1. **Project Goals**

The project goal is a circuit debugging kit that includes a breadboard tester, an IC tester, and a circuit tester. The breadboard tester will determine if there are any defects in it. It will be able to figure out if every pin in the same column is connected, and if no pins in the adjacent columns are connected. Given an IC and its model number, the IC tester will be able to detect if the IC is working properly. If the IC is not working properly, then the IC tester will be able to diagnose the fault to let the user know which part of the IC is working properly. Alternatively, if the model number is unknown, then then IC tester will attempt to determine the IC model. A circuit tester will be able to check if the built circuit matches the desired function.

1. **Product Need**

Building logic circuits are already aggravating enough due to faulty wiring, faulty wires, wrong logic, power supply errors from fan-out, short circuiting, and being off by 1 column. They are even more aggravating after rebuilding the circuit multiple times to figure that the circuit does not work due to simple problems such as faulty integrated circuits or a defective breadboard. Our breadboard debugging kit can solve both these problems. The user will be able to find out if either the breadboard or the integrated circuit is defective. If neither is defective, the user will know to check for other problems associated with breadboard circuits.

The current solution for testing a breadboard is to send an input signal through one pin hole and check if the signal exists in the other 4 pin holes of the same column and check if the signal does not exists in the pin holes in the adjacent columns. This method is very tedious and time consuming.

There exists an IC tester that is very expensive and high capability, which is impractical for an average ECE student. Without buying the IC tester, the user would have to check if the integrated circuit is giving the correct output based on the given input. This forces the user to actually build the circuit before finding out whether or not the IC is defective. If faulty wires, wrong logic, or incorrect input was used to test the circuit, then the user will assume the IC was defective when it is actually not. Our product will allow the user to test if the IC is defective without having to build an actual circuit first. In addition, the device will be a fraction of the cost of the expensive IC tester.

Our device will also include a circuit tester. Technically, the IC tester and circuit tester will be the same system. The main difference is that the IC tester will be using pre-programmed data while the circuit tester will require the desired function. The device will measure the outputs of the circuit and compare them to the computed outputs of the function entered by the user. The results will then be displayed to the user.

1. **Technical Specifications**

*Breadboard Tester*

The breadboard tester is a ≥30 pin (5 rows and ≥6 columns) circuit that connects to the microprocessor and fits into a breadboard. The microcontroller sends a signal (+5 V) to a pivot pin. Every nearby pin should attempt to read the signal to check for a connection between itself and the pivot pin. If the signal is not detected in the same column or if the signal is detected in an adjacent column, then pause and display the location of the error on the LCD screen.

*IC Tester*

An IC can be placed onto a provided socket and the IC model number can be entered into the microcontroller (e.g. 74LS08 (Quad AND) via touchscreen). Then, the user will then initiate the test. If the IC behaves like a 74LS08 for every set of inputs and outputs, then the IC should be working properly. Otherwise, our device will notify the user whether the IC is faulty. If only certain gates are not working properly, the user will be notified exactly which gates are not working properly.

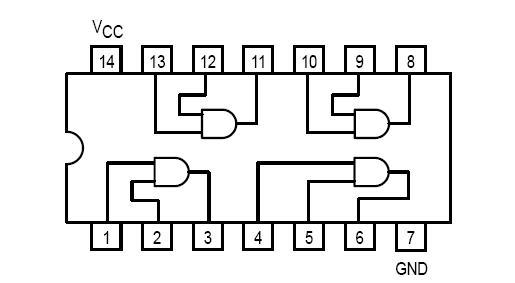


Figure : 74LS08 (Quad AND)

Alternatively, if the IC model number is not available, then the user can choose the “Unknown IC” option. The microcontroller can go through all ICs in the data library and attempt to find an IC model that behaves exactly like the inserted IC. If the device finds a match, then it will display the model number to the user. Otherwise, the device will assume either the IC is faulty or the IC model is not in the data library.

*Circuit Tester*

A circuit tester works very similar to the IC tester. Instead of having the inputs and outputs line up like the way they are on an IC, the user has to manually place the header pins in the input and output lines of the circuit. On the LCD, the user will choose the inputs and outputs desired that correspond to each pin. The device will then apply the inputs chosen by the user and test whether the circuit outputs match the desired outputs chosen by the user. The data will be displayed in the form of a chart that will highlight matching outputs in green and mismatched outputs in red.

*Touchscreen* *LCD Display*

The LCD must have no or minimal dead pixels and must be visible in normal, indoor lighting. The LCD display should display the correct menus and available user options upon startup. The touch input function should be working correctly on every part of the display. All user inputs, such as IC numbers, must be correctly displayed.

*Header Pin Cluster*

The pin cluster should be examined for bent or loose pins. The pins must be uniformly (and correctly) spaced.

*ZIF Socket*

The ZIF socket must have the same pin spacing as a common breadboard for IC insertion. All pin holes must be tested and working. ICs should fit snugly and removed easily when testing is complete.

*Wires*

There will be a great deal of wires that connects the microcontroller to the header pins. Those wires should be flexible, durable, and well protected.

1. **Engineering Design Alternatives**

*Design One (Single Pins)*

This single pin design is similar to just inserting a wire into a breadboard. Instead of a 5 by 6 pin cluster, the device will have 30 single pins. The user inserts n pins into n sockets of breadboard and our device will tell whether or not those n sockets are connected. This design allows for the most flexibility and ease of use. Also, using separate pins will reduce or eliminate the possibility of bent pins from repeated insertions and removals of the header pin cluster. Each pin will be numbered so the user can choose which pin to be the pivot pin via the LCD screen. The display will output which pins are connected and disconnected from the pivot pin.

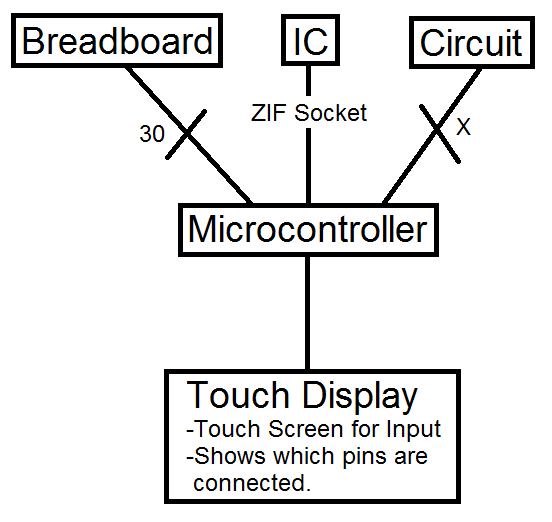
**

Figure : 30 Single Pins with Touchscreen LCD

*Design Two (Computer Display)*

The device will just consists of the pins for testing ICs and the breadboard and a microprocessor to select which pins to test. Our device will be powered via USB from a computer. All information our device gathers from testing, either an IC or a portion of a breadboard, will be sent to the computer and the computer will be used as a display as well as the processing component. All user input will be entered on the user’s computer. The design will ultimately save us and the consumer money because we would not need to purchase an external display. The downside is 1) the compatibility due to different operating systems and 2) having to figure out how to transport data to and from our device and the user’s computer.

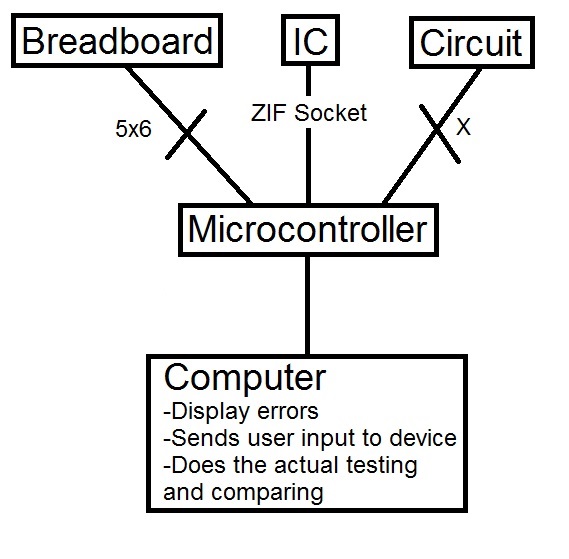


Figure : 5x6 Pin Cluster with Computer Display

*Design Three (Built-in breadboard with tester)*

The device will be almost the same as our original design except it will have a built-in breadboard. This way, the user will be able to both test and build the circuit on one device. The breadboard will be built over the tester portion of the device and will be connected to the microprocessor below. There will also be an LCD display like our original design but it will be placed at the top of the breadboard portion of the device and also connected to the microprocessor below. This design allows the user to test while building the circuit on the same device. If there is something wrong with the breadboard, the tester will detect it and show it on the LCD display. The breadboard portion will be detachable and can be replaced if there are problems with it. Instead of using a ZIF socket, there will be a certain part of the breadboard that will be used specifically to test ICs.

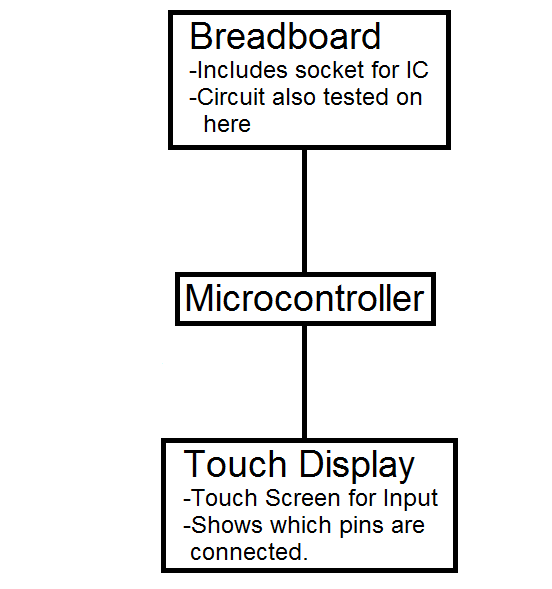


Figure : Breadboard with Built -in Tester and Touchscreen LCD

1. **Design Alternative Evaluation Criteria**

Our chosen design alternative evaluation criteria are the following:

Cost to Consumer

Development Cost

Technical Knowledge on Design

Time Needed to Complete Design and Development

Potential Hazards in Developing and Using the Product

Compliance with Regulations or Standards

Easy of Satisfying Technical Specifications

Ability to Manufacture

Easy of Product Use

Lifespan

Portability

1. **Selection of Design Alternative**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Design 1** | | **Design 2** | | **Design 3** | | **Original Design** | |
| **Criteria** | **Weight** | **Rank** | **Score** | **Rank** | **Score** | **Rank** | **Score** | **Rank** | **Score** |
| Cost To Consumer | 9 | 2 | 18 | 4 | 36 | 1 | 9 | 3 | 27 |
| Development Cost | 8 | 4 | 32 | 2 | 16 | 1 | 8 | 3 | 24 |
| Technical knowledge on design | 10 | 2 | 20 | 1 | 10 | 4 | 40 | 3 | 30 |
| Time needed to complete design and development | 7 | 4 | 28 | 2 | 14 | 1 | 7 | 3 | 21 |
| Potential hazards in developing and using product | 7 | 3 | 21 | 4 | 28 | 1 | 7 | 2 | 14 |
| Compliance with regulations or standards | 8 | 2 | 16 | 4 | 32 | 1 | 8 | 3 | 24 |
| Ease of satisfying the technical specifications | 6 | 4 | 24 | 2 | 12 | 1 | 6 | 3 | 18 |
| Ability to manufacture | 5 | 3 | 15 | 4 | 20 | 1 | 5 | 2 | 10 |
| Ease of product use | 9 | 2 | 18 | 4 | 36 | 1 | 9 | 3 | 27 |
| Lifespan | 8 | 1 | 8 | 4 | 32 | 3 | 24 | 2 | 16 |
| Portability | 5 | 1 | 5 | 4 | 20 | 3 | 15 | 2 | 10 |
| **Total Score** |  |  | **205** |  | **256** |  | **138** |  | **221** |

Each criterion was given a weight between 1 and 10 (1 being not important and 10 being very important) and each design was given a rank between 1 and 4 (1 being the worst and 4 being the best). The score for each criterion was calculated by multiplying a design’s score by the weight of the criterion. The design’s total score is computed by adding every row in the score column. The design with the highest score is the best design whereas the design with the lowest score is the worst design.

*Technical Knowledge on Design*

This was chosen as the most important criterion because if we lacked the knowledge, we would not be able to implement the many functions of our product. This is the main reason we do not want to focus on creating design number two (computer display) because we are unsure of how to import and export data to and from our device and a computer.

*Ease of Use*

We felt the simpler it is to use our product, the more the consumer will want to use it. We wanted the minimal amount of ambiguity with pin insertion and user input. We will attempt to design the testing pins in a way where the majority of electrical and computer engineering students would have no problem using our product without a manual.

*Cost to Consumer*

Our product was designed to with students in mind. We want to keep the price low enough that at least universities would consider purchasing a few to have in the many labs on campus. This would allow easy access for students. At the same time, are trying our best to keep cost low so a student can afford to purchase one for himself/herself in he/she chooses.

*Best Design*

Based on our evaluation criteria, the best design alternative based on our chosen criteria is the design that will use a computer as a display and the processing unit (design two). Unfortunately, we are not confident we can implement this idea. Nonetheless, we will attempt this idea depending on the time. We will first attempt our original design. The breadboard tester and IC/circuit tester portion is required in both designs, so we will focus on completing the design of that portion first. We will start by programming a microcontroller and use a LCD touch screen for display and user input. We will attempt to use a computer as the display and processing unit only if we can get our original design working first.

1. **Design and Production Cost Analysis**

*Resource Requirements*

|  |  |  |  |
| --- | --- | --- | --- |
| **Quantity** | **Part** | **Part Description** | **Estimated Cost** |
| 10 | Header Pins | 5 Pin Connector | $3.67 for 10 |
| 10 | Header Pin Housing | 1x6 Clippable Housing | $0.67 for 10 |
| 5 | SIPO’s | Serial In, Parallel Out | $1.95 for 5 |
| 5 | PISO’s | Parallel In, Serial Out | $1.75 for 5 |
| 1 | 28 Pin Socket | ZIF Socket, 28 Pin, 0.3” | $2.95 |
| 1 | LCD Screen | Serial TFT LCD – 3.2” with Touchscreen | $85.00 |
| 1 | Microcontroller | Header Board for LPC2294 | $49.99 |
|  |  | **Total** | $146.97 |

*Cost Estimate*

The primary cost of our resources will be the LCD screen and the microcontroller. The price for this project is estimated to be around $147 before shipping and handling. Other expenses will occur if we choose to enclose our device in housing for a better user interface.

1. **Task Allocation and Schedule**

*Schedule*

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Month/Week** | **Phase of Project** | **Task to be Completed** |
| May-June | Summer Part 1 | Ordering Parts | Make final decision on parts and ordering/receiving them. |
| July-August | Summer Part 2 | Proof of Concept Output/Input Control | Modify any flawed designs.  Finalize the design of the logic of our device. |
| 08/26/2013 | Week 1 | Building the 5 by 6 Pin Cluster and Testing for Flexibility | Test the sturdiness of the soldered header pins, checking for incorrect connections. Optimizing size of pin cluster. Test how easy/difficult it is to send and receive signals to each point. |
| 09/02/2013 | Week 2 | Programming the Microcontroller for Pin Connectivity | Work on chaining the SIPOs together so that we can send more than just 8 bits out using only 1 clock and data signal. Work on figuring out how to actually store the data so that it will show up in the SIPOs as expected. |
| 09/09/2013 | Week 3 | Work on actually being able to send and read each point of the device that connects to the breadboard. |
| 09/16/2013 | Week 4 | Work on sending a current to only 1 point at a time and reading the signals of neighboring points. |
| 09/23/2013 | Week 5 | Programming the Microcontroller for IC and Digital Logic Circuit Verification | Work on being able to store IC and Logic functions into memory and sending out the data control the devices. |
| 09/30/2013 | Week 6 | Work on reading every output. Work on matching the output with the expected outputs. |
| 10/07/2013 | Week 7 | If the outputs does not match the expected outputs and if it is an regular logic IC, then work on attempting to identify which gates do and do not work. |
| 10/14/2013 | Week 8 | Programming the Microcontroller for Output on to LCD Display and User Input | Learn the commands of the LCD. Attempt to show the bitmaps for some inputs. |
| 10/21/2013 | Week 9 | Continue to attempt to identify which logic gates do and do not work. Then attempt to show that information on the LCD display. |
| 10/28/2013 | Week 10 | PCB Mounting and Programming the Microcontroller for IC Detection (if time permits) | Create the design in AutoCAD for drill files, silkscreen, copper traces, etc. Putting together the finalized circuit and attempt further testing. |
| 11/04/2013 | Week 11 | Create the design in AutoCAD for drill files, silkscreen, copper traces, etc. Putting together the finalized circuit and attempt further testing. |
| 11/11/2013 | Week 12 | Debugging/Testing | Testing and debugging the Breadboard tester until we obtain our desired results. |
| 11/18/2013 | Week 13 | Testing and debugging the IC/Circuit tester until we obtain our desired results. |
| 11/25/2013 | Week 14 | Testing and debugging the Input and LCD Output until we obtain our desired results. |
| 12/02/2013 | Week 15 | Final Testing and Results | Complete Final Testing |
| 12/08/2013 | Week 16 | Complete Report |

1. **Simulation/Modeling Results**

See Appendix B.

1. **Description of the User Interface**

See Appendix C for software flowchart.

User Interface has been covered in **Technical Specifications**

1. **Additional Issues**

*Circuit Input Logic*

Some people connect their circuit inputs to high (Vdd) and use switches connected to pull-down resistors (resistors connected to ground) for low logic. Other people connect their circuit inputs to ground and use switches connected to pull-up resistors (resistors connected to Vdd) for high logic. Using the circuit tester may require breaking the primary inputs and outputs apart so that we can insert our device.

1. **Conclusion**

During the course of the design work, we came up with ideas, collaborated, and presented our ideas. Our ideas were almost never the best and many of them were not plausible, but we would not have known that unless we discussed them with each other. This is because we were not aware current existing products and flaws with our own ideas. The brainstorming of ideas really forced us to work in a group to figure what we are capable of. Then we designed the project based on our combined capabilities.

Working in a group had its ups and downs. There were quite a few times where we had disagreements on the project ideas and the design. Eventually, we were able to overcome those arguments and decide on a project idea we could all agree on. Working in a group was not all arguing and disagreements. When thinking of design alternatives, we were able to think of new ideas based on another person’s bad idea.

Lastly, through this design process, we were given the opportunity to present our ideas to the rest of the class and a faculty advisor. This was a good chance to attempt to convey our ideas and receive constructive criticism.