SIGNATURE ANALYSER

by

S. N. Hettiwatte

Abstract

Analog circuits are tested by stimulating them with known test patterns and checking each node in the signal propagation path with an oscilloscope. If and when an unexpected response is detected, that part of the circuit is suspected and further investigated in order to locate the defective component.

Digital circuits however differ radically from analog circuits not only in the type of waveforms that are encountered but also they may be having more than one signal input. But if a digital system can be properly controlled so that its inputs form a well defined input pattern, then each and every node will be exercised through a fixed pattern of events. For each node, the same pattern will occur every time the stimulation program is run, and this can be used to verify the correctness of the node. If the measured pattern differs from the expected, then a fault can be inferred between it and the stimulation points of the system. Given a fixed test pattern, then for any node the pattern measured will be unique and can be used to check it on a go/no go basis. The term signature has been applied to the measured response of a node to a known test pattern, by analogy with the uniqueness of a person's hand written signature. This concept has led to an area called signature analysis.

1.0 Introduction

Common test equipment for analog circuit testing are the multimeter and the oscilloscope. For digital circuit testing common equipment are the logic probe, the logic pulser and the logic comparator. These hand held instruments are used either to excite or to check individual nodes in a digital circuit. An oscilloscope can be used to a limited extent to test digital circuits.

The hand held tools mentioned above play an important role when analysing faults in conventional digital circuits. However their application is limited in the case of microprocessor based bus structured systems where information appear on a large number of lines simultaneously and in a sequential form. To understand the operations that occur in a microcomputer system one has to have equipment which can capture and display, in a suitable form, the information on many lines, and have the ability to qualify the information he wishes to study clearly. Simple hand held tools cannot perform these functions and this has led to the development of test equipment specially designed to cope with the demands of troubleshooting in complex bus structured systems.

To address these needs, the technique of signature analysis and the signature analyser for component level troubleshooting of microprocessor based systems had been developed [5]. A signature analyser compresses a long digital bit sequence derived from a circuit node typically to a 16-bit signature, and displays it in a suitable format.

These signatures are unique to those nodes in the system if the system under test is excited by the same input. By comparing these actual signatures to the reference signature, a trouble-shooter can back trace to a faulty node. By designing or retrofitting signature analyser to digital products, a manufacturer can provide manufacturing test and field service procedures for component level repair, without dependence on expensive board exchange programmes [5].

2.0 Theory

If a digital circuit can be stimulated with a known repeatable test pattern, and if a node in this circuit generates a repeating binary sequence when the above test pattern is repeated, the repeating binary sequence at the node can be used to identify the node. This repeating binary sequence at a node may contain hundreds of bits. Therefore, to identify the node, there must be a method to identify the correct bit sequence which should occur at a node when the circuit is stimulated by the repeatable test pattern.

Basically there are two methods to identify a node. These are transition counting and cyclic redundancy checking (CRC). Signature analysis emulates transition counting, while using CRC.

Eng. S. N. Hettiwatte, Bsc Eng., AMIEE, AMIE(SL)
Lecturer in Electronic and Communicition Engineering,
Open University of Sri Lanka

2.1 Transition Counting

One established technique is simply to count the number of transitions of the signal from one logic state to the other, and use the final count value as an identifier for the node. This is known as transition counting.

The implementation of transition counting as a testing method must employ some form of instrument which will count every transition that occur between start and stop signals.

2.2 Cyclic Redundancy Check (CRC) Codes

CRC codes are used to detect single bit and multiple bit errors in high speed synchronous data links. To explain the CRC scheme, consider a bit stream polynomial B(X) and a generator polynomial G(X). If B(X) is divided by G(X), the quotient is Q(X) and the remainder is R(X).

$$B(X) = Q(X).G(X) + R(X)$$

In a CRC scheme the bit stream to be sent is divided by a generator polynomial and the remainder or the residue is appended onto the transmitted bit stream. At the receiving end, the incoming bit stream and residue are divided by the same polynomial used at the sending end.

If no error occurs during transmission, the remainder after this division must be all zeros. If not the position of 1's indicate where the errors has occurred in the transmitted bit stream during transmission.

If a repeating binary sequence at a node is divided by a generator polynomial, the remainder or the residue due to this division can be used as an identifier for that node.

3.0 Design

The basic block diagram of the design is shown in Figure 1. Referring to **Figure 1**, the gate will be opened by the rising edge of the start/stop signal. At this moment the clock signal is also input to the shift register through the control unit. When the gate is open the logic probe will input a serial bit stream to the shift register, which is a serial input parallel output (SIPO) shift register. The feedback network will implement the division of the bit stream by some characteristic polynomial. The gate will be closed by the trailing edge of the start/stop signal. This will also block the clock signal input to the shift register. After the gate is closed the residue in the shift register is the remainder due to the division. This remainder is the 16-bit signature. This signature is then latched and subsequently displayed using a suitable code with the aid of four seven-segment display units.

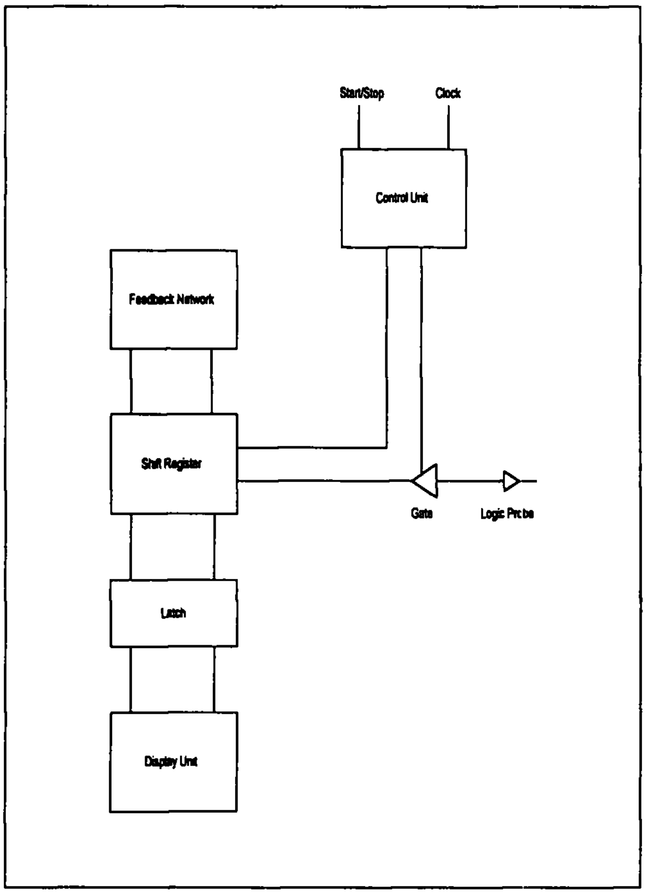

Fig 1: Block Diagram of the Signature Analyser

4.0 Implementation

4.1 Shift Register and Feedback Network

The 16-bit shift register, which is a 74673 has two registers. One is the 16-bit serial I/O shift register and the other one is the 16-bit parallel-out storage register. The shift register is configured to operate in serial input parallel output (SIPO) mode. The XOR feedback makes the final output dependent on the entire bit stream and the inputs of the XOR feedback are derived from 7, 9, 12 and 16th bits of the shift register and the output is again XORed with the input bit stream as shown in **Figure 2**.

4.2 Logic Probe

The circuit diagram of the logic probe constructed is given in **Figure 3**. As Figure 3 shows a buffer amplifier, a schmitt trigger and an inverter are used in the circuit. The buffer amplifier has a high input impedance which prevents the system being loaded and the schmitt trigger is used for pulse reshaping. The buffer amplifier is constructed using an LM 324 op-amp and the schmitt trigger is an 7413 inverting schmitt trigger. A 7400 inverter is used to get a non inverting output.

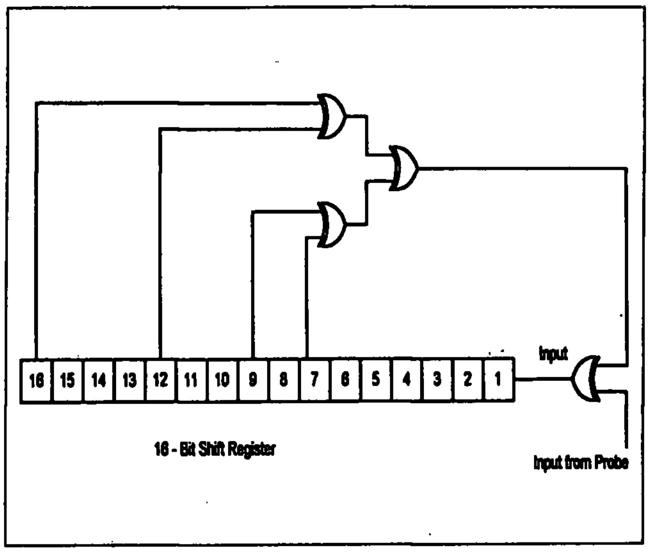

Fig 2 : Shift register with feedback

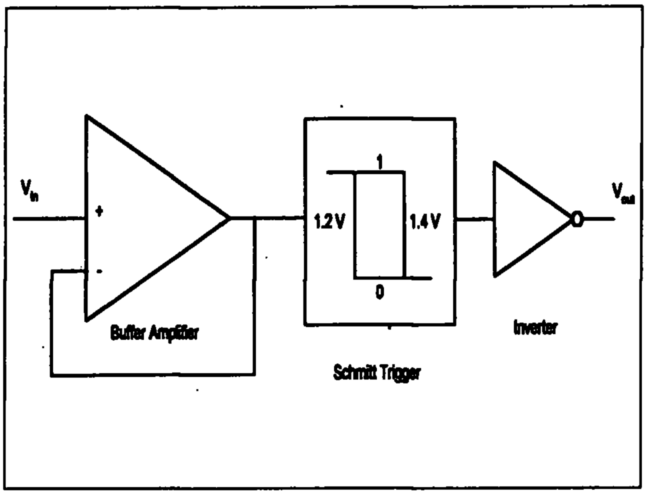

Fig 3 : Logic probe

4.3 The Control Unit

In order to get a stable and constant signature for a particular node the data stream allowed to pass into the shift register has to be properly controlled. That is the bit stream should enter the shift register at the start of the signature intake and the gate should close with the stop signal. The start/stop signal for the signature and the clock signal for the shift register are derived from the system under test (SUT). The gate control and other control signals required by the shift register are derived by the control unit shown in **Figure 4.**

The operation of this unit can be briefly described as follows:

When the reset switch is pressed before taking a signature, it sets the flip flop 1 (ff1). Thus the Q output of ff1 is at logic 1 and so do the D inputs of flip flop 2 (ff2) and flip flop 3 (ff3). It also clears the Q outputs of ff2 and ff3. The pressing of the reset switch also clears the parallel-out storage register of the 74673 shift register. This is due to the application of logic 0 on STMR* (Store Master Reset Input) pin of the shift register.

With the arrival of the rising edge of the start/stop signal, the input D in ff2 is transferred to its output while ff3 is not affected. This will make the output of the XOR gate logic 1. This allows the clock from SUT to pass into the shift register; sets the shift register to WRITE mode and the logic probe input to pass into the shift register through the gate.

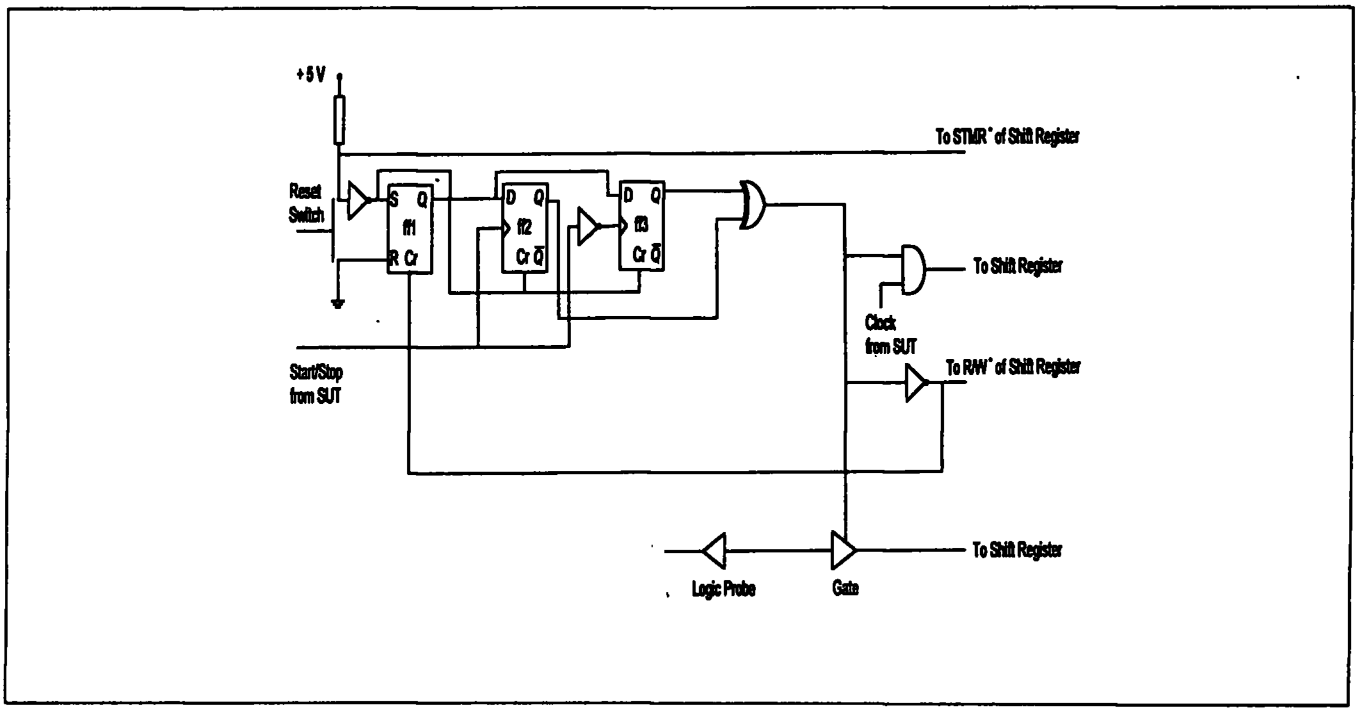

Fig 4 : The Control Unit

Then with the arrival of the trailing edge of the start/stop signal, the input D in ff3 is transferred to its output while ff2 is not affected. This will make the output of the XOR gate logic 0. This blocks the clock from the SUT; sets the shift register to READ mode and blocks the logic probe input to the shift register through the gate. Thus a stable signature will remain in the shift register, which will be held by the 74273 latches, ready to be displayed.

4.4 The Display Unit

The display unit, shown in **Figure 5** is a multiplexed display unit whose clock runs asynchronously with the system clock. The outputs from the latches are input to the 16-input 4-output multiplexer designed with 7408 AND gates and 7432 OR gates. These 4 outputs are input to the 2716 EPROM decoder, which has a 2 k ¥ 8 memory.

The EPROM decoder is programmed to decode the 4 inputs to either hexadecimal format or to the Hewlett Packard display code [2], given in **Table 1**, which can be selected by the output character selector switch. The outputs of the decoder are applied to the seven segment displays via 7416 inverter drivers.

The display clock is designed using a 555 timer. The frequency of the clock is set high enough to make the display appear consistent. The decoder selects the multiplexer input port and the corresponding seven segment display together. So the correct digit is displayed at the correct instant.

5.0 Error Detection in Computer Systems Using Signature Analysis

5.1 Testing of System Kernel

The microprocessor and its associated ICs of a computer system are known as its kernel. This kernel can be tested by taking signatures from nodes in a free running mode.

The computer is set to free-run mode by forcing the CPU to obey the same instruction, for example NOP (No operation) instruction. This is implemented by clamping a fixed instruction op-code into the data pins of the CPU [3]. This does nothing but increase the program counter by one every time. When the last address in the memory map is read the program counter automatically gets reset and this process continues. This type of activity is ideal for signature analysis because of the repetitive nature and every possible bit pattern applied to the address bus during this cycle.

During free running the most significant address bus line will be low for half the total number of addresses and high for the second half. So this signal is ideal for start/stop signal needed by the signature analyser. For the clock signal READ line can be used since in the free running mode all the operations are merely read operations and READ signal will oscillate as each and every address is put into the address bus.

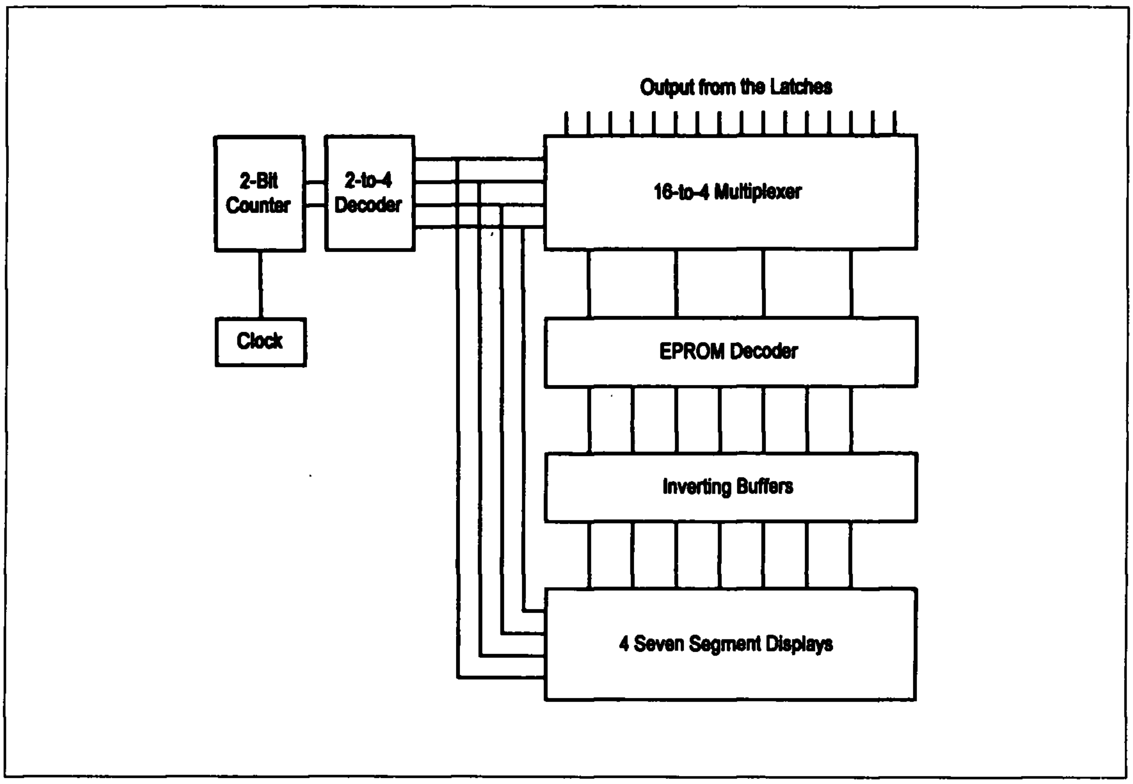

Fig 5: The Display Unit

Table 1- Display Codes

Binary Input	Hexadecimal Code	Hewlett - Packard Display Code
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	A
1011	b	C
1100	C	F
1101	d	H
1110	E	P
1111	F	U

5.2 ROM Testing During Free Run

Even though the primary function of the free run testing is to exercise the system kernel, it is possible to test any ROM during free run testing. Since during free run operation each and every memory location is read and ROM contains fixed instruction in them, it will form a fixed repeatable loop. Hence it can be compressed to a unique signature by selecting proper start/stop and clock signals.

Before a ROM chip is read, its CE (Chip Enable) line is activated. So it is possible to use this line as the start/stop signal and since READ is enabled when each and every location is read, READ line can be used as the clock.

5.3 ROM Testing Using a Test Loop

ROM can also be tested by running a ROM test loop instead of the free running test. In this form of testing the start/stop signal is also generated by the software.

5.4 RAM Testing Loop

Unlike ROM there is no fixed bit pattern stored in RAM. So when we want to test with the signature analyser we want to first store some fixed bit patterns in the RAM and try to get the signatures. For example, storing '00H' and 'FFH' in alternate RAM locations.

5.5 Input / Output Testing

Input / Output ports can also be tested as ROMs and RAMs are tested.

6.0 Conclusion

The primary aim of this project was to design and implement a signature analyser which can match with the requirements of such an equipment for test purposes.

Although the signature analyser designed differs from the commercially available ones, it is capable of performing basic functions with good accuracy and it is less expensive. This signature analyser was implemented using components available in the local market.

7.0 Acknowledgements

The author wishes to express his sincere appreciation to the project supervisor, Dr. G. S. Atukorala, Senior Lecturer, Department of Electronic and Telecommunication Engineering, University of Moratuwa, without whose guidance this project would not have been a success.

The author also would like to thank Mr. Ajith Pasqual and Mr. Amith Munindradasa, both of whom are lecturers attached to the Department of Electronic and Telecommunication Engineering, University of Moratuwa, for their invaluable help in carrying out this project.

Finally the author wishes to thank his colleagues and project team members Ms. Devika Bandara, Mr. M. F. Mohamed Firdhous and Mr. A. W. I. Mendis, for their support in this group work.

8.0 References

(1) Brey B. B., " The Z-80 Microprocessor Hardware, Software, Programming and Interfacing ", Prentice-Hall, 1988.

(2) Hewlett-Packard, " Guidelines for Signature Analysis" , Application Note 222-4, July 1990.

(3) Holland R., " Microcomputer Fault Finding and Design ", Macmillan, 1991, pp. 79 - 86.

(4) Middleton G., " New Digital Troubleshooting Techniques - A Complete Illustrated Guide ", Prentice-Hall, 1984, pp. 225 - 252.

(5) Williams G. B., " Troubleshooting on Microprocessor Based Systems ", Pergamon Press, 1984, pp. 131 - 162.

(6) Tooly M., " Servicing Personnel Computers ", 3 Ed., Newnes, 1992.