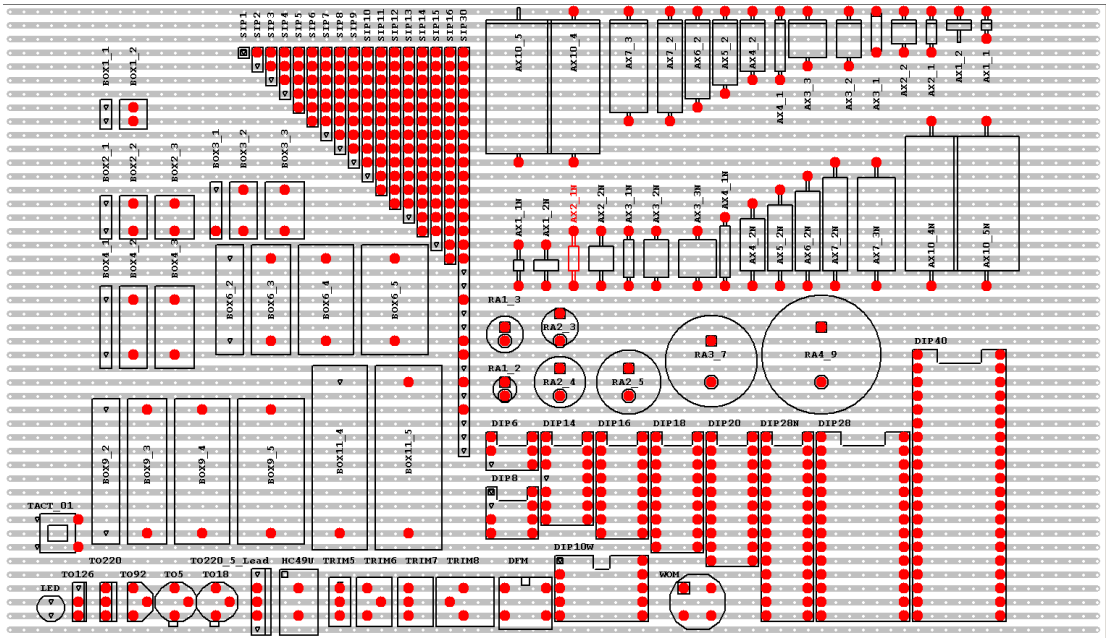


Figure 20: Component packages

Task 11

Construct and test the logic probe on Veroboard as demonstrated in the lab. When laying out this board you may wish to change a component's package i.e. use a smaller / larger resistor. The supported component packages are shown in figure 20. Each package is given a name e.g. the standard resistor is AX2_1N. Within TinyCAD highlight the component to be changed, modify the package name, then re-import the updated netlist into VeeCAD. You may wish to prototype your design on blue board before implementation.

Appendix A : Logic Probe Circuit Diagram