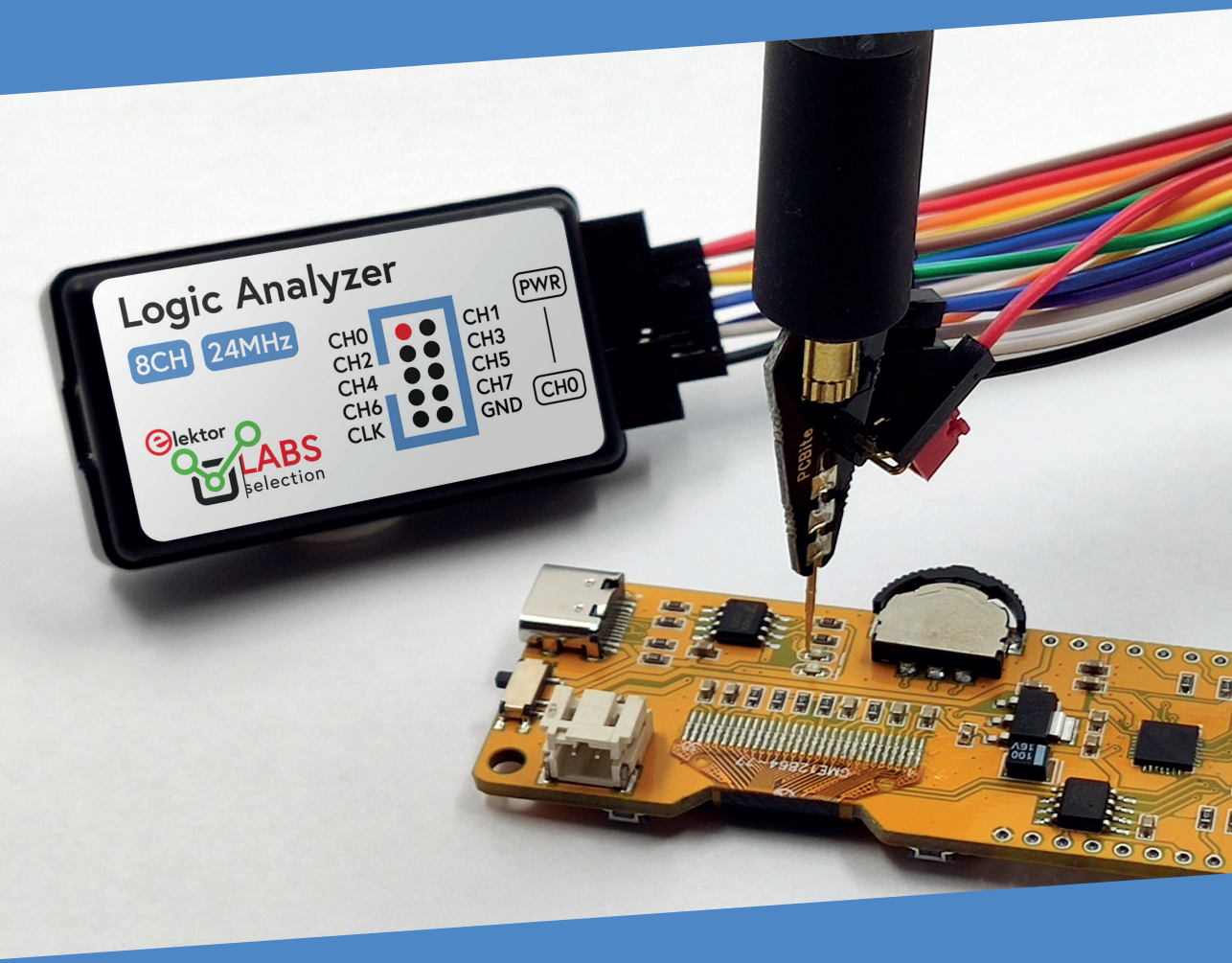
elektor books

Logic Analyzers in Practice

PC USB Logic Analyzers with Arduino, Raspberry Pi, and Co.



Jörg Rippel

elektor
design > share > earn

Logic Analyzers in Practice

PC USB Logic Analyzers with Arduino, Raspberry Pi and Co.

Jörg Rippel

elektor
design › share › earn

This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
PO Box 11, NL-6114-ZG Susteren, The Netherlands
Phone: +31 46 4389444

Declaration

The author, editor, and publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, and hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause. All the programs given in the book are Copyright of the Author and Elektor International Media. These programs may only be used for educational purposes. Written permission from the Author or Elektor must be obtained before any of these programs can be used for commercial purposes.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 978-3-89576-592-6 Print
ISBN 978-3-89576-593-3 eBook

www.elektor.com
Translator: Martin Cooke
Prepress Production: D-Vision, Julian van den Berg
Printed in the Netherlands

Contents

Preface

Digital signals and complex communication protocols can be found in almost every electronic circuit nowadays. It has become easy to connect sensors and displays to our microcontrollers with just a few lines of software and a Raspberry Pi or Arduino. We hope that everything works flawlessly, but sometimes it doesn't, and it becomes necessary to examine the digital signals and the protocol used in detail.

But how do you do that? Today's protocols have become so complex that trying to verify and decode them can be quite puzzling.

This is where a Logic Analyzer can come to your rescue. This book provides you with guidance for taking a leap into the digital world and demonstrates various methods to decode common protocols by working with a wide range of examples. It helps you navigate your first steps with a logic analyzer successfully and offers a structured approach with the most suitable tool for troubleshooting digital circuits; the logic analyzer.

This book showcases multiple models of flexible and widely used USB logic analyzers in various price ranges, highlighting their strengths and weaknesses. You will learn about the important criteria necessary for various application fields so that you can choose the device that suits your needs.

Whether you're working with Arduino, Raspberry Pi, or Raspberry Pico, the illustrated example circuits enable a quick start in protocol analysis and can serve as a foundation for your own experiments.

You will become familiar with all the essential terms and relationships, conduct your own experiments and analyze protocols independently. With this book by your side you will build a comprehensive knowledge in the realm of digital signals and protocols.

I wish you a lot of fun along the way!

Chapter 1 • What You Need to Know About Logic Analyzers

1.1 What is a Logic Analyzer?

A Logic Analyzer is designed to capture and record multiple digital signals simultaneously. Think of it like a video camera that can record a longer or shorter film depending on its resolution. If the video has an 8K resolution, the storage space on the SD card fills up quickly. If you record in VGA resolution, you might capture fewer details, but you can record a much longer video. A Logic Analyzer works on a similar principle. With a high sample rate, you capture more of what's happening on the bus, but the recording duration is relatively short.

What sets apart an Oscilloscope from a Logic Analyzer? A Logic Analyzer is similar to an oscilloscope, but it's primarily used in digital hardware development and troubleshooting due to its specific functionality. It offers the advantage of having more channels and being designed specifically for this purpose. While an oscilloscope typically has 2 or 4 channels, a basic modern Logic Analyzer already comes with 8 or 16 channels, and more advanced models offer over 32 channels. In the professional realm, devices with 128 channels are not uncommon. Early on in the analysis of microprocessor circuits, the need for these numerous channels became evident. Protocols can run not only on many parallel lines but can also be quite complex. To manage this complexity, the Logic Analyzer must have additional capabilities in its toolkit. It measures the connected signals simultaneously on all channels and can capture specific protocol events using a trigger. This allows for the visualization of complex dependencies among parallel signals.

Many aspects are reminiscent of an oscilloscope, and the parallels are undeniable. However, the creators of these two devices had different requirements in mind when building them. Unlike the oscilloscope, which was originally designed in the age of analog electronics, the Logic Analyzer has been developed and optimized in every aspect to work with digital signals.

To fully understand the complexity of the protocols used in digital circuits, a Logic Analyzer has become indispensable for circuit protocol analysis. If a problem arises in a digital circuit, for example, if a display controlled by the I²C protocol is not functioning as expected, a developer can try various things in the source code and libraries to find the error, but what's actually going on in the hardware remains elusive. Is the issue related to signal timing or clock problems, signal line interference, or bus collisions? A Logic Analyzer is often the only tool that can precisely pinpoint the problem.

The Logic Analyzer is however not necessarily the ultimate tool for troubleshooting. Using a Logic Analyzer can sometimes be time-consuming and complicated. You might spend several days poring over datasheets, protocol descriptions, protocol analyzers, and setting up trigger events just to get close to identifying the problem.

Only when you've exhausted all the software-based options for troubleshooting, you will need to dig deep with a Logic Analyzer and invest the necessary time.

At first during troubleshooting, you might feel overwhelmed by the abundance of information available from the Logic Analyzer. Don't let this put you off! It gets better every time you use it. Over time you'll become skilled and adept at handling the controls and interpreting the display. You'll find efficient ways to systematically narrow down and locate the problem, even with new protocols and issues.

To better understand a Logic Analyzer, you need to grasp some basics of how it functions. A Logic Analyzer not only visualizes the timing of the digital signals it captures but can also interpret them if desired. This is done using special protocol analyzers that analyze and decode the specific protocol into a visually understandable format. Each protocol requires its own protocol analyzer.

Sometimes an oscilloscope can complement a logic analyzer in troubleshooting. It excels in accurately displaying analog signals due to its significantly higher sampling rate and signal bandwidth compared to a logic analyzer. An oscilloscope (scope) is the go-to choice when examining analog signals. While it can also view the same digital signals as a Logic Analyzer, it's typically not optimized for this purpose and is limited to measuring up to four signals by using a 4-channel scope. It's therefore of little use when dealing with a digital circuit featuring a 32-bit data bus and a 64-bit address bus.

What is a scope good for? The scope is designed to provide a highly precise representation of an electrical signal. It excels at showing small irregularities, spikes, or signal overshoots. For instance, it's particularly useful for investigating signal integrity. This is not a strength of most Logic Analyzers. A Logic Analyzer doesn't display these irregularities because its main function lies elsewhere. It represents an idealized form of signals by categorizing them as either high or low signals based on their logic levels. The Logic Analyzer has a different perspective on signals, as its signal capture and measurement are structured differently to a scope. Its primary concern is capturing all the 0s and 1s, not the precise signal waveform. As an extreme example, a sinusoidal waveform will be represented as a square wave by a Logic Analyzer.

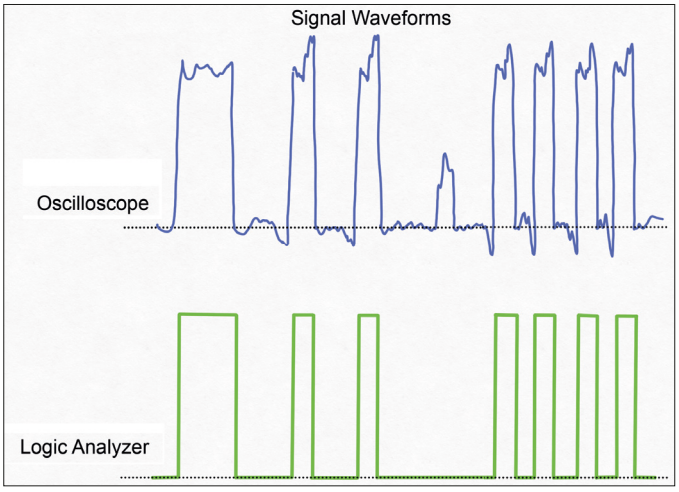


Figure 1.1: Comparison of a waveform displayed on a scope and a logic analyzer.

Keep this difference in mind: when troubleshooting digital signals, an Oscilloscope can sometimes come to your aid because some signal issues might go unnoticed by a Logic Analyzer. As you can see in Figure 1.1, a relatively weak signal in the middle of the sequence is not recognized as "High" by the Logic Analyzer, and its interpretation of the signal sequence remains "Low" at that point. But what about this potential signal? Was it too weak? Could it have been a noise signal that, if slightly stronger, would have been interpreted as "High" and disrupted the logic? Is this signal attenuated because it's being loaded by the Logic Analyzer probe impedance? (The connection of a Logic Analyzer can have effects on the circuit, which we will look at later). This puzzle needs to be solved, and that's where a scope can come in handy since it's designed for analyzing individual signals. In certain types of errors, an Oscilloscope can also assist with digital circuits. However, errors and issues related to messaging protocols are the strengths of a Logic Analyzer.

Structure and History of an MSO

In the past, oscilloscopes or just plain 'scopes' used to be analog in their construction. These devices directly displayed the signal applied to the input on the screen. The signal received at the input jack was processed and directed onto the inner surface of the cathode-ray tube as an electron beam. This process excited the phosphor layer on the inside of the screen, making a bright line visible on the display. Today, these devices have all but disappeared from the market. In some workshops, you might be lucky enough to spot the once very popular Hameg 203 scope.

Today's oscilloscopes on the market are known as DSOs, short for Digital Storage Oscilloscopes. They operate digitally, transforming the analog input signal into a digital signal stored in a data memory. From there, the signal information is further processed using an FPGA and a lot of software before it's displayed on a screen.

The construction and operation of modern oscilloscopes are entirely different from earlier analog oscilloscopes. Consequently, the most significant technical difference between an oscilloscope and a Logic Analyzer has disappeared.

The result of this evolution is the MSO, the Mixed Signal Oscilloscope. The digital architecture of today's oscilloscopes allows for the combination of both functions. Since the signal at the input jack is already converted into a digital signal, the transition to a Logic Analyzer is no longer a long journey. The Logic Analyzer and the Oscilloscope merge together. For this reason, some manufacturers offer their MSO as two devices in one, an oscilloscope with an integrated Logic Analyzer.

Figure 1.2: Beneath the display on an MSO you will usually f ind the logic analyzer input pod connector.

Back to the Logic Analyzer.

So, the Logic Analyzer is a measuring instrument that can simultaneously capture, record, display, measure, and decode digital signals across all its channels. When analyzing the recorded signal waveforms, it allows you to examine the temporal correlation and dependencies between the signals. Protocols can be deciphered and visualized using protocol analyzers, making the control and data signals visible to the human eye.

Figure 1.3: The DSLogic Plus logic analyzer — up to date and powerful.

Designed for both hobbyists and professionals, modern Logic Analyzers that connect to a computer via USB offer a wide range of possibilities. When used in conjunction with the accompanying software, they enable the analysis of digital signals, a capability that used to be found only in professional devices at a multiple of the price just a few years ago. The

constant development of plug-ins for this software provides a continuous expansion of features required for decoding ever-evolving protocols. This is a significant advantage compared to earlier devices that had static capabilities. Decoding new digital protocols remains possible long after the purchase of the Logic Analyzer, simply by updating the software.

The Purpose of a Protocol Analyzer

Regarding protocol analyzers, there's an important point to consider in understanding their purpose: A protocol analyzer makes the content of the protocol visible and comprehensible. It doesn't necessarily mean that it's easy to understand what this protocol does and transmits. For instance, often, the contents of registers from a sensor are transmitted, where a numerical value is encoded in hexadecimal. This introduces another layer where you need to decode the received content, for example, by converting that numerical value to obtain the temperature in Celsius or Fahrenheit. This final step happens in the device's software. You won't see this part in the communication between the sensor and the microcontroller. The sensor's manufacturer provides the necessary steps in the accompanying datasheet, which must then be implemented in the firmware to reach the value eventually displayed on the screen. This is often where many users face difficulty, leading them to give up after their first attempt at using a Logic Analyzer and then feel deflated. A protocol analyzer is useful here to make information readable and comprehensible. It still doesn't do all the work for you; it just makes it all a bit easier.

As mentioned earlier, Logic Analyzers are also available as part of an oscilloscope. As an additional module, alongside the regular analog inputs of the oscilloscope, the user can access another 16 or 32 digital channels. In such a Mixed Signal Oscilloscope (MSO), you have the opportunity to evaluate both analog and digital signals within a circuit at the same time and in relation to each other. When conducting the same analysis with two separate devices, a USB Logic Analyzer and an oscilloscope, this isn't as straightforward. Analyzing analog and digital signals in relation to each other with two separate devices often requires synchronization of the triggers between both devices. Although simultaneous capture and analysis are technically possible, the results still need to be merged across the device boundary, either visually or by transferring the collected samples to another system for further error analysis. Sounds difficult, error-prone, and complicated, doesn't it?

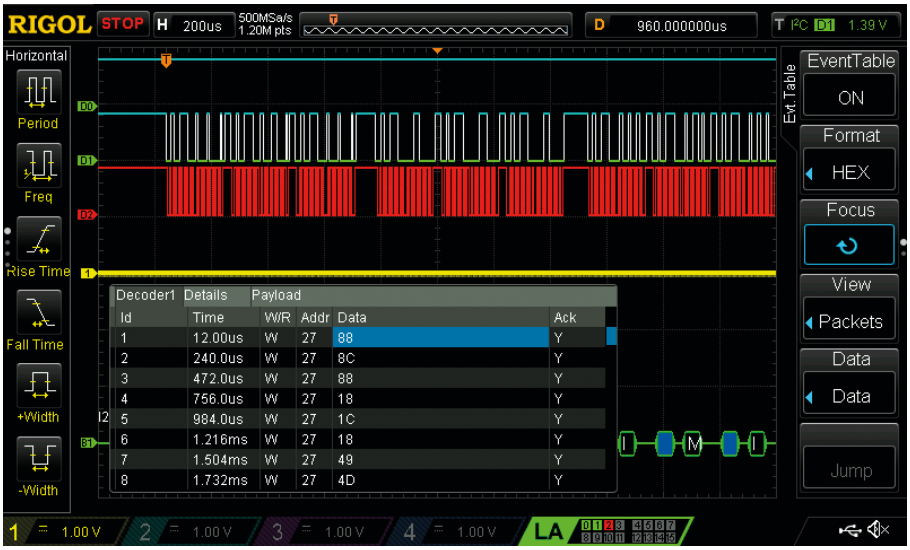


Figure 1.4: Analysis of the I²C-Protocol using a Rigol MSO.

To be fair, it's worth mentioning that the need for such a complex troubleshooting is quite rare. In most cases, a trained eye is sufficient, and more complex situations can usually be resolved using a degree of creativity.

A more significant drawback is the behavior of some MSOs. When using the integrated Logic Analyzer, they may deactivate one or two of the analog inputs or only decode the digital signals that are visible on the display, rather than decoding the complete signal history stored in memory. In this regard, a USB Logic Analyzer is much more flexible and powerful than the Logic Analyzers included in entry-level MSOs.

However, don't dismiss the idea of using an MSO for your purposes just yet. These devices have their strengths, which I'll discuss later. There's a right tool for every troubleshooting task, and in some cases, an MSO might be the best choice.

Now, you already have some knowledge about the hardware of a Logic Analyzer. Let's move on to the software.

1.1.1 What software can you use with a logic analyzer?

Every USB Logic Analyzer comes with software that is essential to complement the hardware of the Logic Analyzer. This software should be capable of controlling individual channels, enabling signal recording, and facilitating the analysis of the recorded signals. The analysis part is often achieved with the help of protocol analyzers, which are usually provided as plug-ins within the software.

One well-known piece of software for this purpose is the open-source software Sigrok. It supports a wide range of Logic Analyzers. In many cases, the software that comes with the Logic Analyzer from the manufacturer can be replaced with Sigrok. This is useful if you want

to continue using an old Logic Analyzer that is no longer supported or if your Logic Analyzer is based on a design from the open-hardware community. Sigrok is a command-line program running in the background, while the display and user interface are provided by the Pulseview program.

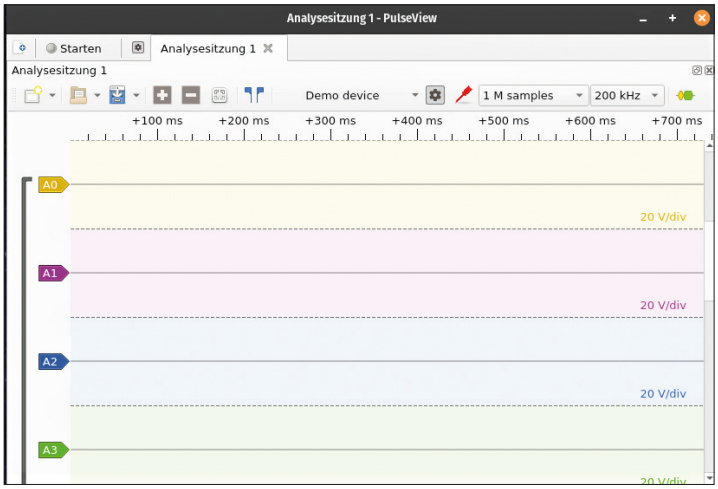


Figure 1.5: The Sigrok Pulseview Analyzer window — clear and straightforward.

The approach in this book is to provide the simplest possible introduction to working with a Logic Analyzer. One of the devices used in this book is a straightforward and budget-friendly model, stocked in the online Elektor Store at an affordable price. The cost-effectiveness of this basic, no-name USB Logic Analyzer is unmatched. With its 8 channels and up to 25 MHz bandwidth, it's well-suited for most tasks.

Figure 1.6: An affordable 8-channel Logic Analyzer for getting started with protocol analysis.

To use this Logic Analyzer, you'll need Sigrok and Pulseview. You can download the latest versions of this open-source software from the internet at http://sigrok.org. This makes it easy to dive into the world of decoding digital signals.

1.1.2 Will a basic logic analyzer do the job?

Modern microcontrollers often feature an I²C or SPI interface. The simplicity of the SPI and I²C protocols, their wide support for microcontroller peripheral communications, cost-effectiveness, and ease of implementation on circuit board designs has made these protocols ubiquitous. The serial interface RS-232 is still commonly used as well. This means that a simple low-cost 8-channel Logic Analyzer with a bandwidth up to 20 MHz will be sufficient for the majority of home lab setups.

The need for a Logic Analyzer with significantly more channels, for analyzing wide data or address buses like those used in a memory modules, is quite rare for a hobbyist. The clock frequencies on such buses are often so high that much more expensive and professional equipment would be required to carry out an analysis. In these early stages, save the money you would invest in a Logic Analyzer to enter this professional league. As you progress through the book, you will learn about the key points that matter. By the end of the book, you will know precisely which Logic Analyzer could be your next step after this simple 8-channel 24 MHz model. I promise! The principles are essentially the same: If you can handle one Logic Analyzer well, you'll quickly adapt to operating a different model.

How many channels will I need?

The peripheral interface protocols used by modern microcontrollers require fewer signals. When connecting a Logic Analyzer, in addition to a GND (ground) connection, you only need channels to connect to each of the following signals:

SPI = 4 signals (SDO/MOSI, SDI/MISO, SCK, SS)
I²C = 2 signals (SCL, SDA)
1-Wire = 1 signal (DQ)
RS-232 = 2 signals (TX, RX) (+ CTS, RTS i.e., 4 signals when using data flow control)
HD44780 = 7 or 11 signals (4 or 8-bit data transfers, RW, RS, E)

1.2 Why do I need a logic analyzer?

You need a Logic Analyzer because digital circuits are no longer built with logic gates where the signal's path and states can be easily traced. In the past, you could verify the correct operation of a digital circuit using a logic tester, known as a "*Logic Probe*". At various test points within a circuit or at the pins of the ICs (Integrated Circuits), you could check for a high or low signal and follow the signal path through the sequences of ICs until you find the error. You would then repair the faulty area, replace the affected IC, and with any luck the circuit would work again. The logic was "hardwired" within the ICs. Today, logic is defined by software.

Some Logic Probes could also count in binary, these high-end versions were called "*logic scopes*".

Figure 1.7: A logic scope from the 1980s which can also count in binary.

Apart from repairing old circuits from the 1970s and 1980s, you won't come across such devices these days. Modern digital technology operates using communication protocols with far more complex sequences than the old logic circuits. Due to the wide variety of modern microcontrollers and microprocessors, a multitude of protocols are in use within circuits. Overall, the density of circuits has significantly increased due to the much higher level of integration.

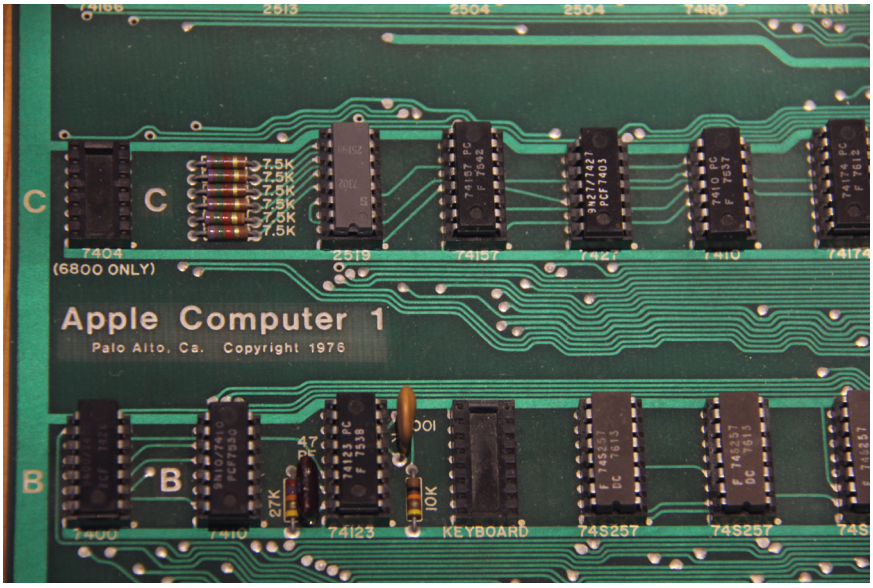


Figure 1.8: Logic chips on the first Apple motherboard were so widely spaced… 'you could drive a bus through there.'

While hobbyists these days primarily work with digital technology — almost everyone uses an Arduino, Raspberry Pi, or Pico in their projects — very few use a Logic Analyzer for troubleshooting.