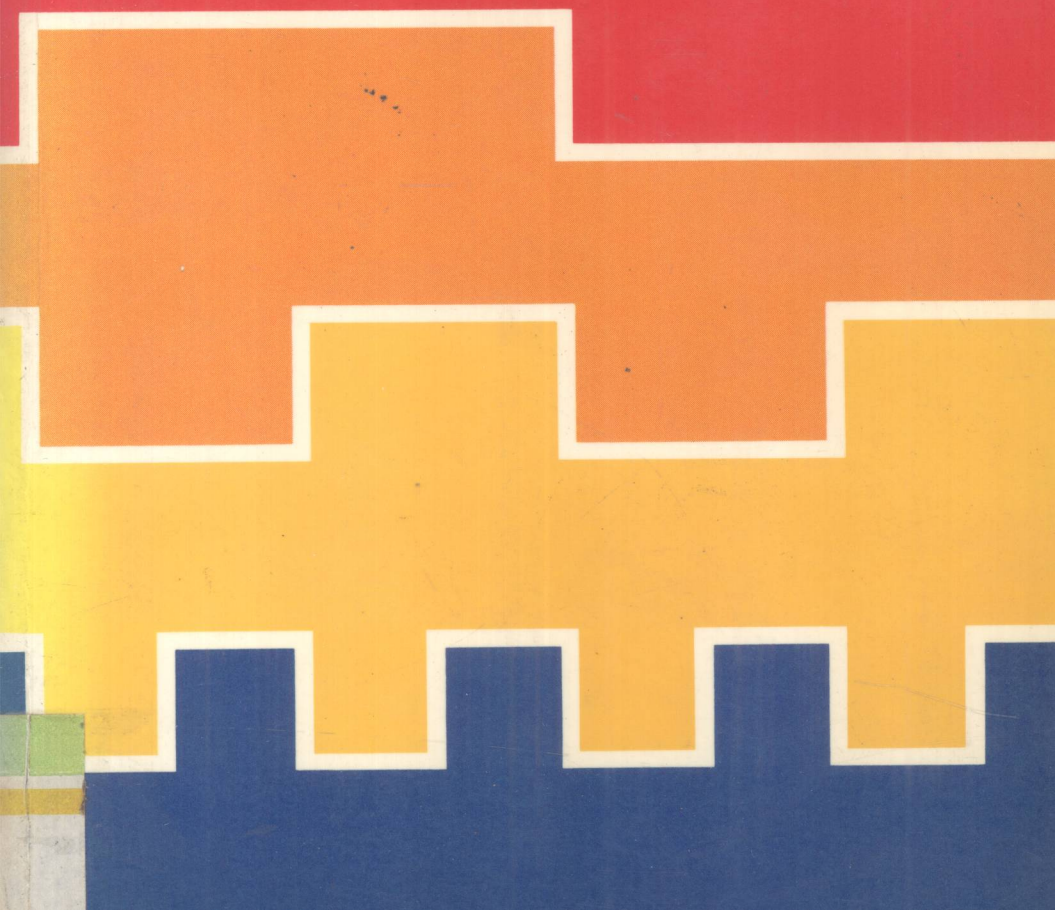


# **MICROPROCESSOR SYSTEM SERVICING**

**JOHN D. FERGUSON LOUIE MACARI  
PETER WILLIAMS**



TN 47  
F352

8960040

# MICROPROCESSOR SYSTEM SERVICING



E8960040

**JOHN D. FERGUSON**

*Computer Science Department  
Strathclyde University  
(formerly MEDC)*

**LOUIE MACARI**

*Senior Electronics Engineer  
James Howden & Co., Glasgow  
(formerly MEDC)*

**PETER WILLIAMS**

*Director  
Microelectronics Educational Development Centre (MEDC)  
(Paisley College)*



**Prentice / Hall**



**International**

Englewood Cliffs, NJ London Mexico New Delhi  
Rio de Janeiro Singapore Sydney Tokyo Toronto

---

*Library of Congress Cataloging-in-Publication Data*

Ferguson, J. D. (John D.)

Microprocessor system servicing.

Includes index.

1. Microcomputers—Maintenance and repair.
2. Microcomputers—Maintenance and repair.

I. Macari, L. II. Williams, Peter, 1937–

III. Title.

TK7887.F47 1986 621.391'6'0288 86-21246

ISBN 0-13-581132-5 (pbk.)

---

*British Library Cataloguing in Publication Data*

Ferguson, John D.

Microprocessor system servicing.

1. Microprocessors — Maintenance and repair

I. Title II. Macari, L. III. Williams, P. (Peter), 1937–

621.391'6 TK7895.M5

ISBN 0-13-581132-5

---

© 1987 John D. Ferguson, Louie Macari & Peter Williams

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Prentice-Hall International (UK) Ltd. For permission within the United States of America contact Prentice Hall Inc., Englewood Cliffs, NJ 07632.

Prentice-Hall Inc., Englewood Cliffs, *New Jersey*

Prentice-Hall International (UK) Ltd, *London*

Prentice-Hall of Australia Pty Ltd, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana S.A., *Mexico*

Prentice-Hall of India Private Ltd, *New Delhi*

Prentice-Hall of Japan Inc., *Tokyo*

Prentice-Hall of Southeast Asia Pte Ltd, *Singapore*

Editora Prentice-Hall do Brasil Ltda, *Rio de Janeiro*

Printed and bound in Great Britain for

Prentice-Hall International (UK) Ltd,

66 Wood Lane End, Hemel Hempstead, Hertfordshire, HP2 4RG

by A. Wheaton & Co. Ltd, Exeter.

1 2 3 4 5 90 89 88 87 86

ISBN 0-13-581132-5

## **MICROPROCESSOR SYSTEM SERVICING**

**To Lyn, to Anita and to Susanne**

---

# Preface

Servicing a microprocessor system requires a different base of knowledge from that for conventional electronics servicing. It does not demand a different attitude. The same questioning attitude, the same logical process of elimination holds good. Designers and technicians alike can have confidence that the good habits they have acquired, whether in analog or digital circuits, or in an earlier electromechanical era, will transfer to the microprocessor and its systems.

It would be foolish to deny the problems implicit in this new technology, but with the right attitude they can be kept in perspective. In this book we start from the assumption that the reader has a general knowledge of electronics, including digital techniques, with a logical approach to problems. We do not assume a detailed knowledge of microprocessors but a good appreciation of their principles will be helpful.

It is the purpose of Part I to bring these themes together so that readers with differing backgrounds can appreciate the new techniques presented in the body of the book. The aim is not to teach digital electronics nor to provide yet another 'Micros for Beginners'. Rather, it is to review these topics from a servicing standpoint in order to place the known facts in a new framework. For the practicing electronics engineer much in this first section will be familiar, though the viewpoint may be less so. Readers with a programming background may find these hardware ideas strange at first but without them the route to microprocessor servicing is blocked.

Part II looks at the techniques available for tackling the servicing problem. At each stage from development through production to user maintenance there will be common difficulties to which these techniques can be applied. The kind of information generated by any one method will vary in how helpful it will be to the individual. A designer needs to know why a system is not behaving as expected, while a serviceman working to tight time and cost budgets will give priority to the method that pinpoints a quick cure.

The ideal would be for each user to have a full range of professional equipment. Part III looks at simple designs that can be constructed to cover straightforward tasks and to offer cost-effective training tools. Because each user will have different priorities, the designs are presented

in outline form suitable for adaptation by experienced electronics engineers and technicians.

Chapter 1 reviews the characteristics of digital devices and the appropriate fault-finding techniques. Microprocessor systems are considered in Chapter 2, emphasizing those aspects that help the understanding of servicing ideas. In Chapter 3 the characteristics of the interface between microprocessor systems and peripherals is considered from the same standpoint. An introduction to fault finding in Chapter 4 looks at those general points that can be checked without special equipment. Self-testing, discussed in Chapter 5, assumes that the system functions sufficiently to allow it to generate test patterns to probe suspect sections. A logic analyzer (Chapter 6) monitors the sequential bit patterns on the system bus lines to reveal both straightforward component faults and the more subtle effects of timing errors. Once data transmission takes place there are additional sources of faults, and Chapter 7 introduces the characteristics and appropriate testing procedures for serial interfaces. In-circuit emulation is a technique available on development systems as a design tool but Chapter 8 looks at its role in fault finding. Chapter 9 covers signature analysis in which a system is first documented running a pre-defined routine: the bit-pattern at any point results in a unique code, any departure from which signifies a fault. Automatic test procedures for production lines can have application in fault finding but, as Chapter 10 shows, the actual equipment may bear little resemblance. Part III provides ideas for test equipment based on microcomputers and user-designed test boards. Chapter 11 illustrates this approach with a simple in-circuit emulator based on a standard microcomputer. Signature and logic analyzers can be implemented as low-cost single board devices for training purposes (Chapter 12). The serial interfaces between computers and their peripherals can be tested by using the serial analyzer and character generator of Chapter 13.

J.D.F., L.M., P.W.

---

# Contents

Preface xi

## PART I: MICROPROCESSOR SYSTEMS

### 1 Digital Devices and Systems 1

Current Probe	4
Logic Pulser	6
Logic Clips	9
Logic Families	9
TTL	10
Pulse Characteristics: Current Transients	17
Slow Transitions	18
Tristate TTL	18
Fault Tracing	19
ECL Emitter Coupled Logic	20
I <sup>2</sup> L Integrated Injection Logic	21
NMOS/PMOS/CMOS	22
High-Speed CMOS	25
Summary	27

### 2 Microprocessor Systems 28

System Components	28
The System Buses	29
(a) The Data Bus	29
(b) The Address Bus	30
(c) The Control Bus	32
Basic System Timing	35
Memories and Memory Maps	37
Address Decoding	40
Using Read-Only Memories for Address Decoding	43
Special Requirements of Dynamic Memory	44
The Microprocessor	47
Microprocessor Operation	53
(a) Arithmetic and Logic	55
(b) Data Transfers	56
(c) Branching Instructions	57
(d) Subroutines, Interrupts and the Stack	57
(e) Miscellaneous Instructions	59
Input/Output Circuits	59
Summary	62



**3 Interfacing 63**

Parallel Interfacing 64

The Centronics Interface 66

IEEE 488 Interface Bus 68

A controller – The HP85 Microcomputer 72

A Listener/Talker – The Solartron Locator 73

Serial Interfacing 75

Asynchronous Serial Transmission 78

Serial-Parallel and Parallel-Serial Data Conversion 79

Serial Communications Standards 86

(a) Single-Ended Voltage Interfaces – RS232, V24, RS423 86

(b) Current Loop Systems 89

(c) Differential Mode Systems, e.g. RS422 and RS485 91

Representation of Printable Characters 91

Summary 92

**PART II: SERVICING TECHNIQUES 93****4 Fault-Finding Techniques 95**

Basic System Tests 96

System Clock 96

Reset 96

Other Control Lines 97

Stimulus Testing 97

Free Running the Microprocessor 98

Power Supplies 101

Temperature Effects 102

Noise 102

Interface Faults 103

Peripherals 103

Types of Faults 104

Fault Logging 105

Checklists and Flowcharts 105

Summary 106

**5 Self-Test Principles 107**

RAM Testing 107

ROM Testing 111

Input/Output Devices 116

Summary 116

- 6 Logic Analysis 117**
  - Logic Analyzers – Basic Operation 119
    - Arm/Abort Control 123
    - Clock and Clock Qualifier Controls 123
    - Trigger and Trigger Qualifier Controls 124
  - Testing a Matrix Keyboard 124
    - Connecting the Logic Analyzer 126
  - Summary 127
- 7 Serial Interface Fault Finding 128**
  - General Techniques 128
  - Specialized Serial Interface Testing 129
    - Serial Data Analyzers 129
    - Special Purpose Data Generators 131
    - Back-to-Back Test Software 132
  - Summary 133
- 8 In-Circuit Emulation 135**
  - ICE as a Development Tool 137
  - Emulation on a Microprocessor Development System 137
    - Using the Emulator 140
    - Trace Analysis 142
  - ICE as a Fault Finding Tool 148
  - Summary 149
- 9 Signature Analysis 150**
  - The Signature Analyzer 151
  - A Closer Look at the Signature Window 153
  - Vcc, Vss Signatures 157
  - Free Running as a Source of Stimulus 157
  - An Example – Z80 Controller Board 158
    - Address Bus – Decode Logic Signatures 158
    - ROM Signatures 159
  - Stimulus Software 160
  - Exercisers and Emulators 160
    - Micropod 162
    - A Stimulus Program for a Single Board Computer 163
  - Summary 167
- 10 Computer-Aided Testing 168**
  - Automatic Test Equipment 169
  - Automated Testing 172
    - Fluke 9010A Troubleshooter 173

Operational Functions	173
Example – Implementing a Test Program for a Single Board Microcomputer	175
Memory Mapping	175
Built-In Test Functions	175
Programmed Testing	176
6522 (VIA) Test	177
6520 (PIA) Display Test	178
6532 (RIOT) Keyboard Test	178
Sequencing the Tests	181
Summary	182
Computer-Controlled Signature Analysis	182
Summary	187

### **PART III: SERVICING ON A SHOESTRING 189**

<b>11 Microcomputer Based Testers</b>	<b>191</b>
ICE Hardware	193
Test Software	193
(a) Address and Data Bus Toggle Test	195
(b) RAM ‘Checkerboard’ Test	197
ROM Signatures	197
Implementing a Test Program – An Example	200
Summary	202
<b>12 Single Board Analyzers</b>	<b>203</b>
A Signature Analyzer	203
Logic Analyzer	208
Summary	210
<b>13 Communications Tools</b>	<b>211</b>
An Interface Converter	211
RS232 and RS423 Signals	211
Current Loop Signals	213
Serial Analyzer Add-on for Logic Analyzer	215
A Serial Character Generator	216
Summary	218

**APPENDICES 223****Appendix 1 Microprocessor Comparison 225**

- Microprocessor Details 225
- The 6502 227
  - 6502 Signals 227
  - 6502 Instruction Mnemonics 230
- The 8085 232
  - 8085 Signals 232
  - 8080/8085 Instruction Mnemonics 236
- The Z80 238
  - Z80 Signals 238
  - Z80 Instruction Mnemonics (also NSC800) 240
- The 8088/8086 245
  - 8088/8086 Signals 246
  - 8088/8086 Programming Model 250

**Appendix 2 Technical Details of Three Single Board Microcomputers 252**

- AIM 65 Description 252
  - Overview 252
  - Functional Areas 254
  - R6502 254
  - Control Switches 255
  - Printer Interface 255
  - Display Interface 255
  - Keyboard Interface 257
- SDK-85 Single Board Development Kit 257
- MEDC Z80 Single Board Controller 260
  - The Processor and Associated Circuitry 262
  - Input/Output Circuits 264
  - General Purpose Interface 266

**Appendix 3 Power Supplies and Regulators 267**

- Power Supplies 267
  - Ripple Voltages 268
- Switching Regulators 277

**Appendix 4 Signature and Logic Analyzers – Suggested Circuits 283**

Index 287

# 1

---

## Digital Devices and Systems

Microprocessors need other circuits and devices to form a complete system. Logic circuits of different families differ considerably in their behavior, even when performing similar functions. An outline of these differences is given. Test devices are available that can trace a fault to a specific pin on a device, and their advantages and limitations are discussed.

\* \* \*

The best servicing aid to any system is a well-written manual. As in all other topics covered in this book, the reader should exhibit that true spirit of optimism without which servicing tasks can be so difficult – assume that the manual will be helpful until proven otherwise.

For the moment we ignore those unfortunately too frequent occasions when the manual is missing or hastily put together. It should give clear diagrams of the layout, and with luck a number of critical test points. A typical problem it can help avoid is in the measurement of clock frequency. A counter or oscilloscope may present sufficient loading to disturb the frequency of an oscillator if tapped onto a high-impedance port. Any manufacturer-designated test point should be well buffered, avoiding this risk.

The manual may contain a set of test procedures to trace the main categories of faults anticipated by the designers. Only the foolhardy would ignore such help, though in fairness there may be limited fault information available to the designers at the time of writing the manual.

It is important to realize that just because a pcb is part of a microprocessor system, any standard logic circuits behave in *exactly* the same way from a servicing standpoint as in older logic systems. Although the functions may be more complex, the electrical and mechanical faults are the same – track shorts, solder bridges, open-circuit gates. It may be more difficult to track down the area of the board where the fault occurs, but once narrowed down to one or two chips, all the well-known tests can

be used. In this section we shall review these tests: there are many books dealing with these techniques more fully.

When would these simpler methods be used, and when the advanced techniques discussed later? If automatic or semi-automatic tests are available, use them. They will have been implemented by professionals having considerably more experience and thinking time than you will have under pressure on an unfamiliar system. If by good fortune the fault is one that advertises itself, that is different. Symptoms visible on a display or output port may suggest a driver chip, and a quick test with a pulser and logic probe may identify the fault quickly. Equally, bad fortune may leave the service engineer without the appropriate analyzer or emulator tools at the critical moment. To fall back on these traditional approaches is then the only choice.

Let us identify some characteristics that can be picked up by suitable probes.

Static	Logic 0
	Logic 1
	'Bad' level (intermediate between 0 and 1)
	Open circuit
	Short circuit
Dynamic	Short duration pulses
	Transients
	Low repetition-rate pulses
	Reverse polarity pulses
	Current flow

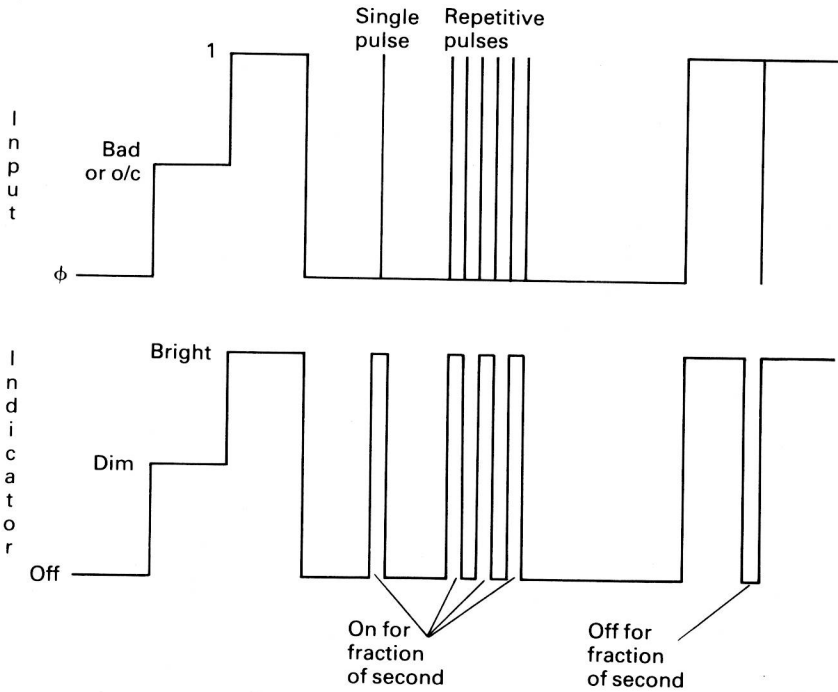
Logic levels should be either 0 or 1. As noted above, the precise levels at which these states are guaranteed varies between device families. Taking the usual TTL values, the voltages below 0.8 V are recognized as logic 0 and above 2.4 V as logic 1. Intermediate values imply a circuit fault, and a logic probe should give a separate indication. An open circuit usually leaves the probe in the same state, i.e. there is no distinction between an open-circuit condition and a 'bad' logic level. Similarly a short circuit to either of the supply lines will ensure a level that will be unambiguously interpreted as a true logic level. A voltmeter (or oscilloscope) will distinguish between a TTL logic/output and a short circuit to the supply line.

A single light-emitting diode with suitable driving stage can provide three states to represent the three groups of possibilities:

'off'	Logic 0 (or s/c to common)
'dim'	Open circuit or bad (intermediate between level 0 and 1)
'bright'	Logic 1 (or s/c to positive supply)

Fortunately we are not restricted to static testing. By adding a transient detector and a monostable circuit to the LED driver, an output pulse of fixed duration is obtained on an input transition. If the probe is connected to a point at logic 1 which experiences a narrow negative pulse of as little as 10ns, the LED is pulsed off for a fraction of a second. A second time delay inhibits its response to repetitive transitions such that a flashing output at  $\sim 10$  Hz is obtained for pulse rates from this frequency up to many MHz.

Thus with a single indicator LED, single short transients of either 0-1 or 1-0 can be detected as well as a continuous pulse train (Fig. 1.1).



**Fig. 1.1** Single indicator monitors static and dynamic logic states

High indicator on }	True logic 1
Low indicator off }	
Low on }	True logic 0
High off }	
Neither on	Bad or open circuit
Both flashing	Low frequency pulse train
Both dim	High frequency pulse train (relative brightness indicates mark/space ratio)

A third LED is added, flashing for a pulse train but off for any static condition. No information is provided on the mark-space ratio of any such

pulse train. If separate LEDs are used to indicate logic 0 and 1 levels with appropriate threshold detectors, more information can be provided (Fig. 1.2).

These alternatives can be extended by, for example, having a latched mode in which an LED is held on following the receipt of a single pulse, even of short duration. Each manufacturer adopts different specifications too for the pulse timings. The probes may respond to minimum pulse widths between 10 and 100ns with output pulse widths between 50 and 300ms. They may have switched threshold levels to make them compatible with different logic families. Check the manual to make sure you are getting the best possible use out of your probe.

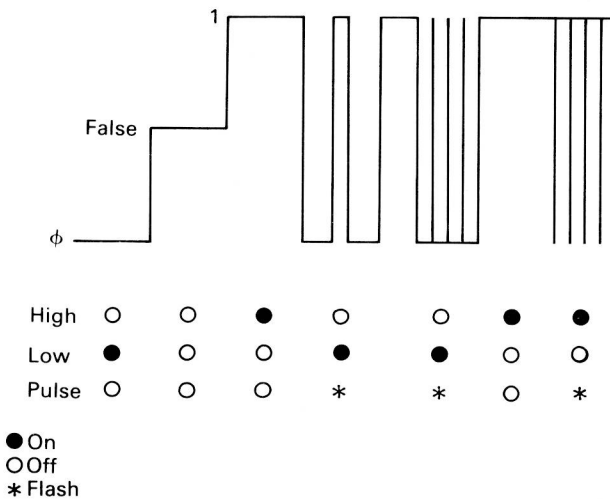


Fig. 1.2 Three LEDs give more information including mark-space ratio

## CURRENT PROBE

Some faults cannot be identified by voltage probes no matter how complex the probes become. A short circuit may be distinguishable at a given node, but not the particular device, connected to that node, which is causing it. Similarly, a short circuit between two points might be within an IC or between pairs of tracks serving these points. We need to know where the current flows are being diverted to, without breaking tracks or removing ICs if at all possible. Though in the last resort this often has to be done, it is both time consuming and risky. It is so easy to introduce other faults in the process.

To detect a direct current is difficult though possible: Hall-effect



devices are one possibility. It is difficult to achieve sensitivities appropriate to the levels in modern digital systems. Pulsed currents are somewhat easier and the current probe is a powerful tool. By aligning the tip of the probe along the conducting path, the field generated by the pulse is detected and the sensitivity is adjusted to keep an indicator on. The probe is then moved along the expected current paths until the indicator goes off, identifying the start of an unexpected conducting path. Note that it is a noncontact measurement, requiring close proximity to the conductor but not a direct electrical connection.

To see the different capabilities of voltage and current probes, consider Fig. 1.3. A pulse train at A should result in an inverted version at B, with the original recreated at D (ignoring propagation delay). No pulse activity should be present at C and E. A voltage probe at D might show no pulses, the output being at logic 1. This could be a fault in this gate, an output short circuited to the supply externally or that B is held at logic 0. Transfer the voltage probe to B. If a pulse train is present, that confirms a fault in the output gate or a short at D. No pulse train could mean:

- 1 the input inverter is faulty;
- 2 a short circuit at B to ground, supply line or other track.

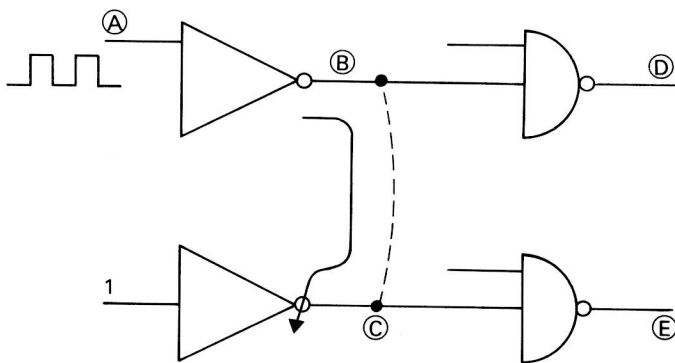


Fig. 1.3 Current probe traces short circuit current

If B is stuck at logic 0 this would check with the logic 1 state of D, suggesting the output gate as 'good'. A current probe at B should show a pulse train if the input inverter is functioning. Adjust the sensitivity so that the indicator is just on. Move the probe towards the output until the indicator goes out. This shows the point on the track at which a short circuit occurs. Inspection of the board should show up a solder bridge or other fault. Failing this, a check of neighboring points such as C might identify where the current is being diverted to, if not directly to ground, pinpointing the physical location of the dotted short circuit.

If a gate is used to drive multiple inputs the design will normally