

**DESIGNING TESTABILITY INTO
AN EXISTING MICROPROCESSOR BOARD**

by

MICHAEL CRAIG PAULSON

B.S., Electrical Engineering, Kansas State University, 1988

B.S., Computer Engineering, Kansas State University, 1989

A MASTER'S THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

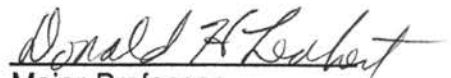
Electrical Engineering

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1992

Approved by:


Major Professor

LD
2668
T4
EECE
1992
P38

Abstract



A11210 189661

This thesis presents the hardware implementation of a microprocessor system built according to the microprocessor-controlled Built-In Self-Test (BIST) techniques as presented by Gordon [Gordon 1991a]. It covers issues relating to the isolation of the edge connector and secondary board areas, to the use and function of a field-programmable gate array (Logic Cell Array)TM, to the implementation of electronic wrap-arounds, and to the general design of the software to support the testability features. In addition, it covers the general use of the ANSI/IEEE Std. 1149.1 test bus for diagnosis, isolation, and board-level partitioning.

Dedications

I wish to acknowledge the contributions of the people who provided me with the support I needed to undertake (and finish) a project of this magnitude. This thesis is dedicated to:

Dr. Don Lenhert for his constant advice both throughout graduate school and during this project. His contributions cannot be overstated. I would especially like to thank him for steering me in the "write" direction.

Dr. Dwight Gordon was also invaluable during the course of this project. He helped me get a conference paper published, always answered my (e-mail submitted) questions, provided a great deal of editing support, and graciously accepted the LCA programming responsibility. I was also able to "draw" on his timing diagram experience to finish this project.

James Lebak, Tom Asay, Rick Dempster, and too many other nameless graduate students to mention, for an enjoyable couple of years.

Finally, and most importantly, I wish to gratefully acknowledge the constant support of my wife and friend Dawn, who has endured my educational persistence. Her sacrifices for my education and this thesis have been greater than my own.

Acknowledgements

Funding for this project was made possible by the Advanced Manufacturing Institute at KSU, under a grant from the Kansas Technological Enterprise Corporation (KTEC).

In addition, Texas Instruments, Inc. graciously donated a complete ASSET development system that is used to provide support for off-board testing.

Trademarks

IBM and PC-AT are trademarks of IBM, Inc.

PAL is a trademark of Monolithic Memories, Inc.

SCOPE and ASSET are trademarks of Texas Instruments, Inc.

SUPERmon is a trademark of Motorola, Inc.

XILINX, Logic Cell Array, and LCA are trademarks of Xilinx, Inc.

Table of Contents

Abstract	i
Dedications	ii
Chapter 1 Introduction	1
Chapter 2 Background	3
Current System Design	3
Testability Study	4
Test Background	5
Testability Concepts	6
Test Problems and the Benefits of DFT	7
ANSI/IEEE Std. 1149.1-1990	10
History	10
Brief Description	10
Fault Detection	16
Chapter 3 Microprocessor Systems	18
Hardware Goals and Constraints	18
Functional Compatibility	19
Mechanical Constraints	20
System Designs	22
Core Components	22
Microprocessor	24
System Components	24
Memory	24
Parallel Port	27
Serial Ports	28
Timer IC	31
Programmable Logic	32
Memory Map	33
Timing Analysis	35
Edge Connectors	36
Chapter 4 Testability	41
Hardware and Software Components	41
SCOPE Octals	41
TBC	44
TBC Description and Use	44
TBC Design Problems	45
Board Partitioning	47
Primary/Secondary Areas	47
System Busses	48
System Area Tests	50

Test Control via LCA	51
LCA Description	52
LCA Uses	52
LCA Configuration	60
Method	60
Problems	61
DFT Structures	66
Test Bus	66
Host Microprocessor Test Bus Control	67
TBC Interface	67
Edge Connector	73
External Test Bus Control (ASSET)	74
Buffering of Test Bus Signals	78
Electronic Wrap-arounds	79
Serial Wrap-Around Test Structures	80
Parallel Wrap-Around Test Structures	82
Programmable Timer Test Structures	85
Test LED	87
Test Software Support	89
 Chapter 5 DFT Costs and Savings	 91
 Chapter 6 Future Work and Lessons Learned	 95
 Chapter 7 Conclusions	 97
 References and Bibliography	 99

List of Appendices

Appendix A: Nomenclature	102
Appendix B: EECE 631 Hardware Reference Manual	109
Appendix C: Traditional System Support Information	144
Appendix D: Traditional Board Schematics	152
Appendix E: DFT System Support Information	157
Appendix F: DFT Board Schematics	169
Appendix G: Timing Diagrams & Equations for Design Verification	174
Appendix H: DFT PWB Specifications, Layout, and Artwork	191
Appendix I: SCHEMA II+ Library Instructions	198

List of Figures

Figure 1. Manufacturing Fault Spectrum. Adapted from [HP 1990].	9
Figure 2. IEEE 1149.1 Components and Typical Manufacturing Defects.	12
Figure 3. A Boundary-Scan Board Design. Adapted from [IEEE 1990].	14
Figure 4. Serial Connection Using One TMS Signal. Adapted from [IEEE 1990]. . .	16
Figure 5. Traditional and DFT Microprocessor System Design Origins.	19
Figure 6. Traditional Microprocessor System Block Diagram.	23
Figure 7. Traditional ACIA/RS-232 Interface Connections.	30
Figure 8. Strappable Serial Power Connections.	31
Figure 9. DFT Microprocessor System Block Diagram.	42
Figure 10. TBC's Vcc and Ground PWB Traces.	47
Figure 11. SCOPE Octal Control Signal Groupings.	50
Figure 12. LCA Block Diagram.	54
Figure 13. TBC Read/Write Interface.	68
Figure 14. TBC Read Cycle Timings.	71
Figure 15. TBC Write Cycle Timings.	72
Figure 16. Isolating SCOPE buffers around the edge connector.	74
Figure 17. Block Diagram of Test Bus Connections.	75
Figure 18. Uni- and Bi-directional Signal Wrap-Around Techniques.	81
Figure 19. PIA Control Line Wrap-Around Techniques.	83
Figure 20. Visibility and Control of the Programmable Timer Signals.	86
Figure 21. DFT Test LED Connections.	88
Figure 22. Traditional Microprocessor System Block Diagram.	110
Figure 23. DFT Microprocessor System Block Diagram.	111
Figure 24. Uni- and Bi-directional Signal Wrap-Around Techniques.	118
Figure 25. Generic System Block Diagram for Timing Analysis.	175
Figure 26. MC6809/6800 Peripheral Write Timing Diagram.	187
Figure 27. MC6809/6800 Peripheral Read Timing Diagram.	188
Figure 28. MC6809/SRAM Write Timing Diagram.	189
Figure 29. MC6809/SRAM (EPROM) Read Timing Diagram.	190
Figure 30. PWB Dimensions in mils (thousands of an inch).	193
Figure 31. DFT System PWB Assembly Drawing.	194
Figure 32. DFT System PWB Artwork, Layer 1 (Component Side).	195
Figure 33. DFT System PWB Artwork, Layer 2 (Non-component Side).	196
Figure 34. DFT System PWB Artwork, All Features (Both Sides).	197
Figure 35. Suggested Placement of SCHEMA II+ Library Part Text.	200

List of Tables

Table 1. Packaging Technology versus Pin-to-Pin Spacing.	8
Table 2. IEEE 1149.1 Test Bus Signals.	12
Table 3. Sample IEEE 1149.1 Test Procedure using <i>EXTTEST</i>	15
Table 4. RAM/ROM Pinout Differences/Similarities.	26
Table 5. System Memory Maps. Adapted from [Lenhert and Gordon 1990].	34
Table 6. DFT System Connector Descriptions.	37
Table 7. System Connector Specifications.	38
Table 8. Equivalent Expansion Edge Connector Signals.	39
Table 9. Peripheral Register Select/Address Combinations.	60
Table 10. DFT System Reset Sequence.	65
Table 11. TBC Read Procedure.	70
Table 12. TBC Write Procedure.	70
Table 13. On-board Test Bus Connections.	76
Table 14. Off-board Test Bus Connections.	77
Table 15. SCOPE Octal Peripheral Test Control Signals.	85
Table 16. Signal Name Prefix Descriptions For Both Microprocessor Systems.	103
Table 17. Traditional and DFT System Specs. Adapted from [Motorola 1982].	121
Table 18. System Memory Maps. Adapted from [Lenhert and Gordon 1990].	123
Table 19. DFT System Connector Descriptions.	124
Table 20. System Connector Specifications.	124
Table 21. Connector J1 Serial Interface Signals.	126
Table 22. Connector J2 Serial Interface Signals.	127
Table 23. Connector J3 Test Interface Signals (DFT System Only).	128
Table 24. Connector J4 Timer Interface Signals.	129
Table 25. Connector J5 Parallel Interface Signals.	130
Table 26. Expansion Edge Connector P1 Interface Signals.	132
Table 27. LCA Pin Signals.	136
Table 28. PAL U15 Pin Signals.	142
Table 29. PAL U10 Pin Signals.	143
Table 30. General Peripheral Signal Descriptions.	176
Table 31. Interface Delay Terminology.	178
Table 32. Interface Delays.	178
Table 33. PWB Artwork Specifications.	192

Chapter 1 Introduction

With the current industry trend to increase microprocessor hardware complexity, a need for an efficient method of testing system functionality is needed. This greater hardware complexity and component density increases the difficulty of system diagnosis (both at the manufacturing and user levels). Traditional approaches to system test either ignore this test need or they address it through the addition of a dedicated test processor--thereby increasing once again the system hardware complexity. This project's approach is to incorporate test features at the very beginning of the design phase and to use the same host microprocessor to test its own functionality by exercising these test features.

The overall project goal is to effect an economic study of the effects of testability on a small microprocessor system. To help achieve this goal a microprocessor system has been designed which incorporates Design For Testability (DFT) and microprocessor controlled self-testability capabilities according to the techniques as presented by Gordon [Gordon 1991a]. This includes the use of partitioning logic to isolate high failure areas such as edge connectors, the implementation of electronic wrap-arounds, and the hardware support for software testing. Specific Ad hoc DFT techniques and the ANSI/IEEE Std. 1149.1 test bus were used to insure controllability and observability at several key points on the board and to provide hardware support for fault isolation at the component level [Abramovici, Breuer, Friedman 1990]. Support for many of these features is contained within a field-programmable gate array (Logic Cell Array) and an on-board IEEE 1149.1 test bus controller.

It is the intent of this thesis to present the hardware implementation of these DFT investments and to contrast the design results with a traditional microprocessor system

that was also designed. Since the project is ongoing, most DFT maintenance benefits will not be seen for some time, as such, few comments can be made about the long-term benefits.

Test software was written to exercise the test features [Lebak 1991]. Together, the hardware and software form the basis for the testability analysis study at Kansas State University (KSU) where the benefits of DFT can be investigated. The idea is to invest enough in DFT so that test benefits are realized, while not increasing the product's development cost to the point that maintenance savings are offset by these additional DFT structures. The benefits derived from DFT is often a much debated topic between test engineers, and designers and management--test engineers always wanting more, and design engineers as well as management always wanting less.

Chapter 2 gives a background on testability, the IEEE 1149.1 standard, and the category of faults that the DFT system can be expected to be detect. Readers with much experience in this area may elect to skip this section. Chapter 3 discusses the hardware objectives of this project and the methods used to achieve them. It also discusses the two microprocessor systems that were designed (the DFT system and a control or traditional microprocessor design) and highlights the design features common to both systems. Chapter 4 focuses on the DFT system design, including test components and structures, test control, and a brief description of the test software. Initial DFT comparisons between the designs are made in Chapter 5. Finally, future work is proposed in Chapter 6. The appendixes cover the nomenclature used throughout this thesis, a hardware reference manual (for the EECE 631 students), a timing analysis of both systems, DFT and traditional system schematics and support information, and finally, the Printed Wire Board (PWB) artwork used in the layout of the DFT system.

Chapter 2 Background

Throughout this document references are made to three completely different microprocessor systems: KSU's current microprocessor system, a system designed with test features, and a system designed without test features. The second and third designs are the primary topics of this thesis and will be discussed in detail throughout this document.

2.1 Current System Design

The current microprocessor system at KSU comprises two microprocessor boards that are electrically connected to one another via expansion edge connectors and a card cage backplane. For description purposes, the PWB that houses the microprocessor and most of the peripherals will be referred to as the main board, and the other board will be referred to as the auxiliary board.

The main microprocessor system boards are either a Motorola MM17 or MM19 board [Motorola 1982]. They consist of a Motorola MC6809 microprocessor, 8-bit addressable RAM and ROM, and components from Motorola's 6800 family of peripherals with no inherent testability features. This list of peripherals includes Motorola's Peripheral Interface Adapter (PIA) to provide parallel I/O capabilities, an Asynchronous Communications Interface Adapter (ACIA) to provide serial I/O capabilities, and a Programmable Timer Module (PTM). Each of these components is discussed in more detail in section 3.2 entitled *System Designs*.

The auxiliary board is a wire-wrapped board designed specifically for KSU's needs. It consists of an additional user PIA, buffered expansion edge connector signals (taken directly from the card cage backplane), and a serial communication link to mass storage

for user programs and projects. Since all of the laboratory systems are placed in a card cage system, the expansion edge connector signals would be unavailable if the auxiliary board did not provide them.

KSU uses these systems in an upper-level undergraduate and graduate-student microprocessor design class (EECE 631) to investigate microprocessor operation. It is used as a platform from which student hardware and software projects can be constructed. Often projects and subsequent debugging sessions lead to component or system failures on the host microprocessor system. The current system at KSU has extremely limited self-test capability so KSU students are unable to ascertain the origin of any system errors--their project or the lab equipment. By monitoring the design process as well as the use of the microcomputers, a microprocessor system with Built-In Self-Testability (BIST) may provide an excellent environment for a testability analysis study.

2.2 Testability Study

To pursue this testability analysis study, a microprocessor hardware design using DFT and BIST techniques was designed and the PWB artwork for fabrication was laid out. This was the first step in pursuing the testability analysis study. The redesigned system is upward compatible with the existing systems, both in terms of address map and I/O capabilities. Compatibility between these systems preserves KSU's large base of software and hardware peripherals, as well as course-development materials. This presents some difficulties that would not necessarily be present in a totally arbitrary system design. None of the original components contain any support structures for testability. Furthermore the current board designs are based on 10-year old technology. Use of

them in the testability study of the economics of DFT would produce a decided bias just from the technological improvements over the last 10 years.

To eliminate this technological bias a control system or traditional microprocessor system was designed which featured traditional components lacking any inherent test capability. This design will be used to contrast results obtained from the testability design. For convenience, this baseline design or microprocessor board designed for control purposes will herein be referred to as the traditional design, and the testability design will be known as the Design For Testability (DFT) design. A more detailed discussion of these two systems can be found in Chapter 3 entitled *Microprocessor Systems*.

Software to control these test features has been written [Lebak 1991]. The goal of the software was to provide test routines that could effectively diagnose on-board or off-board component failures. This would allow a user to determine whether his additions to the microprocessor system (via edge connector signals) were the cause of the failure or if the microprocessor system itself was at fault. On-board component failures would be detected and isolated to the component or cluster level.

2.3 Test Background

Scores of material have been written on the subject of Design For Testability (DFT). However, this section attempts to introduce some of the concepts upon which the hardware test features described in this thesis are based. General test concepts and problems are presented. A brief introduction into the digital test bus standard that was incorporated into the DFT system is also included.

2.3.1 Testability Concepts

Much of the material presented here was taken from *Digital Systems Testing and Testable Design* [Abramovici, Breuer, and Friedman 1990]. Readers wishing to pursue more information on the testability concepts presented here can consult this reference.

A *test* is an experiment to see if a product or device functions as intended. If it does not, a further diagnosis may be required to reveal or *detect* the cause of the failure. Once the exact failure is detected, *fault isolation* can determine which component or group of components is at fault. If testability is designed into the component or system, then fault detection and isolation become that much easier.

Testability is a design philosophy that influences the costs associated with testing. It is a function of product controllability, observability and predictability. The ability to control or establish a known signal value at a given circuit node is known as *controllability*. *Observability* is the ability to determine a circuit node's value and *predictability* is a measure of the circuit's expected behavior with respect to its actual behavior given a unique set of input stimulus. Controllability and observability have traditionally been achieved by directly applying and observing a circuit's stimulus and response.

To test a product, one must control primary inputs (controllability) and propagate these values to the primary outputs (observability). When the predicted response (predictability) corresponds with the product's desired functionality, the product is said to be good. If there is no means of controlling internal inputs or observing internal outputs, then there is no method of empirically determining product predictability [TI 1989].

Structures specifically put in place to ensure that a device is testable are known as Design For Test (DFT) techniques. The DFT system incorporates many design for test features such as:

- test points,
- component initialization,
- and the partitioning of large circuits into smaller ones.

2.3.2 Test Problems and the Benefits of DFT

Initially, DFT techniques place additional constraints on the designer--the designer must allocate additional board space for the test structures. Ironically, the addition of more hardware, reduces overall product reliability. In addition, DFT designs require additional design time, and in some cases, additional training. This translates into larger development costs. However, DFT techniques attempt to reduce test costs by investing in test structures that improve product controllability, observability, and predictability.

Despite these disadvantages, DFT techniques are becoming increasingly important since technological developments are escalating test generation and application costs. This decrease in product testability can be attributed to:

- an increase in the complexity of integrated circuits;
- higher packaging densities such as Surface Mount Technology (SMT);
- double-sided and multi-layer printed circuit boards; and
- finer board pitch etch.

Current board test methods use "bed-of-nails" test fixtures to induce and sense signal voltages. This approach has traditionally been used since adequate probe space has been available. However, as indicated in Table 1, IC pin spacing has increased over the

years. Guided test probes require pad sizes of no less than 0.030" to make reliable electrical contacts [Teradyne 1991]. Techniques used to provide test fixtures of this size are often expensive. In some cases they use the additional real-estate that was gained by using a higher package density in the first place. Furthermore, a "bed-of-nails" test approach is invasive with respect to surface mount devices.

Table 1. Packaging Technology versus Pin-to-Pin Spacing.

Packaging Technology	Pin-to-Pin Spacing
DIP, PGA	0.100"
SOIC	0.050"
PQFP, CQFP	0.025"
TAB	0.008"

These developments place an increased burden on the application of tests. Furthermore, traditional product development activities are partitioned into separate design, test, and manufacturing phases. Thus, nonstructured test methods are often required to verify products since a designer considers his work "finished" when the functional requirements of the design are met, i.e. little or no thought is given to product testability. Worse yet are those designs that are so complicated that complete functional testing is not possible.

Test application is only half of the problem--the other half's being the costs associated with testing. Automatic Test Equipment (ATE) is expensive and becomes obsolete after a few years. Test costs may exceed design costs due to the costs of test pattern generation; fault simulation, detection, and isolation. Finding faults early in a system's life cycle can decrease the costs associated with testing, i.e. "identifying a faulty chip using

board level tests is usually 10 to 20 times more costly than testing a chip" [Abramovici, Breuer, and Friedman 1990, p. 345].

To address the rapidly increasing costs associated with testing and to provide a structured approach to chip-level testability, the IEEE 1149.1 digital test standard was adopted to standardize a digital IC test architecture. Although not a complete solution, this architecture was intended to address many of the typical manufacturing faults that are often experienced. Manufacturing fault data collected from the Colorado Surface Mount Center at Hewlett Packard is shown in Figure 1. It clearly illustrates how almost three quarters of the manufacturing fault spectrum can be attributed to interconnect problems [HP 1990]--i.e. exactly the types of problems the IEEE 1149.1 standard was designed to address.

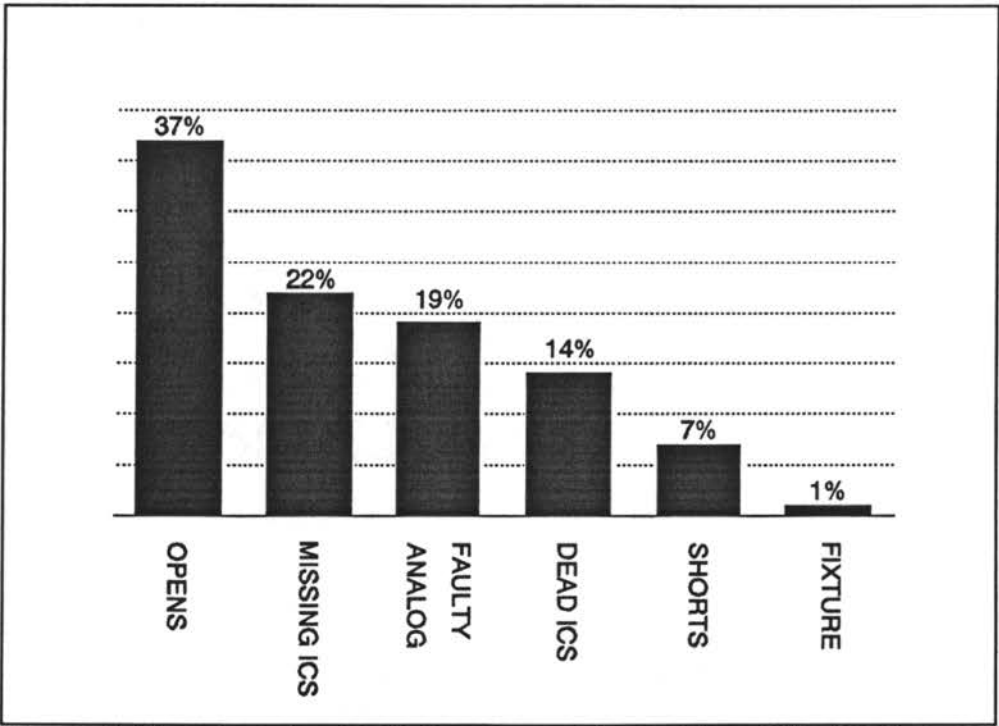


Figure 1. Manufacturing Fault Spectrum. Adapted from [HP 1990].

2.3.3 ANSI/IEEE Std. 1149.1-1990

2.3.3.1 History

The ANSI/IEEE Standard 1149.1-1990 (herein referred to as IEEE 1149.1, boundary-scan, or simply 1149.1) was originally conceived in 1985 by a consortium of European companies who collectively called themselves the Joint European Test Action Group (JETAG). The idea was to develop an industry wide test standard that could address two of the major test problems facing the electronics industry: decreasing test node access and increasing device complexity. Decreased nodal access means less points on which to place guided test probes, and increasing device complexity translates to more costly or complex tests to identify problems.

In 1986, several U.S. companies joined the consortium and the group became known simply as JTAG. By 1988, the JTAG committee had outlined a digital test technique and this technique was submitted to the IEEE, under the auspices of the Computer Society for refinement and implementation as a standard. In February 1990, the proposal became the IEEE Standard 1149.1-1990, *Standard Test Access Port and Boundary-Scan Architecture*. This standard defines a test architecture that can be incorporated into digital devices to simplify the test process and provide a means of controlling and observing values at an IC's digital boundary (pins).

2.3.3.2 Brief Description

The standard's primary objective is to provide support for the detection and isolation of manufacturing defects like those shown below in Figure 2. This list of defects includes shorts (like solder bridges), opens, broken bond wires, Stuck-At-Ones (S-A-1), and Stuck-At-Zeros (S-A-0). The standard's objectives are:

- to test PWB (or other substrate) interconnections between integrated circuits;
- to test the integrated circuit itself; and
- to observe or modify circuit activity during component operation [IEEE 1990, p. 1-1].

To declare 1149.1 conformance, a device must contain a Test Access Port (TAP), a TAP controller, an instruction register, and two special data registers. Furthermore, the standard specifies register lengths, and three required test instructions.

A simplified 1149.1 conformable component is depicted in Figure 2. The TAP, TAP controller and mandatory registers are all illustrated. Two of the mandatory registers are a one-bit Bypass register (BP), and an Instruction Register (IR) of no less than two bits (two bits are necessary to decode the number of required instructions). In addition, a Boundary-Scan Cell (BSC) for each I/O pin on the IC is required. These cells are collectively called the Boundary Scan Register (BSR).

The TAP is a four, or optionally five-wire serial test bus that is controlled via a 16-state state machine known as the TAP controller. These five TAP signals and descriptions are shown in Table 2. The TAP signals and test logic operate independent of the device's normal core logic.

The TAP controller states are changed based on the value of the TMS line at the rising edge of the TCK. The state machine provides two distinct state sequences. One sequence defines the operations that affect the instruction register, and the other sequence defines operations that affect the data registers. Six stable states (i.e. states that the TAP controller can remain in based on a static value of TMS) are provided, of which two affect the test logic itself [IEEE 1990, Figure 5-1].

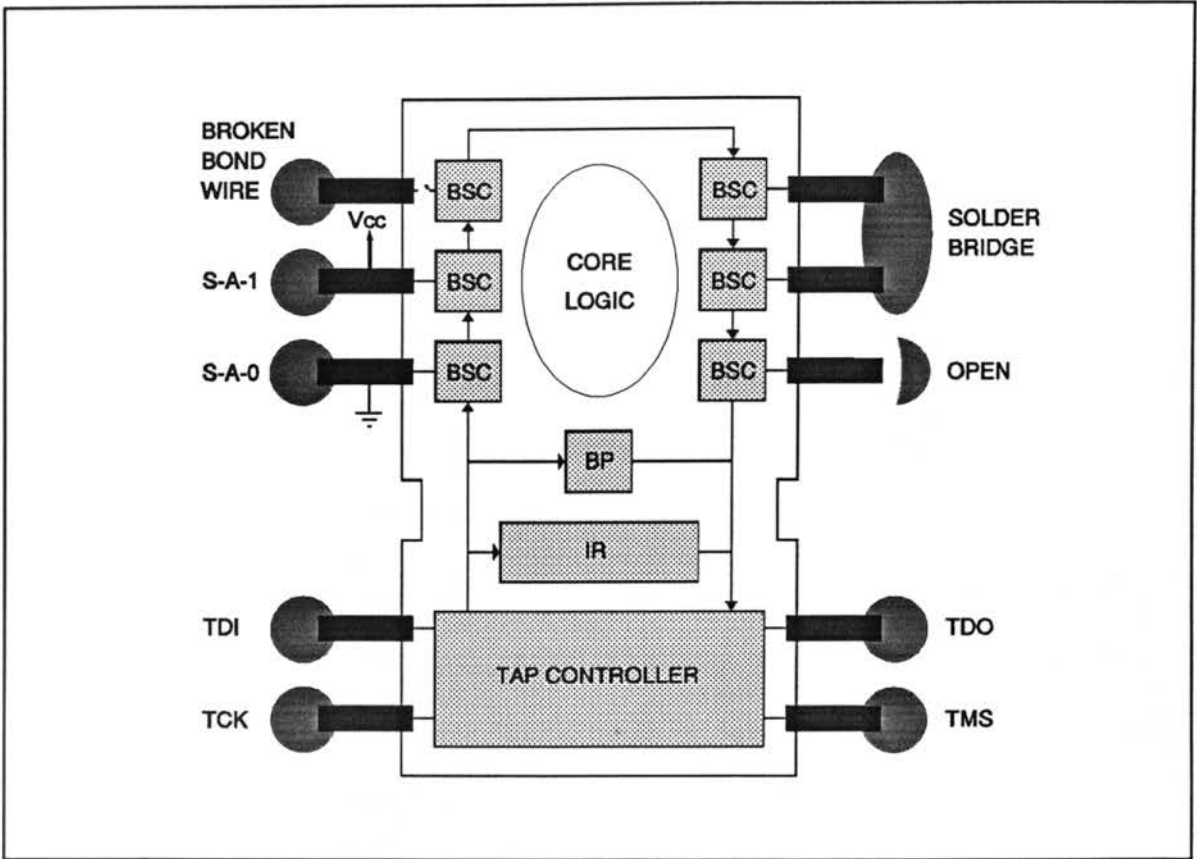


Figure 2. IEEE 1149.1 Components and Typical Manufacturing Defects.

Table 2. IEEE 1149.1 Test Bus Signals.

TAP Signals	Signal Description
Test Data Input (TDI)	Serial Test Input for Instructions and Data
Test Data Output (TDO)	Serial Test Output for Instructions and Data
Test Mode Select (TMS)	Broadcast Signal to Control the TAP Controller
Test Clock (TCK)	Broadcast Signal to Synchronize Serial Shifts
Test Reset (TRST*)	Optional Active-low Signal to Reset Test Logic

TAP controller states include the *Pause-IR* and *Pause-DR* states and the *Test-Logic-Reset* state. The pause states allow a temporary halt in the serial shifting of instruction or data. This state allows a test bus controller to reload serial data. *Test-Logic-Reset* is

provided to disable the test logic and allow the device's normal operations to take place.

The standard specifies three possible methods to reach *Test-Logic-Reset*¹:

1. Power-up
2. Assert the optional TRST*
3. TMS = 1 for 5 TCKs

To be compliant, devices must also incorporate three mandatory instructions: *BYPASS*, *SAMPLE/PRELOAD*, and *EXTEST*. Together these instructions permit visibility and controllability of an IC's periphery. The *BYPASS* instruction bypasses the sometimes lengthy BSR by placing a single-bit register directly between TDI and TDO. The *EXTEST* instruction is used to control the device's output lines by selecting the BSR in the scan path and sourcing the BSC data out of the device. The *SAMPLE/PRELOAD* instruction is used to sample or observe data at the inputs to a device by selecting the BSR and sampling the input data. Theoretically, if all of a PWB's system components were 1149.1 compliant, then no guided probe tests would be necessary, since visibility and control could be achieved via the test bus.

A 100% scannable board is shown in Figure 3. It shows how several 1149.1 compliant components can be interconnected to form a shift-register chain around the periphery of the scannable components. The *EXTEST* instruction could be used to transmit data from one BSC to another. Data could then be sampled via the *SAMPLE/PRELOAD* instruction and bit comparisons between the transmitted and received data could indicate that manufacturing defects like those illustrated in Figure 2 were present.

¹ The standard specifies that all 1149.1 compliant devices that do not have the optional TRST* line will have a power-up reset circuit that forces the TAP controller into *Test-Logic-Reset*.

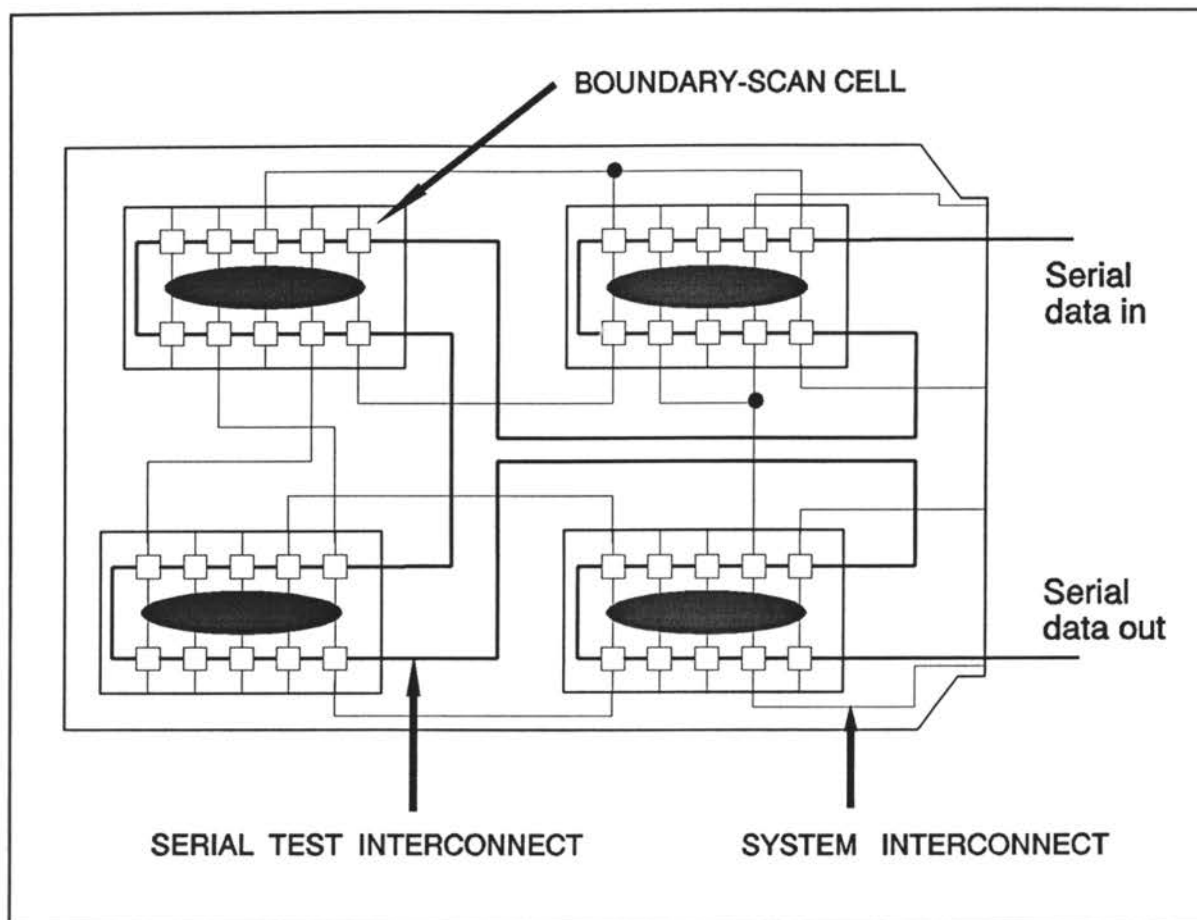


Figure 3. A Boundary-Scan Board Design. Adapted from [IEEE 1990].

Due to the design nature of boundary-scan, the normal device's functions can be overridden by the test logic, i.e. *EXTEST* can be used to drive a component's pins independent of the device's actual functions. In extreme cases, this feature could be used to emulate a microprocessor's operation to other peripherals, provided the processor was completely isolated via boundary-scan components. However, this feature lends itself to misuse in that designers could potentially use this feature to enhance component functionality. This of course, would be a violation of the 1149.1 standard because the test bus could not test functions it was being used to perform.

To prevent the test logic from altering the device's core logic functions inadvertently, several test design measures were instigated. The test logic contains two internal pull-up

(PU) resistors: one on the TDI line, and one on the TMS line. Furthermore, only one stable state exists for the case when TMS is a logic one: the *Test-Logic-Reset* state. Together these features prevent a device's test logic from altering the device's normal functions--unless that was the intent.

The IR defines test operations that will be performed and/or the data registers that will be accessed during this test operation. Instructions are serially shifted into the IR when the TAP controller is placed in the *Shift-IR* state--known as an IR-Scan. These instructions are then decoded when the TAP controller passes through the *Update-IR* state. The corresponding data register states are *Shift-DR* and *Update-DR* which operate in a similar fashion to their IR-Scan counterparts. As depicted in Table 3, these states and the *EXTEST* instruction can be used to control an 1149.1 compliant's IC pins.

Table 3. Sample IEEE 1149.1 Test Procedure using *EXTEST*.

-
- The TAP controller is placed in the *Shift-IR* state via TMS.
 - An *EXTEST* instruction is shifted into the Instruction Register (IR) via TDI.
 - The TAP controller passes through the *Update-IR* state via TMS and the IR decodes the instruction.
 - TDI is now multiplexed to the Boundary-Scan Register (BSR) for data.
 - The TAP controller is placed in the *Shift-DR* state via TMS.
 - Serial data is shifted into the BSR via TDI.
 - The TAP controller passes through the *Update-DR* state via TMS and the BSR drives the device's pins with its data.
-

The standard suggests three possible test bus implementations. In all cases, TCK is broadcast to all system components. The first implementation employs a broadcast TMS and a single daisy-chained TDI-to-TDO scan chain. This serial connection is shown in

Figure 4. It is the method the author chose to implement on the DFT system. Other implementations include a separate TMS line per scan chain with all serial chains terminating in a single TDO connection--dubbed the "Paralleled Serial Chains" configuration. This configuration utilizes the three-state nature of TDO by maintaining one active scan chain at a time. Finally, the third implementation involves a broadcast TMS line with separately connected TDI-to-TDO scan chains. This method is known as the "Multiple Independent Paths with Common TMS and TCK Signals" [IEEE 1990].

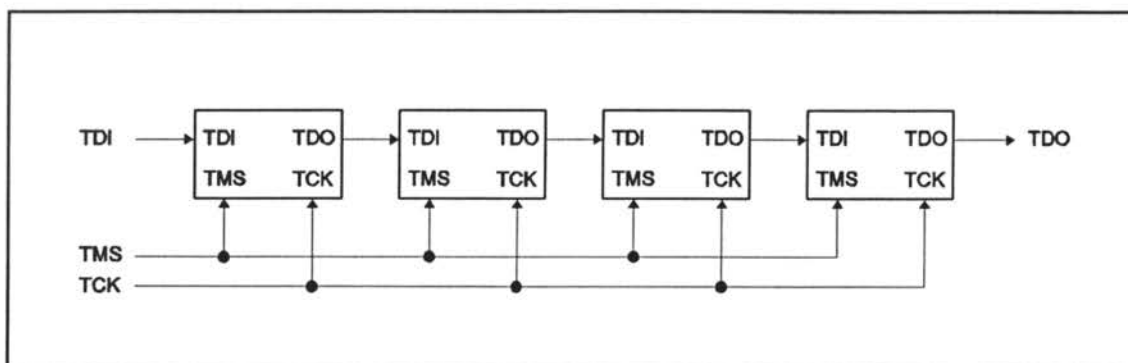


Figure 4. Serial Connection Using One TMS Signal. Adapted from [IEEE 1990].

The IEEE 1149.1 standard is an open-ended standard in that user-defined instructions and registers may be added. For example, Texas Instrument's (TI) System Controllability/Observability Partitioning Environment (SCOPE) family of 1149.1 compliant ICs implements an additional 2-bit Boundary Control Register (BCR) to control additional test functions such as Pseudo-Random Pattern Generation (PRPG) and Parallel Signature Analysis (PSA) [TI 1991]. The author used TI's family of SCOPE octal buffers, transceivers, and latches to implement boundary-scan on the DFT system. See section 4.1 entitled *Hardware and Software Components* for more information.

2.3.4 Fault Detection

The classes of faults that can be detected on the DFT system are a function of the IEEE 1149.1 standard, DFT structures, and DFT software. The DFT structures are the discussed later in the section 4.4 appropriately entitled *DFT Structures*, and the author can only comment on the test software since Lebak was responsible for this portion of the project.

The 1149.1 standard was developed primarily to catch manufacturing defects like those outlined in Figure 2. The classes of faults that cannot be detected via this test architecture includes the class of operational faults such as timing problems, hazards, races, or some internal core component chip failures. This is because the test logic operates entirely independent of the device's normal core logic, so there is no way to synchronize events between the two. Therefore, when device logic signals change, the test logic has no time reference to initiate sampling operations or to indicate when these events occurred. Future proposed 1149 standards will address this and other test issues².

² There are currently four other IEEE 1149 working groups (1149.2, 1149.3, 1149.4, and 1149.5) that plan to address other test-related issues such as at-speed testing, an analog test architecture, and a Modular Test and Maintenance multi-drop bus (1149.5 or MTMbus) that interfaces to 1149.1 scan chains.

Chapter 3 Microprocessor Systems

3.1 Hardware Goals and Constraints

Since two systems were designed utilizing the same technology and core components, the reader should assume that non-test discussions about the project refer to both microprocessor systems. In ambiguous cases, differences in the two hardware implementations will be explained. Although references are made to the DFT system, the primary purpose of this section is to highlight features common to both systems.

The primary goal of the DFT hardware design was to decrease maintenance costs while providing as much hardware support for fault isolation at the component level as possible. Both structured and Ad Hoc DFT techniques were used to place controllability and observability at the areas of the system most prone to failure.

It was important to control design costs so that the benefits of DFT could be realized. Initial capital investments would be high, but the long-term maintenance costs should more than offset any additional component or real-estate costs. There were both functional and mechanical system constraints that affected this test cost.

The traditional and DFT system designs are not meant as a 1:1 replacement for either of KSU's current Motorola-based systems--the MM17 or MM19. Rather, they are meant to be functionally compatible with these microprocessor boards, yet take advantage of new advances in technology over the past ten years to replace many obsolete components. As indicated by Figure 5, these new microprocessor boards are designed to incorporate the current system's main and auxiliary board *components* onto a single microprocessor board. Therefore, the terms "board" and "system" will be used interchangeably when referring to these two systems since a single microprocessor board *is* the complete system.

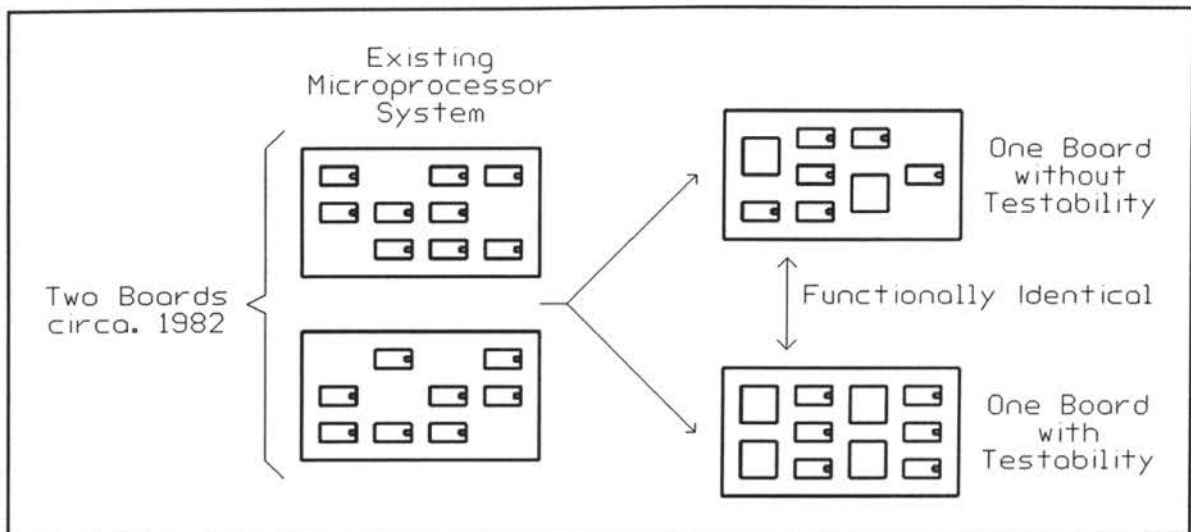


Figure 5. Traditional and DFT Microprocessor System Design Origins.

The functional differences between these three microprocessor systems are minor. In some instances user configurations were enhanced, while in other areas, specific KSU needs necessitated the construction of such items as a dedicated serial connection. A design goal of a system free of hardware jumpers meant that some unneeded flexibility (like the countless choices available for memory) was eliminated.

Choices had to be made about the number of peripherals to include in these new designs since the Motorola MM17 board incorporates three ACIAs and one PIA, while the MM19 design contains one ACIA and two PIAs. Only two ACIAs are necessary to perform the dedicated console and mass storage link connections, so only two ACIAs were required. In addition, the dedicated unidirectional printer PIA was removed, leaving a single user PIA.

3.1.1 Functional Compatibility

The hardware designs were directly affected by KSU's large investment in the current Motorola microprocessor systems. KSU's investment in hardware includes laboratory card cage systems, connectors, cables, peripherals, and a serial communications link to

a PDP-11 for permanent storage of student programs or projects. Software investment includes Motorola's SUPERmon (MC6809 monitor software) and KSU's Mother Hen software (communication software for the permanent mass storage link). KSU professors, instructors, and technicians have also invested a considerable amount of time debugging the system and developing programs.

This investment meant a realizable savings if upward compatibility with the existing Motorola MM17 system could be achieved. To make this goal a reality, the same decade-old MSI (LSI) components were used in both newly designed systems. Furthermore, KSU-specific compatibility was attained by designing the microprocessor systems around the current system memory map. Although certain implementation decisions were mandated by this compatibility, it did not affect the testability study since both systems were affected in the same manner.

Upward compatibility required more than just using the same microprocessor, peripherals, and memory map as the Motorola MM17 system--it meant placing mechanical design constraints on the width, length, and depth of the PWB so that existing card cage systems could still be used. Furthermore, edge connector signals had to be placed in the same location as their Motorola counterpart to maintain connector and cable compatibility.

3.1.2 Mechanical Constraints

Motorola utilizes a multi-layer 9.75" x 6.00" PWB for their microprocessor system. Unfortunately, a multi-layer design was not really an option since the traditional system board could most likely be fabricated in two layers. Keeping the fabrication layers for the two designs the same is crucial to making accurate statistical design comparisons. If the

DFT design required a multi-layer PWB, then a direct DFT versus traditional design cost comparison would have been biased toward the traditional design. Although the DFT design placed some constraints on the PWB artwork specifications, the additional cost of a multi-layer design was avoided.

Initially, the new microprocessor systems were to be a complete replacement for both the main and auxiliary system boards in terms of the current system's components, functions, and connectors. However, there was not enough space available on a 9.75" x 6.00" PWB for the additional connectors and signal routes that such a board would have required. Therefore, decisions had to be made about which components and/or connectors would or could be placed in the (more complex) DFT system.

After suitable block diagrams were constructed, it was determined that a fair comparison between system designs could only be made if the DFT system core and test components were placed on the main microprocessor system board. An auxiliary board would then only be used to provide access to the expansion connector signals. To determine if this was feasible, and address the mechanical space limit, a schematic capture package from Ovation, Inc. (SCHEMA II+) was used to create a 9.75" x 6.00" board outline (1/4" border was left around the edges, and ample space was left for the connectors). SCHEMA provides a visual dot grid that can be used to assist component placement. The schematic library contains IC component layouts of most DIP package sizes [Ovation 1989a]. Together the component grid (0.200" x 0.200" square dot grid) and layout ICs were used to verify that all of the core components and test structures would fit on a single PWB.

This conclusion was based on the size and orientation of the system components. The sizes of the board's core components was well known. However, this conclusion

was also based on preliminary information from Texas Instrument's (TI) on the size of their scannable test ICs, the number of test components to be used, and the ways in which non-scan test fixtures would be implemented. Furthermore, it was assumed that board components would be orientated the same to facilitate component placement.

3.2 System Designs

As indicated above, both the DFT and traditional board designs are upward compatible with their older Motorola counterpart. Both designs feature components from the Motorola 6800 family. For example, the microprocessor is an 8-bit MC6809. I/O capabilities are provided by the MC6821 or PIA which provides two 8-bit parallel ports, and the MC6850 or ACIA which provides two serial communication channels. Timed operations are controlled by the one of three programmable timers on the MC6840 or PTM [Motorola 1983].

The traditional system block diagram can be found in Figure 6. It shows a typical microprocessor design with one set of address, data, and control busses. Standard low-power schottky TTL bus drivers control the bus signals, while Programmable Array Logic (PAL) is used to perform chip select decoding and control these bus driver octals.

3.2.1 Core Components

This section discusses in detail each of the components common to both system designs. Since these components are common they will be referred to as core components. Much of the material presented here was extracted directly from the *Motorola 8-Bit Microprocessor & Peripheral Data Book* [Motorola 1983] or *Programming Microprocessor Interfaces for Control and Instrumentation* [Andrews 1982]. Readers

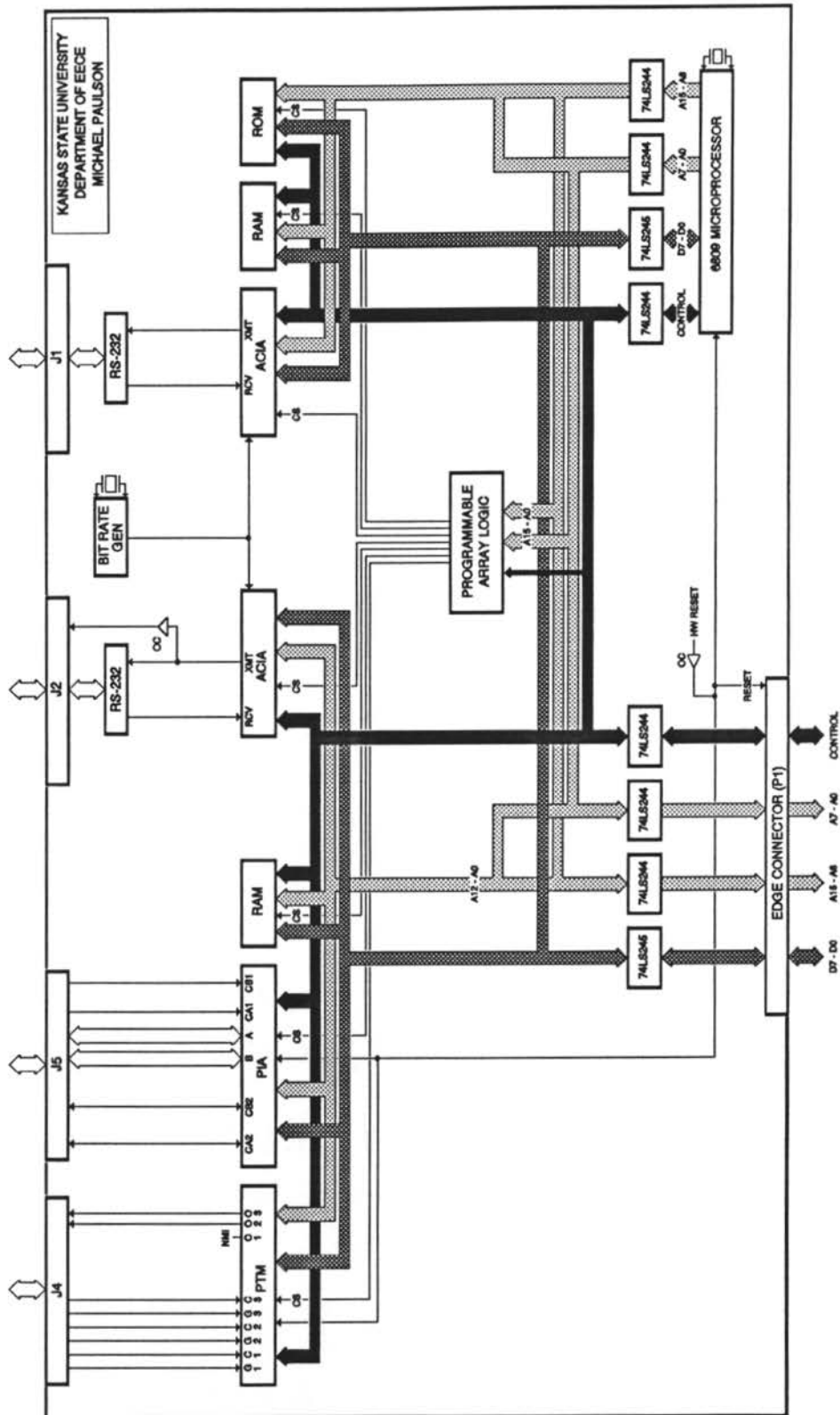


Figure 6. Traditional Microprocessor System Block Diagram.

wishing to find more detailed information should consult one of these references.

3.2.1.1 Microprocessor

The Motorola MC6809 is a byte-oriented microprocessor (8-bit data bus) capable of 16-bit addressing and some 16-bit arithmetic and ALU functions. Two separate 8-bit accumulators (A and B) can be combined to perform these 16-bit arithmetic and logic functions. The microprocessor's five 16-bit registers include X and Y index registers, user and system stack pointers, and a program counter. 8-bit registers include a direct page register and a condition code register. Two bi-phase clock signals (E and Q) are sourced by the microprocessor to simplify peripheral interface timing.

The instruction set includes standard arithmetic functions like add (ADD), and subtract (SUB), and unusual functions like decimal adjust (DAA), which is used to support BCD addition. A few of the most notable addressing modes include inherent, immediate, direct, indexed, and extended. Inherent instructions have no arguments, the necessary information to execute the opcode is inherent in the instruction, while immediate instruction's data immediately follows the instruction. The effective address of a direct instruction is found by the concatenation of two 8-bit values. The upper address is taken from the direct page register, and the lower address is taken from the data byte following this opcode--this provides direct access to 256 addresses within 256 pages. In extended addressing, the effective address of the argument immediately follows the opcode--allowing references to any one of 64K possible memory locations.

3.2.1.2 System Components

3.2.1.2.1 Memory

Both microprocessor systems incorporate byte-oriented single-voltage supply Static Random Access Memory or SRAM (volatile memory) and Erasable Programmable Read Only Memory or EPROM (nonvolatile memory). Unlike Dynamic Random Access Memory (DRAM) found in most desk-top computers, SRAM requires no refresh cycle, thus reducing system complexity.

The DFT system supports an 8K x 8 or 32K x 8 user SRAM and EPROM under LCA-firmware control--although current plans are to use only one EPROM. The traditional system supports only one 8K x 8 user SRAM. The additional DFT system EPROM space was necessary to "house" the on-board testability software. Due to the 64 Kbyte addressing limitation of the microprocessor and the memory map constraints, all of this additional memory was not accessible to the microprocessor at any one time. Microprocessor access to the memory was performed by both EPROM and LCA firmware-controlled bank selection. The EPROM housed the bank select routines, and the LCA controlled the bank select operations. The upper three address lines were used to select one of seven 4 Kbyte bank selected memory blocks--the 4K block that housed the memory bank select routines was in memory at all times.

Only 28-pin devices conforming to the JEDEC-approved 28-pin EPROM and SRAM pinouts can be used interchangeably. That standard defines the placement of pins on both volatile and nonvolatile memories. A few of the important pinout differences/similarities defined by this standard are shown in Table 4. Identical address and data pin placement for these devices is not shown.

An initial design objective was to make the EPROM and RAM memories interchangeable, i.e. either socket could be used. However, the location of certain ROM and RAM-specific signals (A14, Vpp, and WE*) effectively prevented this. Vpp is

Table 4. RAM/ROM Pinout Differences/Similarities.

Pin #	EPROM		SRAM	
	2764 (8K x 8)	27256 (32K x 8)	6264 (8K x 8)	62256 (32K x 8)
27	PGM*	A14	WE*	WE*
26	NC	A13	CS2	A13
22	OE*	OE*	OE*	OE*
20	CE*	CE*	CS1*	CS*
1	Vpp	Vpp	NC	A14

hardwired high via a PU resistor on pin 1 of the EPROM. Although this same pin could have been hardwired high as well on the RAM sockets, WE* is sourced by the R/W* signal for the RAM, but it is an address line on the EPROM (A14). This difference in signal sources made futile memory interchanges without the use of jumpers.

To reduce system complexity, the OE* and CE* on the EPROMs are sourced by the same chip select signal. As indicated by Figure 26 on page 186 in the Timing Analysis section, write operations are not affected by the assertion of the OE* on the RAM. However, the OE* is generally hardwired to a R*/W signal, via an inversion of the MC6809's R/W* line, but this would have required another LCA signal or inverter package. This complication was avoided by hardwiring the signal to CE* which was sourced by the LCA.

Although some pin-incompatibility exists between the 6264 and 62256 SRAMs, (e.g. pin 26 on the 62256 is A13, and it is CS2 on the 6264). It is still possible to use either of these memory devices if care is taken to assert CS2 when accesses to the 6264 are initiated.

3.2.1.2.2 Parallel Port

Byte-wise parallel I/O capabilities are provided by Motorola's MC6821 or Peripheral Interface Adapter (PIA). It features two separate 8-bit parallel I/O ports (ports A and B) with four dedicated handshake lines. Lines CA1, and CB1 are unidirectional input control signals for ports A and B respectively. Similarly, CA2 and CB2 are bidirectional control lines. Six on-chip registers maintain control of the bi-directional communication ports (three per port). These 8-bit registers are selected via two register select inputs--two register pairs reside at the same address. Accesses between these registers is assisted by a register select bit within each port's control register.

The two Control Registers (CRA and CRB) also control the operation of the handshake lines and the two port-specific interrupt request lines (IRQA* and IRQB*). The two Data Direction Registers (DDRA and DDRB) are used to configure each corresponding bit in the I/O port as an input or output (a "0" or "1" respectively). Finally, the two data registers (DRA and DRB) are accessed when data is read or written to either of these ports [Motorola 1983].

The PIA port signals were not buffered off the edge connector. Although signal buffering provides better drive capability and a desirable level of peripheral ↔ student isolation, the buffering disadvantages (e.g. changing a bidirectional PIA pin to a fixed unidirectional one) far outweighed the advantages. On the DFT design, the scannable bus driver octals cost almost four times as much as the peripherals themselves [Asay, Gordon, and Lenhert 1991]. Therefore, from a component-cost standpoint, DFT system

bus driver failures are more costly than peripheral failures³. To maintain compatibility between microprocessor systems, no PIA buffering was performed on either design.

Two of the three PIA chip select lines were hardwired active via pull-up resistors. The remaining active-low chip select is sourced by a PAL in the traditional design, and by the LCA in the DFT design. Thus, accesses to the PIA are made when its single active-low chip select is asserted by the on-board programmable logic. Furthermore, the individual port interrupt request lines (IRQA* and IRQB*) were combined and connected to the microprocessor's IRQ* so that interrupt service requests could be made via either of these signals.

3.2.1.2.3 Serial Ports

Two MC6850s or Asynchronous Communications Interface Adapters (ACIAs) permit serial communication via unidirectional transmit and receive channels on both the traditional and DFT systems. These devices feature modem or terminal control, data serialization, and clock division. One ACIA is a dedicated system terminal, while the other one provides a serial communication link to a PDP-11 for permanent storage of user programs or data.

Two read-only and write-only registers control the transmitted and received data. These four 8-bit registers share two common address locations within the ACIA. Individual registers are selected during the microprocessor read and write cycles by a combination of the R/W* and register select line inputs. The Status and Control Register

³ Diagnosis costs may negate any monetary savings obtained by not including these scannable buffers.

(CR) share a common memory location, while the Transmit and Receive Data Registers (TDR and RDR respectively) share the other location.

The CR controls ACIA communication protocol options like parity, the number of data and stop bits, interrupt generation, and the selection of a clock divisor. Status register information indicates if a received parity error or data overflow has occurred. It can also be used to determine if the TDR is full (TDRF) or if the RDR is empty (TDRE) so additional data can be sent or read [Andrews 1982].

A 1.8432 MHz crystal and MC14411 Bit Rate Generator (BRG) create a 50% nominal duty-cycle clock that the ACIAs use for the synchronization of serial data transfers--it drives the transmit and receive clock inputs (TxCK and RxCK). A "x16" BRG output clock rate was selected via $RS_A = 0$, and $RS_B = 1$. This provides 16 different output frequencies available on pins F1 through F16. On the DFT system, a BRG output frequency of 153.6 kHz was selected (F1) and was sent to the LCA. This permits firmware-controlled baud rates from 75 to 9600 Baud via a counter or register within the LCA and a "x16" or "x64" ACIA frequency configuration. The BRG F8 output was hardwired to the traditional design's ACIA clock inputs. As such, changes in this frequency must be made by the ACIA's internal frequency divider circuit.

As shown in Figure 7, each traditional board ACIA channel is buffered by the MC145406; a TTL to RS-232 driver/receiver. The MC145406 requires both a positive and negative 12 volt power supply to generate the necessary 24 volt swing RS-232 compatible signals. This configuration does not permit one to determine the values on these lines without physically probing them.

Like the PIA, two of the three ACIA chip select lines were hardwired active via pull-up resistors. The remaining active-low chip select is sourced by a PAL in the traditional

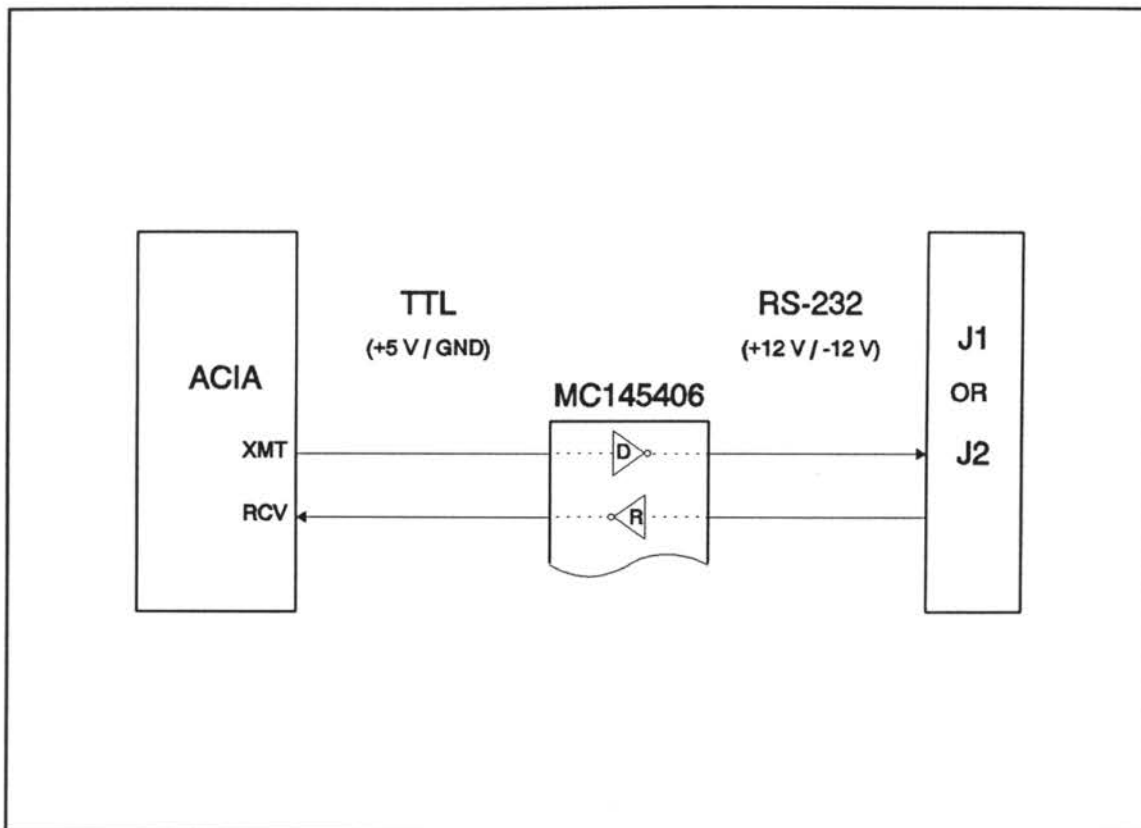


Figure 7. Traditional ACIA/RS-232 Interface Connections.

design, and by the LCA in the DFT design. Thus, accesses to the ACIA are made when its single active-low chip select is asserted by the on-board programmable logic. Furthermore, ACIA operations were simplified by hardwiring some of the ACIA's operational signals. This includes grounding both the Data Carrier Detect (DCD*) and Clear-to-Send (CTS*) lines so that the transmit and receive sections of the ACIA are wired active. Since each ACIA is dedicated to perform a different function, the connector-ACIA signals are distinct. The system terminal ACIA's transmit and receive lines are routed to J1-3 and J1-5 respectively. The Mother Hen or modem ACIA connections are wired just the opposite; J1-3 is the transmit signal and J1-5 is the receive signal, i.e. they are wired to mate to a terminal. Furthermore, an open collector transmit line is connected to J2-19. All of these board outputs are then connected together via a pull-up resistor at a PDP-11

interface. This line is a multi-drop quasi-RS-232 connection to the PDP-11. When service is initiated, files can be stored or retrieved via J2.

As indicated in Figure 8, flexibility in serial communication connections is provided by J6. It supports three solderable power connections to permit pin-strapped power supply connections to +5, or ± 12 volts. Connections of this sort may be necessary for RS-443-type connections.

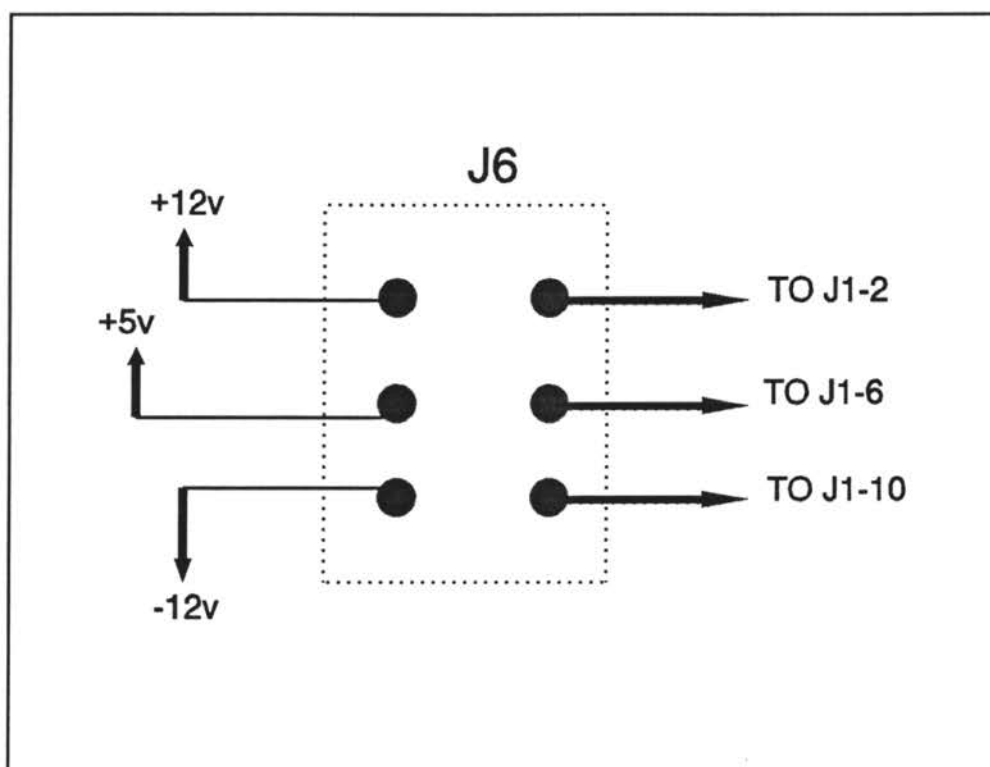


Figure 8. Strappable Serial Power Connections.

3.2.1.2.4 Timer IC

The MC6840 or Programmable Timer Module (PTM) permits programmable timer functions on each system. It can generate timing signals, delays, pulse-trains, or system interrupts, and measure frequencies or pulse widths. It features three separately programmable 16-bit counters or timers, three 8-bit control registers, and an 8-bit status register. All of these registers are selected via the three register select inputs, although

two control registers share a common address. The LSB of the second control register is used to arbitrate between control registers one and three.

The counters can be pre-loaded with a value during counter initialization to alter the number of clock cycles during which the counter will count. The status register contains three separate interrupt flags (one per timer), and one composite interrupt flag that is asserted when any of the three timer interrupt flags are set. Three separate active-low gate inputs or triggers (G*) can be used to selectively enable each timer, while the three clock inputs (C*) can be used to decrement their respective timer. Each timer also features a single active-high output (O) that can drive two TTL-compatible loads while producing an output waveform. Both single-shot and continuous timer modes are available at this output. SUPERmon uses the PTM to generate a Non-Maskable Interrupt (NMI) on O1 to support single-stepping processor trace functions for user program debug purposes. Therefore, that signal was connected to the microprocessor's NMI* signal.

Finally, one PTM chip select line was hardwired active via pull-up resistor. The remaining active-low chip select is sourced by a PAL in the traditional design, and by the LCA in the DFT design. Thus, PTM accesses are made when its single active-low chip select is asserted by this programmable logic.

3.2.2 Programmable Logic

The use of the Logic Cell Array or LCA will be covered in detail in Chapter 4 since it is somewhat complex. However, the programmable logic (two PAL22V10s) used in the traditional design performs relatively simple functions compared to its DFT system counterpart. Unlike the LCA, there were no test functions to support and no reset or software configuration problems to address. Nonetheless, complete address decoding

presented problems in both designs. For the DFT system, the LCA pins were at a premium and the two low-order address pins were removed--so full address decoding was not possible. For the traditional design, full address decoding was made possible by a combination of both IC's logic.

PAL U10 multiplexes the upper address byte from 8 → 3 lines and sends these three lines of information to the lower-address PAL U15. U15 combines this information with the lower address byte and signals BE and BR/W so it can fully decode all the addresses and source the traditional system chip selects.

Due to the size of the two devices, only six buffer control lines control the 16 buffer and transceiver enable and direction pins (a total of eight octal buffers and transceivers). These buffer control signals and EBUSGNT are sourced by PAL U10. Finally, unlike the LCA, five PAL signals were unused.

3.2.3 Memory Map

The different memory maps used by each of the three microprocessor systems are shown in Table 5. A single column was used to indicate both the traditional and current microprocessor system memory maps to emphasize the fact that these systems are almost identical. All three memory maps are virtually identical--this promotes software portability (either user or EPROM-based routines) across all systems. However, some components are not available in all three systems. For example, the DFT design features an on-board Test Bus Controller (TBC) that is not available on either of the other two designs. Furthermore, the Motorola design features two PIAs and three ACIAs--one more than either the DFT or traditional designs include.

Table 5. System Memory Maps. Adapted from [Lenhert and Gordon 1990].

Address (Hex)	Current / Traditional System Functions	Design-For-Testability System Function
0000-1FFF	User RAM	Secondary RAM
2000-CFFF	Unused	Unused
D000-D3FF	Mother Hen Program	Mother Hen Program
D400-D4BF	Memory Test Program	Memory Test Program
D4C0-D4DF	Memory Move Program	Memory Move Program
D4E0-D4FF	Memory Clear Program	Memory Clear Program
D500-D5FF	ACIA Test Program	ACIA Test Program
D600-D7FF	PIA Test Program	PIA Test Program
D800-DFFF	SUPERmon ROM	SUPERmon ROM
E000-E7FF	System RAM	Primary RAM
E800-EBFF	User I/O Area	User I/O Area
EC00-EC0F	Unused	Unused
EC10-EC13	System PIA* / Not Available*	Not Available*
EC14-EC15	System ACIA (Console)	Primary ACIA (Console)
EC16-EC17	Unused	Unused
EC18-EC1F	System PTM	Secondary PTM
EC20-EC53	Unused	Unused
EC54-EC55	System Aux. ACIA* / Not Available*	Not Available*
EC56-EC7F	Unused	Unused
EC80-ECA7	Not Available*	Primary TBC*
ECA8-ECFF	Unused	Unused
ED00-ED1F	Not Available*	Used by LCA*
ED20-EDBF	Unused	Unused
EDC0-EDC3	User PIA	Secondary PIA
EDC4-EEBF	Unused	Unused
EEC0-EEC1	Mother Hen (MH) ACIA	Secondary ACIA (MH)
EEC2-EEFF	Unused	Unused
EF00-EFFF	User I/O Area	User I/O Area
F000-FFFF	SUPERmon ROM	SUPERmon ROM

* denotes system differences.

Since these microprocessor systems are used primarily for educational purposes, much of the address space is left unused for future peripherals. For example, a 44K block of addresses from \$2000 to \$CFFF is available for student use. Students can map additional peripherals into this address space. Furthermore, some of the EPROM areas shown as unused in the DFT column are actually used by the test software during the power-on self-tests. In particular, the EPROM areas from \$D000 to \$DFFF and \$F000 to \$FFFF are often used via bank selection by the test software routines [Lebak 1991].

3.2.4 Timing Analysis

Like all complex digital interfaces or systems, the DFT and traditional designs required timing to verify proper function. To preserve a degree of decency, and to protect sensitive readers, the timing equations and diagrams are located in Appendix G: *Timing Diagrams & Equations for Design Verification*. Only the results of the timing analysis are considered here.

To satisfy some of the DFT interface equations, certain logic delays were assumed. In particular, the LCA propagation delay is a function of the amount of combinational logic used to implement its functions. However, "worst-case" simulation results for a 50 MHz LCA indicated a propagation delay of 35.8 ns [Gordon 1991b]. The minimum LCA propagation delay was assumed to be zero to satisfy some of these system equations.

Although the peripherals and microprocessor are from the same family, the peripheral chip select setup time was violated in both the read and write operations. For peripheral read and write operations, it was violated by 15 ns. In addition, in the traditional system, the microprocessor's read data setup time is violated by about 10 ns—it is marginal for the DFT system.

Since worst-case analysis was performed for both systems, it was interesting to note that *all* of the worst-case parameters were taken from the traditional design. A single worst-case 74LS245 delay is greater than three worst-case 74BCT8245 delays! These analyses also used the fastest available processor and peripherals, so the interface problems suggested by the timing analysis may never present a problem.

3.2.5 Edge Connectors

The DFT system incorporates five interface connectors and one expansion edge connector that connects to a card cage backplane. Information in Table 6 shows the edge connectors and the system functions they serve. The traditional board design does not implement J3 (the test connector). All of the interface connector signals and pinouts are shown in Appendix B: *EECE 631 Hardware Reference Manual*. Interested readers should consult Table 21, Table 22, Table 23, Table 24, and Table 25 for these J1 through J5 interface connectors signals.

The connectors are arranged from right to left in a J1 through J5 order (from a user's perspective as seen in the card cage). The reference designator for these connectors are arranged so that one can read these names from the front of the card cage system. Pin 1 is the right-most signal pin on the component side of each edge connector group. The J6 solder jumper shown in Figure 8 is located next to J1. This *permanent* jumper provides +12, -12, and +5 voltage connections to the primary ACIA in the event that they may be required. A solder connection between two adjacent pads will connect these signals together.

The choice of a connector pinout on the new systems was determined primarily by existing connectors and cables. The J1, J2, and J5 connectors had to be pin-compatible

Table 6. DFT System Connector Descriptions.

Reference Designator	# Pins	Description
J1	20	Primary Serial Interface (Terminal ACIA)
J2	20	Secondary Serial Interface (Mother Hen ACIA)
J3	14	Test Interface
J4	20	Programmable Timer Interface (PTM)
J5	26	Parallel Interface (PIA)
P1	86	Expansion Edge Connector

with their Motorola counterparts so existing cables could be used. However, the J3 (test connector) and J4 (PTM) connector pins were arranged in a fashion that minimized signal interconnect lengths on the PWB. This was possible since no cables existed for either of these connectors.

Initially the test connector was to be an 8-pin male jumper located near the board edge. Later after determining that sufficient connector space was available, the connector was changed to a 14-pin edge connector. To maintain consistency among connectors, the signal pins became the odd pins, and the even pins were grounded. One additional signal (pin 2; the PODHERE* signal) is not grounded on the board, although it is grounded within the cable. (See page 75 for more information on the test bus connections.) Although enough space was available for the test connector, sufficient edge connector space (on the user interface side) was not available for the expansion edge connector signals. Therefore, an auxiliary board will still have to be used to provide these signals from the card cage backplane.

To avoid an inadvertent connection between dissimilar cables and connectors, all interface connectors were polarized differently. Edge connector information in Table 7

shows the placement of connector keys and the size and spacing between the pins. The general polarization trend was to increase the length between the key and the one connector edge so that it would be immediately obvious which cable went with which connector.

Table 7. System Connector Specifications.

Reference Designator	# Pins	Key Location	Pin Width	Ctr-to-Ctr Spacing
J1	20	pin 3/pin 5	0.050"	0.100"
J2	20	pin 5/pin 7	0.050"	0.100"
J3	14	pin 3/pin 5	0.050"	0.100"
J4	20	pin 7/pin 9	0.050"	0.100"
J5	26	pin 7/pin 9	0.050"	0.100"
P1	86	pin 19/pin 21	0.100"	0.156"

Although functional compatibility with the Motorola design was maintained, several expansion edge connector differences exist between these designs. For example, the traditional and DFT designs do not directly support the addition of the slower 6800 peripherals, i.e. ones that require a Valid Memory Address (VMA) or Valid User Address (VUA) signal. These signals are not available on either system since they were seldom if ever used by students. However, one can still use the BA and BS signals to generate VMA ($VMA = BS + BA$).

A unique expansion edge connector pin labelling scheme exists on the Motorola design. Signal names with an asterisk denote active-low signals, while the expansion connector names use a different notation--an apostrophe in the case of the new designs. Motorola uses the numbers from 1 - 42 on the non-component side of the board (even

signals), and alphabetic characters on the component (or odd) side of the board. Since the number of component signals (42) is greater than the number of available letters (26), Motorola also uses inverse alphabetic names and skips nine letters: G, I, O, Q, G', I', O', Q', and Z'. The author originally followed this convention, but problems with the PWB software package (SCHEMA-PCB) necessitated a more standard odd/even pin numbering method. However, cross-references between the two numbering schemes is shown on both schematics (in parenthesis next to the standard pin numbering scheme), and the expansion edge connector table found in Appendix B: EECE 631 Hardware Reference Manual.

In addition, the signals shown in Table 8 are equivalent signals in both the traditional and DFT systems. Since Dynamic Random Access Memory (DRAM) will seldom if ever be used in these microprocessor systems, both the DRAM refresh request and refresh acknowledge signals are coupled with the bus request and bus acknowledge signals. Furthermore, three of the four edge connector clock signals are sourced by the microprocessor's E clock (the other clock is Q).

Table 8. Equivalent Expansion Edge Connector Signals.

EREFREQ = EDMA/EBUSREQ*, i.e. P1-24 (12) = P1-23 (N)
EREFGNT = EBUSGNT, i.e. P1-26 (13) = P1-30 (26)
EMEMCLK = ECLK = EE, i.e. P1-19 (L) = P1-36 (18) = P1-15 (J)

The other edge connectors signals are the same across all three systems. For example, both newly designed systems provide access to a full 16-bits of buffered address lines and 8-bits of buffered data lines. These signals can be reached through the auxiliary board via the card cage backplane.

Chapter 4 Testability

Although the major components on the DFT design are common to both newly designed microprocessor systems, many testability differences exist. This section attempts to illustrate these design-for test features. The block diagram in Figure 9 shows how the DFT system differs from a traditional design. It incorporates partitioned busses, the IEEE 1149.1 test bus, and several DFT structures. A Logic Cell Array (LCA) is used to perform more than just the chip select decoding and bus driver control found in the traditional design—it is used to provide support for on-board test functions.

4.1 Hardware and Software Components

BIST components include Texas Instrument's (TI) IEEE 1149.1 conformable Scan for Controllability Observability Partitioning Environment (SCOPE) octals (slightly shaded in Figure 9), TI's boundary-scan Test Bus Controller (TBC part number SN74ACT8990), and an external boundary-scan test access port. This test port provided an external connection to an IEEE 1149.1 test bus via a dedicated edge connector. This connector, along with a software development system from TI known as the Advanced System Support For Emulation and Test (ASSET) may be used to control test operations via the scannable buffers on the board. ASSET provides the necessary IEEE 1149.1 vectors to control these scannable devices for off-board testing while the TBC provides on-board test bus control.

4.1.1 SCOPE Octals

TI's SCOPE family of ICs feature both BiCMOS (BCT) and Advanced CMOS Logic (ACL) technology. The octal buffers, transceivers, and latches are fabricated in BiCMOS.

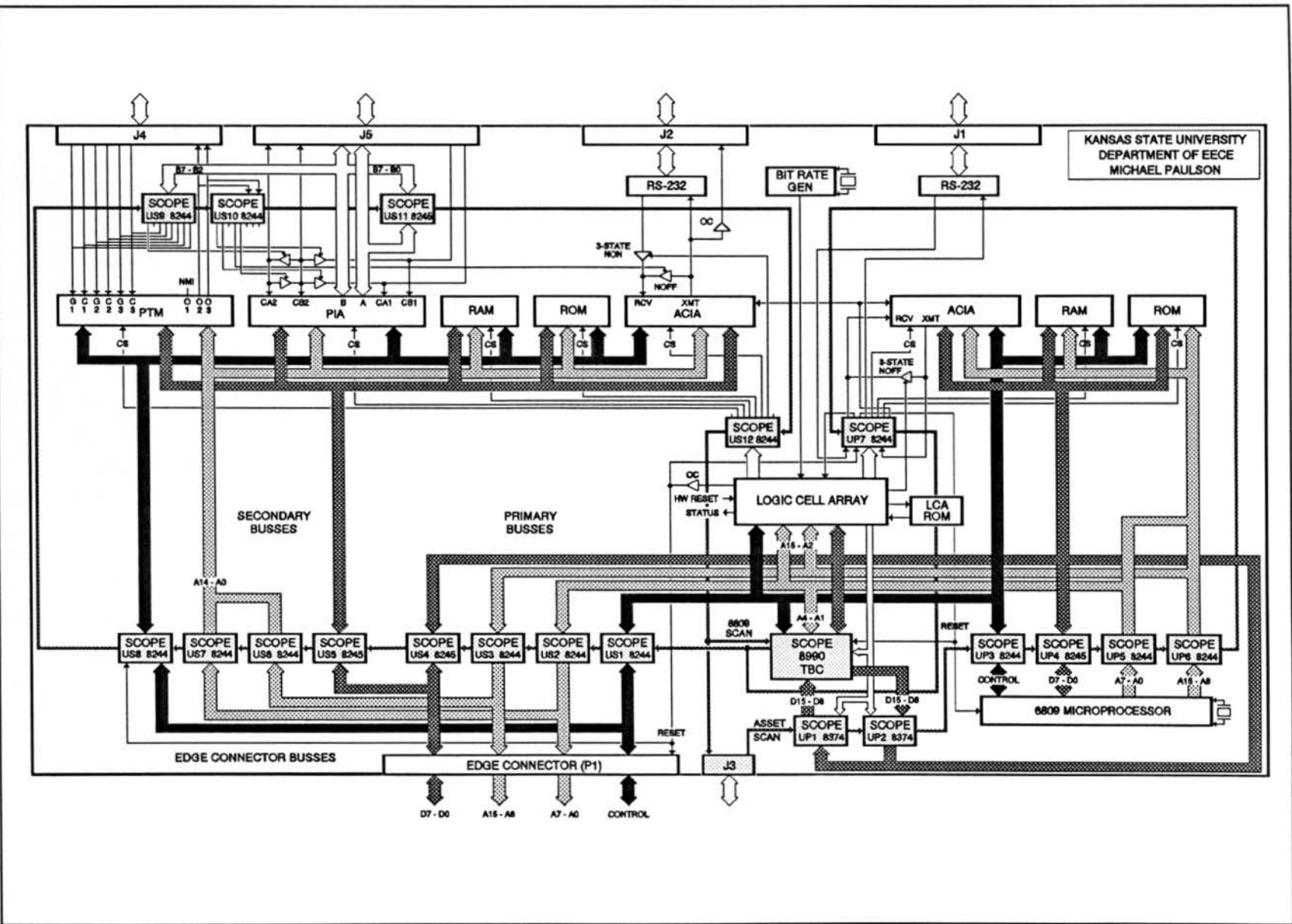


Figure 9. DFT Microprocessor System Block Diagram.

This technology provides up to 64 mA of drive current--ideal for bus driving applications. The TBC features ACL technology for highly complex circuitry in a low-power package [TI 1991].

The ICs implement an 8-bit IR, an additional data register called the 2-bit Boundary Control Register (BCR), and ten additional test instructions⁴. The diagram in Figure 9 shows how the DFT system incorporates thirteen 74BCT8244 octal buffers, four 74BCT8245 octal transceivers, and two 74BCT8374 octal D edge-triggered flip-flops [TI 1991]. Each of these component's functions are identical to their non-test counterparts: the 74244, 74245, and 74374 respectively.

The 74BCT8245 transceivers are used for the data bus and the PIA's electrical wraparound. The 74BCT8374 octals are required for the 16-bit to 8-bit data bus interface between the microprocessor and the TBC. 74BCT8244 functions included controlling electrical wraparound signals, buffering the control and address busses, and buffering other signals such as the peripheral chip selects. In all 19 instances, the octals provided a level of visibility and control that would not have been possible with their traditional board counterparts.

Although the 74BCT8244 and 74BCT8245 are functionally different, their pinouts are virtually identical--the only difference being the control pin functions. The 74BCT8244 contains two 4-bit output enables, while the 74BCT8245 contains an output enable and a direction control pin. Since the Logic Cell Array is the source of all the octal control

⁴ Two instructions that TI implemented as user-defined instructions may be considered by the IEEE 1149.1 working group as recommended instructions. The first instruction places the BSR output cells in a high-impedance state while selecting the bypass register in the scan chain--called HIGHZ by the IEEE working group. The second instruction also selects the bypass register path to TDO, but applies the BSC data to the device's functional inputs and outputs--called CLAMP [IEEE 1991].

lines, devices could be interchanged if one wanted to reconfigure the LCA control software. In most cases this would not be beneficial. However, if off-board Direct Memory Access (DMA) to the primary area is to be allowed, one could replace address buffers US2 and US3 with transceivers.

DMA to the primary area can be achieved despite the fact that the SADDR control signal is hardwired to all four edge connector address buffers (US2, US3, US6, and US7). An assertion of this active-low control line would enable the newly replaced US2 and US3 address transceiver outputs and change the device's direction from B to A, i.e. from the edge connector to the primary area.

4.1.2 TBC

4.1.2.1 TBC Description and Use

TI's TBC is designed to control the IEEE 1149.1 test bus implemented by 1149.1 compliant components. It was incorporated into the DFT design to enable the host processor (the MC6809) to control on-board boundary-scan test functions. It implements all the necessary 1149.1 signals as specified by the standard (TCK, TMS, TDI, and TDO), sources 1149.1 instruction and data vectors, and moves target 1149.1 devices through the TAP controller states. TI chose not to source the optional TRST* signal to 1149.1 target devices--instead TRST* is implemented as an input to reset the TBC itself. The TBC controls up to six independent IEEE 1149.1 serial scan chains each having separate TMS lines. These chains share a single TDO and up to two TDI lines. (Parallel scan chains may terminate in a single TDO signal provided only one of the scan chains is active at a time--the other signals must be three-stated to avoid signal contentions.)

Since on-board test functions use only one TBC scan chain, the TBC is over-designed for the small DFT system--only 1/6 of its channels are utilized on this design. Other semiconductor manufacturers have now introduced their versions of the TBC. For example, AT&T manufactures a Boundary-Scan Module (BSM) that controls just two scan chains--i.e. it is not over-designed for this project. However, product information and preliminary samples of this device were not available when this system was designed.

TBC scan and execution functions are controlled via one of three instructions: STATE, EXECUTE, and SCAN. Transitions from one TAP controller state to another are made by way of the STATE command. The SCAN command is used to serially shift data and instructions among the target 1149.1 compliant devices, and the EXECUTE command is used to decode and execute these serial instructions. Furthermore, the TBC can be paused, stopped, or restarted by a host test processor.

The host interface "features" a 16-bit data bus, and five address lines to select one of the 24 16-bit read/write registers ($2^5 = 32$ address locations, so six are unused) to initiate scan operations. Separate read, write, and interrupt lines (RD*, WR*, and INT* respectively) are available to control communication with a host microprocessor. TI also incorporates a TOFF* signal that effectively three-states all of the device's output pins. The author utilized this pin to avoid test bus contention problems between the on and off-board test busses.

4.1.2.2 TBC Design Problems

As alluded to above, there were a few instances where TI's implementation of the TBC caused problems with the design of the DFT system. For example, the host processor interface was designed specifically for a 16-bit microprocessor with separate read/write

strokes. This of course, caused an interface problem between the TBC and the MC6809-- which is an 8-bit microprocessor with a single read/write signal, and a common data strobe.

TI chose not to make the TBC itself scannable (i.e. a target 1149.1 compliant device) so additional on-board test components were necessary to provide some assurance of a properly working TBC. In the long run this was not a problem, since the additional components that facilitated communication between these devices were scannable. Two 74BCT8374's permitted communication between these devices while also providing a degree of testability. Section 4.4.1.1 entitled *Host Microprocessor Test Bus Control* is dedicated to describing this TBC/MC6809 interface.

One implementation problem the author ran across with regard to the TBC concerns the way the Vcc and ground pins are arranged on the PLCC package. As illustrated in Figure 10, TI placed three of the power pins on the inner pin rows making single 50 mil wide power connections to these pins impossible on any through-hole PWB. Apparently, this device was meant to be placed on a multi-layer surface-mount PWB and interfaced to a 16-bit processor with separate read and write strokes--limitations that introduced interface problems for this project.

To source an adequate amount of current to the TBC, PWB traces were routed *around* signal pins and combined back together to form the 50 mil wide power and ground traces common to the rest of the board. This work-around should avert power problems that might have surfaced had the author not addressed this problem. One might note however, that the manufacturer's choice of pin placement seems rather careless. Xilinx's PLCC implementation did not result in any such routing problems--all of its power pins were placed on the outer row.

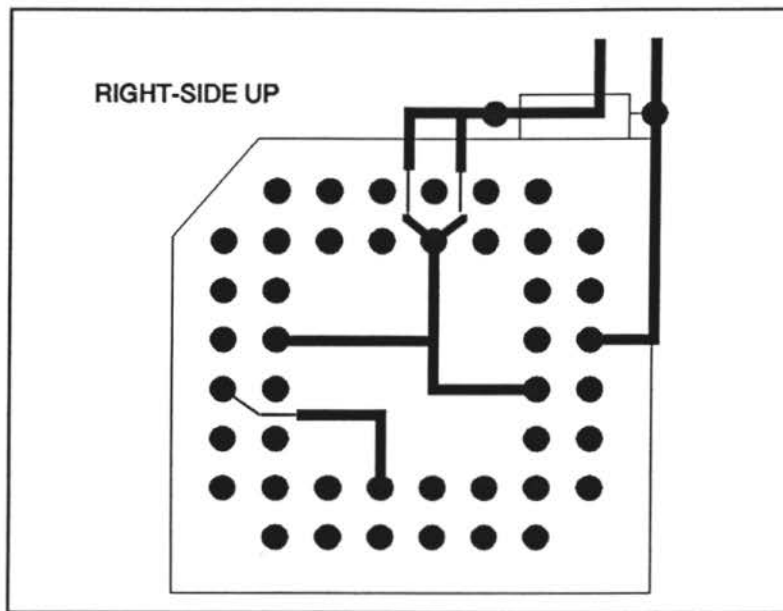


Figure 10. TBC's Vcc and Ground PWB Traces.

4.2 Board Partitioning

To facilitate testability, the DFT microprocessor board was partitioned into two sections: a primary area (sometimes called the system or hardcore area) where self-testability could not be reliably achieved, and a secondary area on which microprocessor-controlled self-test could be performed [Gordon 1991a]. The partitioning of the board was performed via TI's SCOPE octals that are connected in a boundary-scan path that permits component isolation and testing. In addition, they allow controllability and observability of the RAM, ROM, and peripherals.

4.2.1 Primary/Secondary Areas

The primary area includes the following: the microprocessor; a primary RAM and ROM; a Logic Cell Array (LCA) which provides chip select circuitry, facilitates host microprocessor testing control, and controls boundary-scan device buffer enables; a boundary-scan Test Bus Controller (TBC); and an Asynchronous Communications

Adapter (ACIA) that permits serial I/O capabilities as well as off-board observability. (This ACIA informs the operator as to the results of the on-board tests.)

The buffer configuration for the primary area allows complete isolation of both the microprocessor and edge connector by electrically isolating the primary area from the secondary area while power-on self-tests are performed. This keeps failures in the secondary area from interfering with the primary area. Buffers reaching out from the primary area may be selectively enabled once their signals have been tested.

The partitioned sections of the board, and the three separate bus areas: primary, edge, and secondary are illustrated in Figure 9 on page 41. To facilitate readability, buffered signals that traverse more than one of these bus areas are prefixed by a 'P', 'E' or 'S' respectively.

4.2.2 System Busses

Busses from the primary to the secondary area are double buffered. For the address and data busses, signals in one area reach the other board area by passing through these buffer gateways. Since the SCOPE ICs provide buffer and transceiver capabilities in an octal package, it was convenient to use signal groupings of eight. This worked well for both the address and data lines which are 16 and 8-bits wide respectively. However, there are 12 potential observability/controllability points on the microprocessor's control bus.

The problem then was determining which eight signals were most important. The interrupt request lines (IRQ*, FIRQ*, and NMI*) are open collector wire-ORed signals--not conducive to buffering. (Buffering these signals would disable their wired-OR capabilities.) Of the remaining nine control lines, RESET* was made both controllable and observable

by the LCA and its primary SCOPE octal UP7. The remaining eight control lines that are buffered by UP3 and US1 are: MRDY, HALT*, DMA*, R/W, BA, BS, E, and Q. These signals have 'P' and 'E' counterparts after buffering by UP3 and US1 respectively.

These eight original control bus signals do not traverse the entire board--the grouping of control signals changes at the interface between the primary and secondary areas. US8 sources only three control signals to the secondary area: SRST*, SR/W, SE. These are the only control signals required by the secondary peripherals.

Although the control bus in Figure 9 appears to be bidirectional, the signals are singly unidirectional. Five signals are sourced by the microprocessor and three signals are received--the diagram is meant to display unidirectionality in both directions. This required wiring the control signals in a manner that supported this two-way unidirectionality.

The 16-bit data bus interface of the TBC required additional board components (UP1 and UP2) to provide the 8-to-16 bit data interface. (See section 4.4.1.1 for more information on the microprocessor/TBC interface.) These data bus flip-flops (UP1 and UP2) were placed in the primary data bus because they could not be tested by the TBC. Flip-flop test operations are performed via boundary-scan, which of course requires the operation of these devices to communicate with the on-board TBC.

Bus control is achieved by controlling buffer enable and transceiver direction pins. All output enables are firmware controlled via the LCA. Since the LCA is pin-limited, a degree of signal sharing was deemed necessary. As shown in Figure 11, the octal control lines (a square indicates an octal buffer or transceiver) are shared between two or more ICs. For example, the four edge connector address octals are controlled by the same output enable line--SADDR. The flip-flop control lines are a special case since

those lines are shared with the Test Bus Controller. There are 19 buffer and transceiver control lines that originate from the LCA.

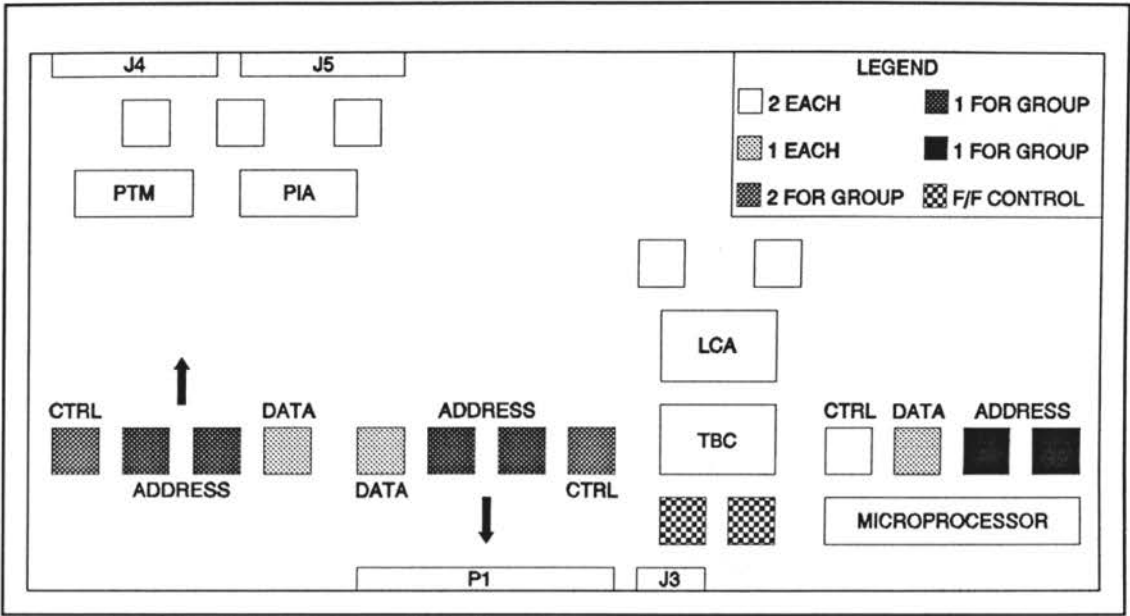


Figure 11. SCOPE Octal Control Signal Groupings.

The data transceivers' direction pins are controlled by the buffered microprocessor read/write line--i.e. PR/W. Transceiver US4 and US5 are wired in a secondary peripheral to microprocessor direction for read operations (PR/W = DIR = 1) and a microprocessor to peripheral direction for write operations (PR/W = DIR = 0). However, each transceiver's output enable line is individually controlled by the LCA. The buffering of the PR/W line by both primary and secondary buffers permits visibility of the transceiver direction control signal by either the primary or secondary test bus.

4.2.3 System Area Tests

Before testing the secondary area, the primary area must be certified as fault-free. On-board testing proceeds in a bottom-up approach. One of the first peripherals to be tested is the primary ACIA since it is used to help display error messages. Test status

for this device is indicated by a firmware-controlled test LED since normal diagnostic messages can not be sent to a terminal if the ACIA or RS-232 interface is dead. A series of "on-off" blinks is used to indicate this test status. Testing the board components in this bottom-up fashion reduces the risk of fault masking and eliminates possible faulty component areas.

Once the primary area is certified as fault-free, the secondary area is tested using a combination of both boundary and non-boundary-scan test procedures [Lebak 1991]. The secondary area consists of a secondary RAM, ROM, and ACIA and all other peripherals such as a Programmable Timer Module (PTM), and a Peripheral Interface Adapter (PIA) that permits parallel I/O capabilities via two 8-bit bidirectional ports. The on-board microprocessor controls all secondary test operations via the boundary-scan test bus controller or by the use of peripheral I/O wraparounds. A boundary-scan test connector provided an interface to TI's ASSET system that permitted diagnosis of faults in both the primary and secondary areas.

As alluded to earlier, no hardware support for the testing of peripheral interrupt request lines was designed into the DFT system. This support would have required additional open-collector buffers to provide the wire-ORed capability these signals required. (The SCOPE buffers do not have open-collector outputs.) Instead, software routines verify interrupt request line connections. Successful interrupt requests are confirmed by changing the interrupt request vector to point to a test software routine that sets a test register (IFLG). The interrupt is then generated, and this register is checked for the presence of a logic one to determine if such a branch was taken [Lebak 1991].

4.3 Test Control via LCA

4.3.1 LCA Description

To reduce hardware real-estate and provide a consistent approach to test control, a Field Programmable Gate Array (FPGA) was chosen to facilitate the test process. The FPGA that was chosen was a product from Xilinx (3000 series) that programmed itself upon power-up via a serial ROM. This reprogram feature was used to effect a complete hardware reset of the entire system via a pushbutton reset switch and appropriate reset circuitry.

Xilinx's Logic Cell Arrays (LCAs) are best described as user-definable high-density ASICs. Whereas most Programmable Logic Devices (PLD) replace five to ten SSI/MSI devices, LCAs offer much greater logic densities (tens of thousands of gates), a greater number of I/O pins, and toggle rates ranging from 50 to 125 MHz [Xilinx 1991]. This enabled the author to incorporate a myriad of functions into a single programmable device (to be programmed at a later date) while conserving precious board real-estate. Although the Xilinx LCA's incorporate a higher logic density than the DFT design required, the proportionally higher pin count of these devices was completely used by the DFT system.

4.3.2 LCA Uses

This section discusses the areas in which the LCA proved invaluable. Throughout this LCA section, the terms *programmed* and *configured* are used interchangeably. Many issues are discussed briefly since they are covered in detail later (e.g. the system reset and the on-board test interface). However, portions that pertinent only to the LCA are discussed in depth.

The LCA provided many advantages such as configuration registers in place of hardware jumpers. This eliminated the need to both move and remember where hardware jumpers were placed. Hardware jumpers were effectively replaced by software jumpers and all default settings can be reloaded on power-up via the LCA reprogramming cycle. In addition, certain signals may be controlled and/or observed through registers designed within the LCA. These include (but are not limited to) the enables for the buffers, the baud rate clock for two serial ports, and the peripheral chip selects. The LCA also facilitated communication between TI's 16-bit TBC and Motorola's 8-bit microprocessor by serving as an interface to scannable flip-flops that controlled the upper eight bits of the TBC's data bus. The 80 different LCA signals incorporated into the LCA design are shown in Figure 12.

The two LCA SCOPE octal buffers (UP7 and US12) shown in Figure 12 provide visibility and controllability for signals indigenous to their areas⁵. This list of signals includes three primary and five secondary active-low chip selects (PACIA*, PRIRAM*, PRIROM*, SPTM*, SPIA*, SECRAM*, SECROM*, and SACIA*) and the edge connector and primary reset signals (ERST* and PRST*). In addition, the baud rate generator's frequency (MAXFREQ), the primary ACIA transmit and receive signals (PRXD, PTXD) and the secondary wrap-around control signal (SBUFCTRL) are all scannable. In both cases, UP7 and US12 are the last scannable buffers in their respective scan chains.

To minimize the probability of the inadvertent reconfiguring of the test board, the firmware has the capability to selectively enable and/or disable features contained within

⁵ These scannable buffers could have been eliminated if a scannable LCA was used in place of this non-scannable one. The Xilinx 4000 series LCAs are IEEE 1149.1 conformable. However, these devices were not available when the LCA circuitry was being designed.

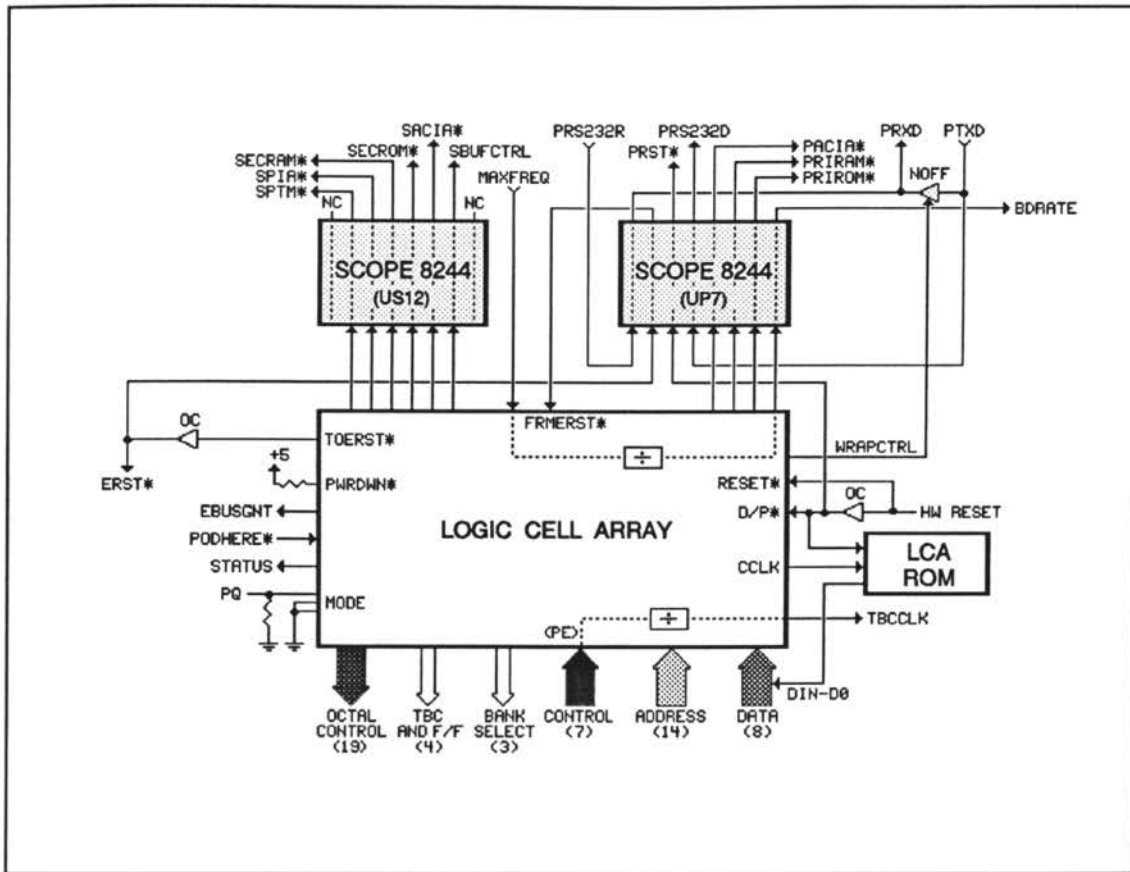


Figure 12. LCA Block Diagram.

the LCA. For example, electrical wraparounds for serial I/O devices were controlled via LCA firmware (WRAPCTRL for the primary ACIA's TxD → RxD, and SBUFCTRL to disable the secondary ACIA's RxD line). Most of the LCA registers are not available within the microprocessor address space unless the "LCA_test" register (an LCA-internal register defined by Lebak) has a specific "test mode" value [Lebak 1991].

At power-on all control signals have deterministic conditions. Buffers necessary for primary operation are enabled. Those buffers that are not necessary for primary operation and have the possibility of interfering with proper microprocessor operation are disabled. These buffers are later enabled via LCA firmware.

The incorporation of both the primary control and address busses into the LCA allowed reconfiguration of the system memory map by providing the necessary signals

to decode peripheral chip selects. Data bus connections to the LCA were also included, since the LCA provides support for the test software, on-chip read/write registers, and sources the serial and TBC clock signals. The serial baud rate frequency was divided via a user-configurable register or counter within the LCA. A frequency of 153.6 kHz was sent from the bit rate generator through the LCA so that an ACIA "divide by 16" choice produces a frequency of 9600 Hz.

Since the LCA is pin-limited, the serial versus parallel configuration method provided the smallest pin count penalty. To further reduce the number of required configuration pins, some LCA signals performed dual functions based on the LCA's program status. For example, the LCA signal M2-PQ is used to select the master configuration mode *prior* to configuration, and is used as an input for the primary Q clock (PQ) *after* configuration. Other signals performing double duty include the High During Configuration (HDC) or PODHERE* signal, and the Low During Configuration (LDC*) or UP7 buffer enable. These dual function pins permitted visibility or control of additional I/O pins that would normally be reserved for configuration purposes. Special care had to be taken in the case of the M2-PQ signal so as not to disturb its grounded state prior to configuration. During configuration PQ becomes three-stated allowing M2 to become strapped to ground via a resistor. After programming is complete, the pin becomes the microprocessor PQ signal.

An active PODHERE* signal indicates that the ASSET system is present for off-board testing. This signal permits the LCA to observe the presence of the ASSET system so that the test STATUS output (test LED control signal) can be used to indicate this presence as well the status of on-board tests.

Support for on-board testing is achieved by controlling communication between TI's 16-bit TBC and Motorola's 8-bit microprocessor. A test register within the LCA controls four interface signals to the scannable flip-flops and TBC. Dedicated TBC read and write subroutines control the assertion and negation of these interface signals.

Maintaining system memory-map compatibility with KSU's current microprocessor system and providing enough memory space for both the system and test firmware necessitated the use of a page-mapped 32K x 8 primary EPROM. This EPROM was large enough to "house" all of the ROM firmware (SUPERmon, Mother Hen, and test routines). However, the address space compatibility was such that not all of this 32K memory was not directly addressable without the use of page-mapped hardware. To address this problem, the LCA provides support for memory bank selection. The three primary address lines (PA14, PA13, and PA12) are routed through the LCA so that during normal circumstances (when SUPERmon is present) the three primary bank select address lines exiting the LCA (PBSA14, PBSA13, and PBSA12) are the same three lines entering. When the board is in test mode these address lines will be translated differently to allow banks of test software to be selectively enabled [Lebak 1991].

Finally, the nature of the serial reprogramming cycle made it necessary for the LCA to become the source of the DFT system's reset. Immediately following an LCA initialization cycle, the microprocessor and peripherals had to be configured to a known state. Two signals are responsible for this system reset--PRST* for the primary area, and SRST* (buffered ERST*) for the secondary area. Furthermore, the LCA can monitor a hardware reset (RESET* and DONE/PROGRAM*) or an edge connector reset (FRMERST*), or it can generate an edge connector reset (TOERST*).

KSU's current microprocessor system provided hardware flexibility by incorporating almost 30 sets of jumpers on the board to select a myriad of functions. Configuration choices include a +5 and/or ± 12 V serial power supply connection, baud rate speed selection, serial and parallel port configurations, RAM/ROM configurations, and address decoding for peripheral chip selects.

The author determined that the DFT design should also provide most of the flexibility that the hardware jumpers provided, but not at the cost they consumed⁶. Jumpers can provide total flexibility, but at a cost of board real-estate, reliability, and a lack of testability. The LCA provided the flexibility of jumpers by implementing hardware functions in firmware. This "soft" hardware could be reconfigured to permit future system updates, or to provide additional features not originally anticipated.

However, not all of the current microprocessor hardware jumpers were replaced by LCA firmware, in many cases it was easier to provide configuration functions elsewhere. For example, providing serial power supply connections via the LCA could prove to be a "FRYghtful" experience. Instead, serial power connections are provided by J6, a "solder pad jumper" that allows solder connections between 5 or ± 12 V power supplies and serial connector pins. Memory configuration is provided by the way in which the memory sockets are wired. (See the section entitled *Memory* on page 24 for more information.)

To provide as much flexibility as possible, the DFT design proceeded on the assumption that the 100-pin XC3042 would be used. This package provided up to 82 user I/O pins and 4200 useable gates--more than enough for the relatively simple combinational functions that the PLD would need to support [Xilinx 1991]. Later it was

⁶ All 24-pin memory devices like 2708s and 2716s are no longer permitted. Instead, only memory devices conforming to the JEDEC approved 28-pin DIP pinout (e.g. 2764s and 27256s) are allowed.

discovered that the 100-pin version was not available in a socketable PLCC or PGA. This device was only available in Surface Mount Device (SMD) packages like Quad Flat Packages (QFP). Although the SMD consumed little real-estate, this package could not be used because:

- A testable board with self-test diagnostic capabilities should have the capability to *replace* bad ICs once fault isolation to the component level has been determined--this requires socketable ICs.
- KSU does not have the equipment that is required to wave-solder a SMD to a PWB.

An investigation of the next larger socketable LCA size--a 132-pin PGA, revealed a real-estate penalty due to its large footprint. Finally, the 84-pin package that is available in both a Pin Grid Array (PGA) and a Plastic Leadless Chip Carrier (PLCC) was chosen. Although less user I/O pins would be available, the penalty was not as great as initially assumed. The 100-pin package provided only 88 signal pins of which 82 were available for I/O (six are dedicated to configuration or power conservation and 12 are dedicated to power). The 84-pin package contains just four power pins leaving 80 signal pins and 74 user I/O pins [Xilinx 1991]. Therefore, only eight user I/O pins were lost by the migration to this smaller package.

To reach this new objective, several steps were taken to minimize the number of required LCA I/O lines and remove the less critical ones. Signals with similar functions were combined and unnecessary or obsolete signals were removed. The criteria for pin removal was simply that the loss of the signal pin should not impact the LCA's ability to perform its chip select logic and other responsibilities.

As depicted in Figure 11 on page 50, the bus driver control lines were grouped together to reduce the number of sourced signals from a possible 35 ($19 * 2$ control lines per octal, minus three data transceiver direction pins) down to 19. Furthermore, since address decoding was performed by the LCA, a single chip select was sourced to each peripheral. To maintain a degree of consistency, all these LCA sourced chip selects are active-low, i.e. all the active-high peripheral chip selects are wired active via pull-ups. During LCA configuration, all LCA I/O pins are pulled-high, which of course, disables any active-low signals.

The lower two address pins (A0 and A1) were casualties of the new pin limitation, i.e. they were not included in the LCA design. This produced no adverse effects on the peripherals, because their register select inputs were hardwired to the lower address lines as shown in Table 9. However, it did affect LCA operations. Each LCA register is located in four microprocessor addresses. Furthermore, the LCA cannot distinguish between a NMI or a reset acknowledge by the microprocessor. The MC6809 permits an external device to determine the microprocessor's state of operation. The microprocessor signals BA and BS indicate a normal state of operation, an interrupt or reset acknowledge, a synch acknowledge, or a halt acknowledge. Address lines A1 through A3 can be decoded to provide additional information--like the difference between an interrupt or reset request [Motorola 1983]. Since the LCA does not receive A1, it cannot make this determination.

The removal of other address lines would have been more costly. For example, if A12 was not included, then the \$Exxx addresses could not be distinguished from \$Fxxx addresses. This would not be a problem for write operations since SUPERmon is located in ROM at \$F000 - \$FFFF, but read operations would have caused data bus contentions.

Table 9. Peripheral Register Select/Address Combinations.

Address	MC6821 (PIA)	MC6840 (PTM)	MC6850 (ACIA)
A2		R2	
A1	R1	R1	
A0	R0	R0	R

Data bus lines could have been removed (there is no special reason eight data bits have to be routed to the LCA), but this is a convenient size for an 8-bit microprocessor to digest, and it would be somewhat awkward from a software standpoint if the data bus was not a multiple of eight bits.

4.3.3 LCA Configuration

4.3.3.1 Method

An LCA configuration mode is determined by the logic levels applied to the three mode selection inputs (M1, M2, and M3) prior to configuration. Five LCA configuration modes are available. These modes are divided into master, peripheral, and slave modes. In master mode, the configuration program is loaded from an external memory--in a parallel fashion via an EPROM, or serially via a serial ROM. The microprocessor may directly configure the LCA in the peripheral and slave modes. In addition, voltage thresholds for LCA pins can be configured as either TTL or CMOS-compatible [Xilinx 1991].

To reduce programming complexity, processor intervention, and reduce the LCA board real-estate, the master serial method of programming was chosen. This method utilizes an 8-pin configuration ROM (XC1736) to serially program the LCA. An LCA

internal crystal oscillator generates a configuration clock for the synchronization of serial data transfers. The serial ROM provides enough memory to program two XC3020s, or either one XC3030 or XC3042 [Xilinx 1991].

The Xilinx XC3020, XC3030, and XC3042 are pin compatible and are available in the 84-pin PLCC package (though the 3020 has many unconnected pins and the 3042 has many unbounded pins). This permits the use of a less expensive FPGA (generally, the less dense the LCA, the less the cost). However, the 3020 cannot be used on the DFT design since it does not have the required number of user I/O signals, i.e. many pins are left unconnected, but the XC3030 has a sufficient number of user I/O pins and gates.

4.3.3.2 Problems

Initial LCA configuration introduced new design problems. Special care had to be taken so that the microprocessor did not come out of reset before the LCA did. Otherwise the microprocessor chip-select for the ROM may not be configured before the first ROM instruction fetch. In addition, the active-low enables on the buffers surrounding the microprocessor are sourced by the LCA. During configuration, all non-configuration LCA pins are pulled-high. This effectively disables all of these buffers. Therefore, the microprocessor and peripherals had to be reset to a known state *after* the LCA had completed its initialization--preferably within one or two clock cycles of each other. (This reset problem could have been eliminated through the use of TI's anti-fuse FPGA. Unfortunately the support tools for TI's product line were not available at the start of this project.)

The method chosen to perform the system reset was to assert the active-low reset lines on all the register-based peripherals until LCA configuration was complete. The LCA

sources both board area resets--one for the primary called PRST*, and one for the secondary called SRST*. To complicate matters, all partitioning logic that buffered these device reset lines had to be active during this initialization period.

Several design decisions had to be made to provide a complete system reset. The following assumptions/decisions were made to complete the necessary reset circuitry. Supporting arguments for these assumptions follow.

1. The LCA will completely reconfigure itself every time the hardware reset switch is pressed.
2. Upon power-up, the LCA reconfiguration cycle will last long enough for the crystal oscillator to become fully functional.
3. There will not be any instances when the microprocessor, TBC, and other primary peripherals will be reset independently.
4. No LCA firmware will be used to effect a primary area reset.

Support arguments:

1. This is different from the current microprocessor system's reset circuitry. That system's reset is a "smart reset" in that a complete initialization of system components does not take place every time the hardware reset switch is depressed. In some instances, this can cause problems since the system cannot always accurately ascertain whether a complete reset is required or not. This results in differing initialization states--a condition unacceptable for efficient system testing. To avoid this "smart reset" problem, the LCA is reconfigured every time the reset switch is depressed.

2. The MC6809 requires the crystal oscillator to be fully functioning when it "wakes up" [Motorola 1983]. The MM17 system provides this capability by generating a 500 ms low-level RESET* signal to the MC6809 upon power-up via a one-shot monostable multivibrator [Motorola 1982]. To avoid adding additional reset circuitry to the DFT system, the duration of LCA program cycle should long enough to avoid microprocessor initialization problems⁷.
3. The TBC will only be used in microprocessor system self-test operations. The other peripherals should be reset when the microprocessor is reset. Therefore, all system peripherals and the microprocessor are reset together during a system reset.
4. Using a portion of the LCA to effect a system reset means that a soft or hard failure in this area of the LCA would disable the system reset. Failures of this type may be impossible to diagnose by the host microprocessor since on-board test control is provided by the LCA itself.

To facilitate the reader's understanding of how the reset operation proceeds, the sequence of events that constitute a system reset are described in Table 10. This reset procedure is highly dependent on the way in which LCA reconfiguration is initiated, implemented, and indicated. The LCA provides many signals that can be used to effect an external reset, bus enable or PROM enable during configuration. The D/P*, High During Configuration (HDC) and Low During Configuration (LDC*) signals are active during the configuration of the LCA.

⁷ Since the traditional design incorporates PAL programmable logic (configurable via a fuse programmable array) additional auxiliary board reset circuitry may be necessary to avoid potential microprocessor power-on problems. Alternatives to this approach include depressing the reset switch after the system had powered-up, or adjusting the RC time constant of the power-on reset.

As indicated in Table 10, the high-to-low transition on RESET*, followed by the logic low levels of both this signal and D/P* initiate an LCA reprogram cycle. This combination of signal values is easily accommodated by the hardware reset circuitry and pushbutton switch. The D/P* pin is an open drain I/O signal that can be used both to initiate a reprogram cycle (controls CE* and OE* of the serial ROM) as well as indicate programming status. The hardware reset circuitry uses an open collector 7407 buffer which drives this D/P* line low.

System area resets were implemented differently from the LCA although the end result was the same. As Figure 12 on page 53 indicates, the primary and secondary area resets (PRST* and SRST*) are not the same signal. SRST* is a buffered edge connector reset (ERST*) signal, but PRST* is independent of either of these signals--it is sourced by the LCA program status pin D/P*. PRST* remains asserted *during* LCA reconfiguration, while ERST* is negated *after* LCA configuration has been completed. This drives the grounded SRST* signal high.

In both instances, the reset signals were buffered by SCOPE octals whose output lines could be three-stated by their respective enables. The LDC* program status pin served as a control signal enabling UP7 to buffer the primary reset signal. The secondary reset relies on the operation of signals in both US8 and the LCA. When the LCA begins its reconfiguration cycle, all user I/O pins are pulled high which causes buffer outputs to become three-stated. This reset line was strapped low to disable PIA or PTM activity until SCTRLB2 could be asserted low from the LCA again.

The low-to-high secondary area reset signal is generated by enabling US8 from the LCA (via SCTRLB2) so that ERST* is passed to the strapped SRST* signal. Both system area resets are low during LCA programming so that the register-based peripherals and

Table 10. DFT System Reset Sequence.

-
- The hardware reset switch is depressed for at least 6 μ S or LCA power is cycled from below 0.1 volt to between 2.5 and 3 volts [Xilinx 1991].
 - The LCA's RESET* and DONE/PROGRAM* (D/P*) pins are pulled low. This initiates an LCA reprogram cycle.
 - The LCA is placed in a Master Serial Mode reconfiguration state by the grounded M0, M1, and M2 signals.
 - The LCA holds the D/P* pin low to indicate a new program cycle to the serial ROM, while it three-states all LCA user I/O signals and asserts its Low During Configuration (LDC*) output.
 - UP7's upper output enable is controlled by this LDC* signal. It buffers the D/P* signal (PRST*) which is passed to the both the microprocessor and TBC thereby holding these devices in a reset state. During power-on the microprocessor requires its reset line to be active until the crystal oscillator has become fully operational [Motorola 1983].
 - Similarly, US8's output enables are three-stated since all LCA user I/O pins are pulled high during LCA configuration. This allows SRST* to be pulled low by a pull-down resistor, thereby holding the secondary PIA and PTM in a reset state as well.
 - The serial ROM's output and chip enable lines are held active by the LCA D/P* signal thereby beginning the program cycle.
 - Serial data is sent synchronously (synchronized to the CCLK output) to the LCA's DIN input via the serial ROM's DATA output.
 - When the LCA reprogram cycle is complete, the D/P* signal is asserted high. This disables the serial ROM and initiates a low-to-high transition on PRST* (controlled by this D/P* signal) which permits the microprocessor and TBC to come out of reset.
 - Immediately after the program cycle is complete, SCTRLB2 is asserted and ERST* is negated by the LCA, permitting the PIA and PTM to come out of reset by producing a low-to-high SRST* signal.
-

microprocessor are inactive during this period. The ACIA's contain internal reset circuitry that holds a master reset bit in the control register low until it is set by the microprocessor [Andrews 1982]. Immediately following this LCA program cycle, the resets are driven high, allowing the peripherals to come out of reset. To ensure that the system reset

proceeds as planned, LDC* and SCTRLB2 must be programmed low and ERST* must be programmed high upon LCA configuration. This will permit the primary and secondary resets to be negated.

ERST* is configured as an open collector output. This permits the observation (FRMERST*) as well as generation (TOERST*) of an edge connector reset. Thus, edge connector devices can be reset, and they can request a system reset. Edge connector resets can also be observed via the serial test bus since this signal is buffered by UP7--the primary LCA SCOPE octal.

Initially, one might think that this method would reset the primary and secondary board areas at different times. However, the LCA provides the capability of programming the D/P* line to indicate program completion either one cycle before or after LCA reprogramming has been completed [Xilinx 1991]. If it is programmed to occur one cycle after LCA configuration, then the negation of ERST* (from LCA) can occur almost simultaneously with the D/P* signal's high assertion. Thus effectively resetting all system areas at once.

4.4 DFT Structures

4.4.1 Test Bus

This section describes the implementation of the IEEE 1149.1 serial test bus in the DFT system. The primary focus of this section is to describe the on and off-board test interfaces, how these test busses can be independently or jointly operated, and the visibility and controllability that is realized by incorporating this bus into a microprocessor system. The actual software test routines are discussed in depth by Lebak.

This four-wire test bus provides support for software testing of board interconnects. Both on and off-board test bus control is achieved by the inclusion of the TBC and the dedicated ASSET test connector. The primary test bus is controlled via the dedicated ASSET test connector, while the secondary test bus is controlled by the host microprocessor and the on-board TBC.

4.4.1.1 Host Microprocessor Test Bus Control

4.4.1.1.1 TBC Interface

The secondary area IEEE 1149.1 test bus was under host microprocessor control by utilizing TI's TBC as a primary area peripheral. TI's TBC controls all major boundary-scan functions such as the scan and execution instructions required of all IEEE 1149.1 conformable devices (TI's SCOPE octals). Initially, the DFT octal component strategy was to include only TI's buffer and transceiver SCOPE octals--i.e. until TI's data sheets for the TBC were finalized. They indicated that the TBC was a 16-bit peripheral that provided no upper or lower data byte strobes. This created a communication problem for the 8-bit MC6809. This section focuses on this TBC/microprocessor interface.

To simplify this test control interface, interrupt service requests by the TBC are not permitted, the INT* and RDY* (READY*--indicates times when TBC can be accessed) output pins were not used. To enable the TBC to scan at different rates, the test clock input TCKI is sourced by a firmware controlled LCA signal that can be frequency divided from the microprocessor's 2 MHz E clock.

To rectify this data bus problem, a firmware-controlled 8 ↔ 16-bit interface was constructed. The pertinent LCA, TBC, and flip-flop interconnections that are necessary to control communication to and from the TBC are shown in Figure 13. A dedicated

"READ" (UP2) and "WRITE" (UP1) flip-flop latch the upper data bus byte during read and write operations, while the MC6809's data bus is hardwired directly to the TBC's lower 8 data bits. The upper read and write flip-flops are necessary to present a full 16 bits of data to the TBC at one time. These read and write operations are governed by four interface signals sourced by the LCA. The LCA generates both the TBC's RD* and WR* lines as well as the flip-flops' CLK and OE* lines. Together these components present a 16-bit word to the TBC while maintaining an 8-bit data interface to the microprocessor.

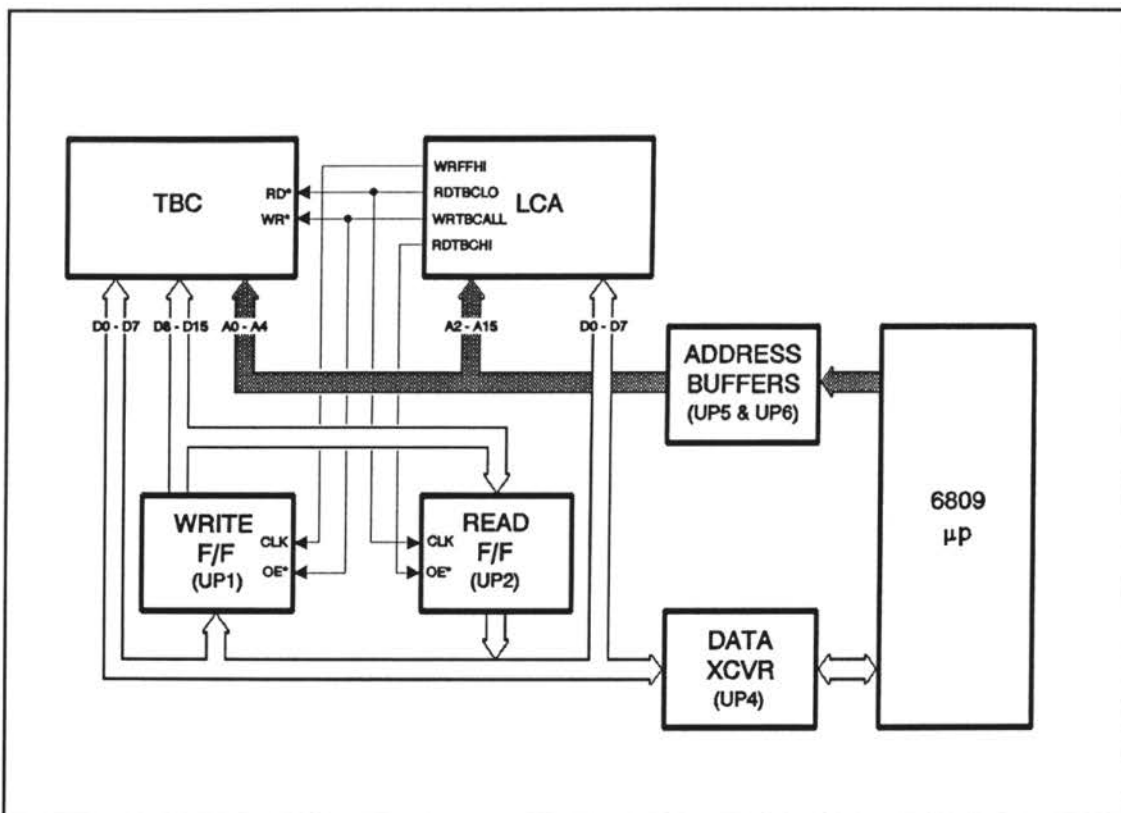


Figure 13. TBC Read/Write Interface.

A read-low first, write-high first protocol was established to enable communication between these dissimilar devices. This protocol required synchronizing the assertion and negation of the interface control signals with the data on the data bus. Each 16-bit TBC word either read or written requires two consecutive microprocessor read or write cycles and one TBC read or write cycle.

For description purposes, the signals will be referenced by their LCA net names. For example, the shared TBC-RD*/UP2-CLK signal will be referenced by the LCA signal name RDTBCLO. The diagram in Figure 13 also shows the address and data bus buffers placed between the microprocessor and its test peripheral. These address and data bus drivers introduce timing delays denoted by T_A and T_D , respectively. The read and write procedures are outlined in Table 11 and Table 12, respectively. The references to the read and write timing delays can be found in Figure 14 and Figure 15. Together, the timing diagrams and procedures illustrate how communication was established with the TBC. Furthermore, since little timing data was available for the TBC, no timing analysis was performed on either the read or write operations, although critical timing delays are pointed out. Therefore, the diagrams are meant to convey the sequence of events that occur between the TBC, flip-flops, LCA, and microprocessor, rather than an actual timing analysis of this interface.

Since this is a testability design, it was only natural to enlist SCOPE octals to perform the dedicated read and write latching operations. This would have the added benefit of permitting a degree of test on the interface that permits host-microprocessor test control operations. Initially the author decided to use TI's 74BCT8373's to perform these interface functions. However, certain potential timing problems can be avoided by using edge-triggered latches like the 74BCT8374s instead of their level triggered counterpart--the 74BCT8373.

This is an important point because if TBC read operations are to be successful, $t_h < t_{ck}$ (see Figure 14), i.e. data must be latched by UP2 *before* the TBC's output is disabled or there will be no data available to latch. Edge-triggered latches like the 74BCT8374 begin latching data sooner than their level-triggered counterparts, so timing problems

Table 11. TBC Read Procedure.

-
- A TBC register is selected for reading via the lower five address lines (addresses \$EC80 through \$EC9F). A T_A time later the TBC receives this address.
 - RDTBCLO is asserted low. A t_{rd} time later the TBC places the *lower* data byte on the data bus for the microprocessor to read a T_D time later.
 - RDTBCLO is asserted high after at least t_{su} . This causes UP2 to latch the *upper* data byte a t_h time later while at the same time causing the TBC to three-state its data bus a t_{ck} time later.
 - RDTBCHI is asserted low. This causes UP2 to place the *upper* data byte on the data bus a t_{oe} time later and the microprocessor reads this data byte a T_D time later.
 - RDTBCHI is asserted high. A t_{od} time later UP2's output is disabled ending the cycle.
-

Table 12. TBC Write Procedure.

-
- A TBC register is selected for writing via the lower five address lines (addresses \$EC80 through \$EC9F). A T_A time later the TBC receives this address.
 - The microprocessor places the *upper* data byte to be written on the data bus for UP1 to receive a T_D time later.
 - WRFFHI is asserted low, then high. This causes UP1 to latch the *upper* data byte a t_h time later.
 - The microprocessor places the *lower* data byte to be written on the data bus for TBC to receive a T_D time later.
 - WRTBCALL is asserted low. This causes UP1 to place the *upper* data byte on the TBC's upper data bus a t_{oe} time later.
 - WRTBCALL is asserted high after at least p_{wwl} . This causes the TBC to write *both* the upper and lower data bytes a t_{wr} time later while at the same time causing UP1 to three-state its output a t_{od} time later ending the cycle.
-

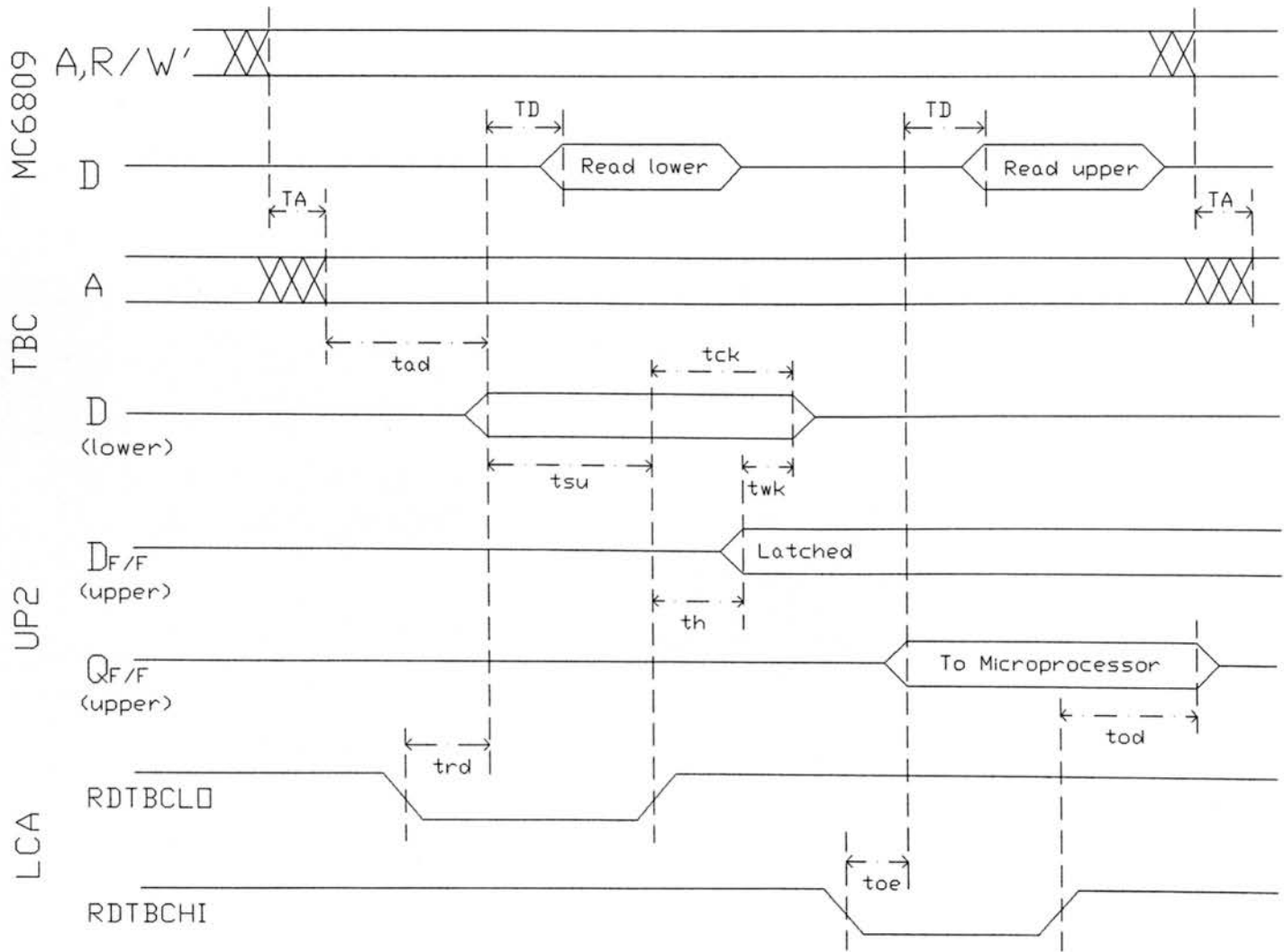


Figure 14. TBC Read Cycle Timings.

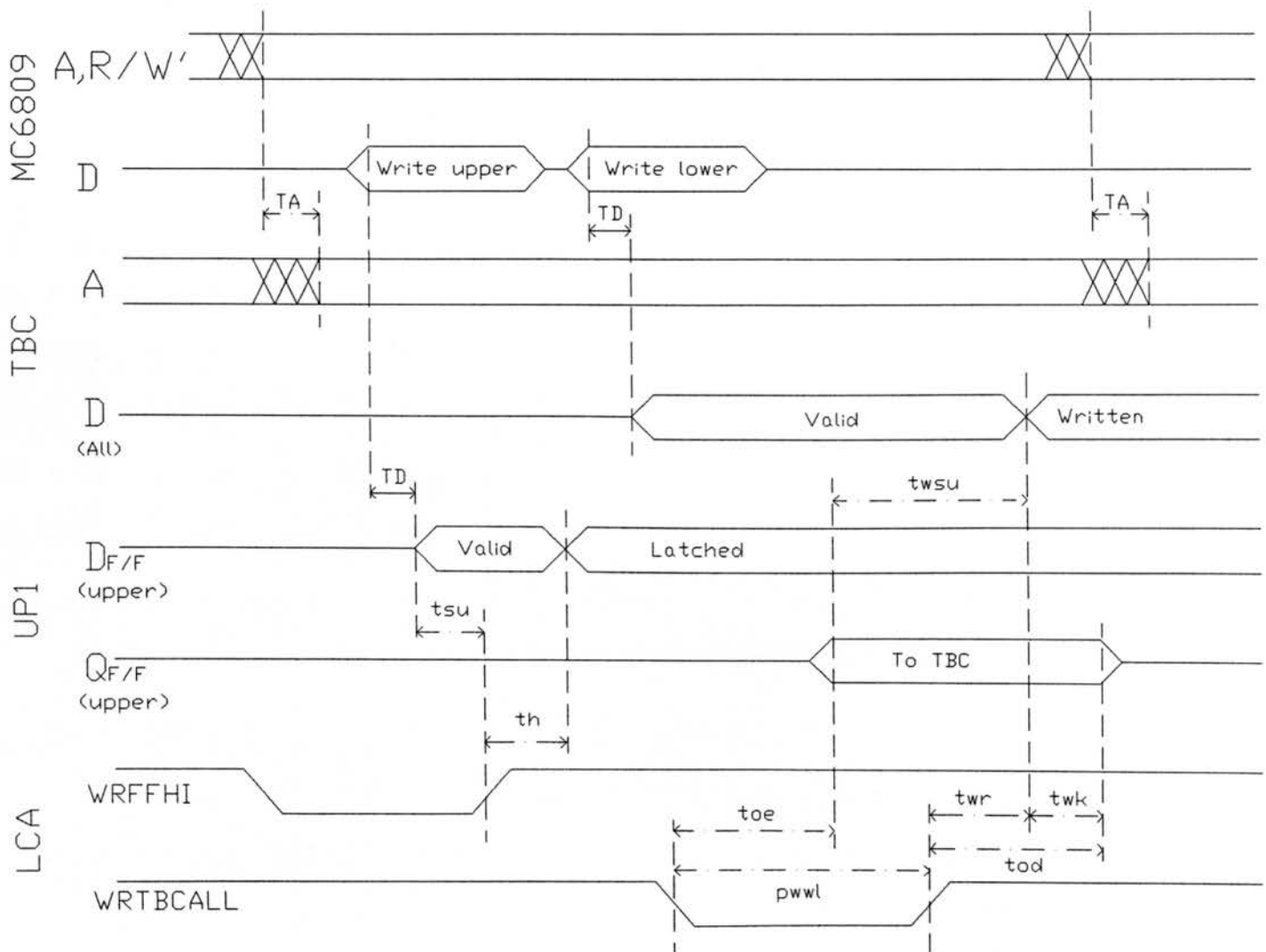


Figure 15. TBC Write Cycle Timings.

should be avoided⁸. An alternate protocol would have been to allocate separate control signals for these inputs (one for TBC-RD*, and one for UP2-CLK). However, due to LCA pin limitations, these signals were combined.

Write operations suffer from a similar *potential* communication problem. Data is latched by UP1 via a dedicated control signal input (WRFFHI), and data is presented to the TBC *before* TBC-WR* is asserted. However, UP1's output enable (OE*) and TBC's WR* strobe are controlled via the same LCA signal--WRTBCALL. TBC-WR* is edge-triggered (rising edge), while UP1-OE* is level-triggered (active-low). Therefore, once again, the TBC must write the data word *before* the UP1's output is disabled or only the lower byte will be written.

The interesting point about this exercise is that none of this elaborate 8-to-16 bit interface would be necessary if TI had only provided upper and lower data strobes on the TBC. An 8-bit microprocessor like the MC6809 could easily interface to the peripheral by synchronizing byte data transfers with the assertion of these two signals.

4.4.1.1.2 Edge Connector

Host microprocessor control of the test bus permitted operational verification of secondary board areas such as the edge connector. Since the microprocessor boards (user and system) are all connected together in a card cage, they all contain card edge connectors [Motorola 1982]. Functional compatibility with the existing microprocessor system was a design requirement, so this edge connector was incorporated into the

⁸ If the edge-triggered latches do not solve the read timing problem, then UP2-CLK could be sourced by WRFFHI, i.e. the PWB trace from LCA-RDTBCLO to UP2-CLK could be cut and a connection between UP2-CLK and UP1-CLK should be made. Once this change is made, a corresponding LCA firmware change will also have to be made.

design. Due to the high failure rate of exposed edge connectors, it was deemed necessary to place isolating SCOPE buffers around the busses to and from this connector. As depicted in Figure 16, the isolating buffers helped partition the board design by effectively separating the busses into primary and secondary areas. This permitted full observation and control of the edge connector data, address, and control busses.

These edge connector SCOPE octals also provide an efficient method of self-testing this connector. Internal pseudo-random pattern generation and parallel signature analysis operation of these devices provided an efficient test technique.

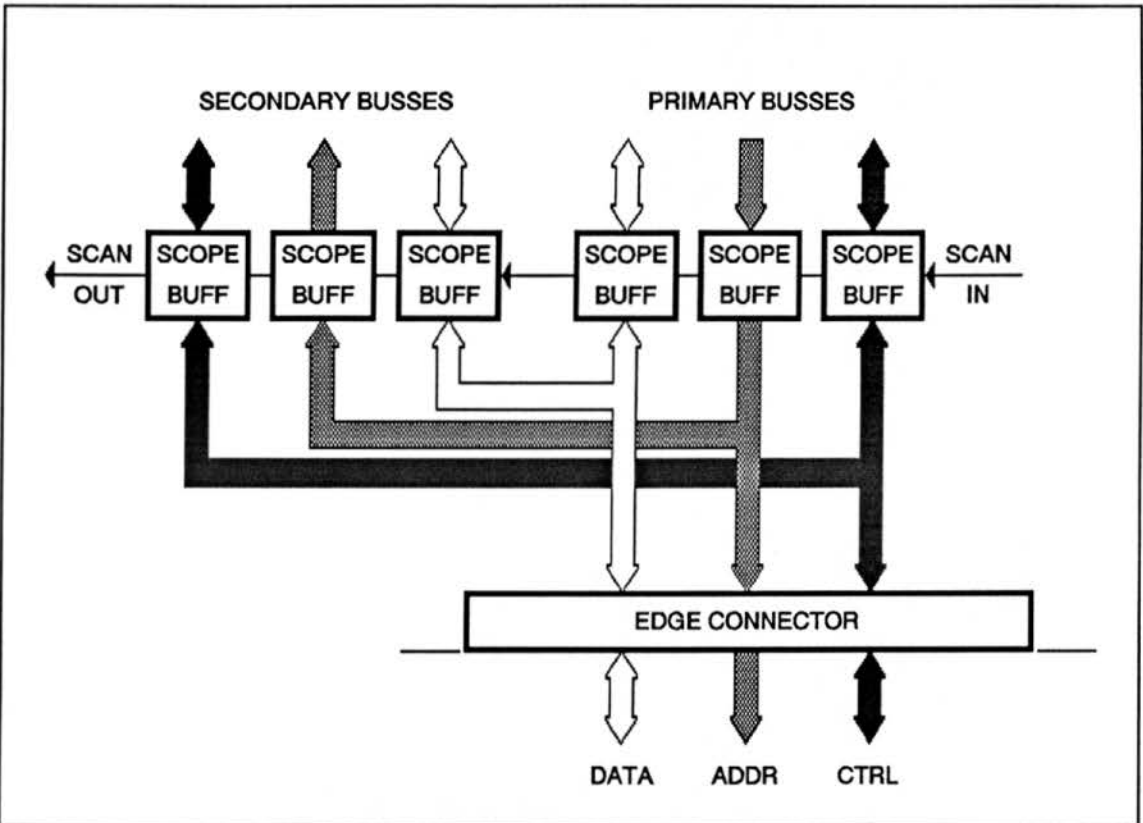


Figure 16. Isolating SCOPE buffers around the edge connector.

4.4.1.2 External Test Bus Control (ASSET)

An edge connector dedicated to the scan testing of board interconnects provides access for off-board testing via ASSET and the boundary-scan chain. Figure 17 illustrates the DFT system test bus connections and how these signals were buffered. When the ASSET system connector is present, the primary and secondary scan chains that serve as partitioning logic are automatically linked into one long scan chain. Since an early hardware objective was to provide a design free of hardware jumpers, no jumpers could be used to implement this test bus connection. Table 13 and Table 14 show how this was achieved while bus contention problems between ASSET and the on-board TBC were avoided. The long ASSET scan chain allows us to supplement the testing of the primary-secondary interface (in the event that enhancement is needed).

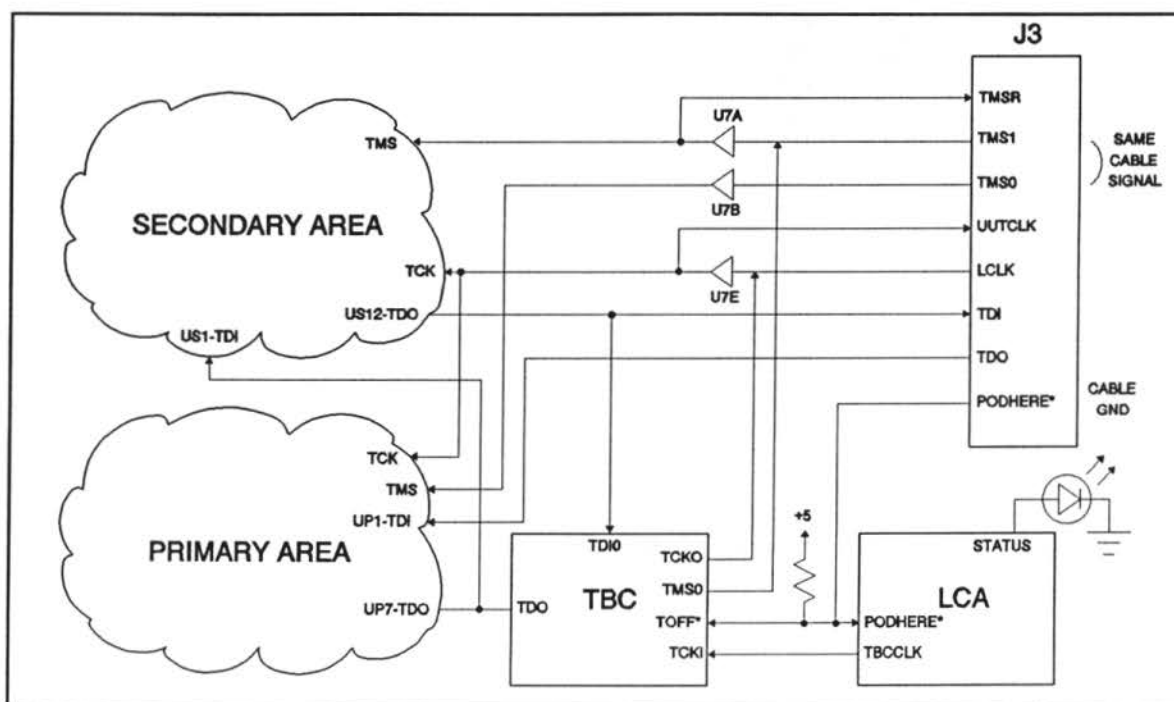


Figure 17. Block Diagram of Test Bus Connections.

The three buffers shown in Figure 17 are 7407 open-collector non-inverting buffers that can sink up to 40 mA of current [TI 1988]. The 1 k Ω PU resistors are not shown in the diagram. Other buffers in the same hex package also served to provide two wired-OR

Table 13. On-board Test Bus Connections.

-
- When the ASSET connector (pod) at J3 is not present, the PODHERE* signal is pulled high via a pull-up resistor. This permits on-board TBC operations.
 - At this point, the microprocessor exercises TBC read and write commands to control on-board scan testing.
 - The TBC receives a test clock input (TCKI) derived from the E clock, sourced by the LCA TBCCLK signal.
 - The TBC clock output (TCKO) is buffered by U7E and broadcast to all system area TCK component inputs. Since the ASSET pod is not present, the J3-LCLK and TBC-TCKO signals do not conflict.
 - The other broadcast signal--the secondary area TMS signal (TBC-TMS0) is buffered by U7A and passed to all the secondary scan components. Again, signal contentions between TBC-TMS0 and J3-TMS1 are avoided since no connection is made at J3.
 - Serial test vectors are shifted out of TBC-TDO and passed to TDI on the first SCOPE octal in the secondary area (US1-TDI). At this point, the last scannable octal in the primary area (UP7), is three-stated since its TMS line (J3-TMS0) is pulled high via a pull-up resistor (at the output of the open collector buffer). Therefore, TDO signal contentions between UP7 and the TBC are avoided.
 - The secondary area's last scan chain component is US12. Its serial output (US12-TDO) is transmitted to one of the TBC's scan inputs (TDI0) which completes the secondary scan chain.
-

connections for the Mother Hen ACIA (secondary) transmit line, and the edge connector reset.

Although TCK is broadcast to all system components, two separate system area TMS lines emanate from the test connector J3. These two TMS signals help prevent bus contention problems between the ASSET test connections and the TBC test connections. As Table 13 indicates, when on-board tests are performed, the primary area TMS signals are inactive. These two TMS signals are electrically connected together within the ASSET test cable. Common cable connections could have been avoided by joining the J3 edge connector signals TMS0 and TMS1 together. However, since both ASSET and TBC test bus connections utilize the same buffers, this connection would have enabled the primary

Table 14. Off-board Test Bus Connections.

-
- When the ASSET connector (pod) is present, the PODHERE* signal is pulled low via a ground signal from the ASSET cable. This disables TBC operations, and three-states all of that device's outputs. The LCA status output can now be used to indicate that the ASSET pod is present.
 - LCLK is buffered by U7E and broadcast to all TCK component inputs. For ASSET resynchronization purposes, J3-UUTCLK also receives this buffered signal.
 - The two TMS signals from J3 (J3-TMS0 and J3-TMS1) are actually the same TMS signal from ASSET--they are connected together in the cable. These signals are buffered by U7B and U7A and passed to the primary and secondary areas, respectively. For TMS resynchronization purposes, ASSET receives a TMS return signal.
 - Serial test vectors are shifted out of J3-TDO and passed to TDI on the first primary area SCOPE octal (UP1-TDI).
 - After information is serially shifted throughout the primary scan chain, UP7 passes this information to the secondary scan chain via a UP7-TDO → US1-TDI connection. Since all of the TBC's outputs are three-stated, TDO signal contentions between UP7 and the TBC are avoided.
 - The secondary area's last scan chain component is US12. Its serial output (US12-TDO) is transmitted to J3-TDI which completes this combined primary/secondary scan chain.
-

area TMS signals--and the SCOPE octal TDO outputs. Therefore, the connection between these signals must take place within the cable. Similarly, a cable ground signal is used to indicate the presence of the ASSET test connector (PODHERE*). The TBC data sheets do not clearly state whether TOFF* has an internal pull-up, so one was added.

The combined ASSET scan chain completely isolates the microprocessor from its peripherals. Although no core components have embedded test structures, visibility and control of the RAM, ROM, and peripherals is still achieved by controlling the address, data, and control busses. Certain signals were placed in the primary area scan chain so that the microprocessor would not have access to these signals via the secondary scan chain. For example, the primary area reset (PRST*) is both controllable and observable

from the primary area scan chain only--that is via ASSET. This prevents the microprocessor from inadvertently resetting both itself and the TBC while secondary area tests are performed.

4.4.1.3 Buffering of Test Bus Signals

As shown in Figure 17, the broadcast ASSET signals are buffered to improve their fan-out capability. The serial TDO signal drives a single BCT load and thus is not buffered. However, both TMS and TCK are buffered. TMS1 drives 12 octal TMS inputs, while LCLK drives all 19 SCOPE octal TCK inputs and one cable signal back to the ASSET system.

As indicated by (1), the open-collector 7407 PU resistor values were calculated based on a fan-out requirement of 20⁹. TI's *TTL Logic Data Book* [TI 1988] and their *Advanced Interface Logic and Bus Interface Logic Data Book* [TI 1991] were consulted for the 7407 and BCT electrical characteristics. (2) and (3) show the maximum and minimum PU resistance calculations for the test interface buffers. Since R_{PU} must be between 126 and 1044 ohms, i.e. $126 \Omega \leq R_{PU} \leq 1044 \Omega$, an arbitrary resistance of 470 Ω was chosen. Faster switching speeds are gained by choosing lower values, while higher values yield lower power dissipation [Motorola 1989b].

$$R_{PU(max)} = \frac{V_{CC(min)} - V_{OH(DRVmin)}}{M \cdot I_{OH(DRV)} + N \cdot I_{IH(RCV)}}, \quad R_{PU(min)} = \frac{V_{CC(max)} - V_{OL(DRVmax)}}{I_{OL(DRV)} + N \cdot I_{IH(RCV)}}$$

(1) Maximum and Minimum Open-Collector PU Resistance Equations [Motorola 1989b].

⁹ These equations were derived from a set of normalized TTL load pull-up resistor equations listed in Motorola's *Fast and LS TTL Data Book* [Motorola 1989b, p. 3-4].

$$R_{PU(max)} = \frac{(4.75 - 2.4) V}{1 \bullet 0.25 mA + 20 \bullet 100 \mu A} = 1044 \Omega$$

(2) Maximum PU Resistance.

$$R_{PU(min)} = \frac{(5.25 - .7) V}{0.40 mA - 19 \bullet 200 \mu A} = 126 \Omega$$

(3) Minimum PU resistance.

where the Drive and Receive values are:

<u>7407 (DRV)</u>	<u>BCT Octals (RCV)</u>	<u>General Params</u>
$V_{CC(max)} = 5.25 V$	$I_{IH} = 100 \mu A$	$N = 20$ (# inputs)
$V_{CC(min)} = 4.75 V$	$I_{IL} = 200 \mu A$	$M = 1$ (# wire-OR outputs)
$I_{OL} = 40 mA$		
$I_{OH} = .25 mA$		

4.4.2 Electronic Wrap-arounds

In addition to boundary-scan test features, non-boundary-scan test features are incorporated utilizing existing on-board components in the test process. This reduced system test overhead. Traditional testing of I/O channels requires the use of mechanical wrap-arounds. In the DFT design, three-stateable electronic wrap-arounds connect controllable peripheral outputs to observable peripheral inputs under firmware control (at speed). One limitation of this technique is that an external device connected to these pins might drive the signal with a lower impedance than the wrap-around buffers. This would lead to an erroneous diagnosis of the system. To address this difficulty the test software contains prompts for the user to remove all external device connections in the event that this situation occurs.

These electronic wrap-around tests permit the detection of manufacturing faults like stuck-at faults, shorts, and opens--which can amount to about half of the manufacturing defects normally experienced [HP 1990]. The measure of self-test operations that these structures permit, help determine if these components are faulty. References in this

section are made to LCA-specific signals. These signals can be found in the schematics or in Figure 12 on page 53.

4.4.2.1 Serial Wrap-Around Test Structures

Figure 18a illustrates such a wrap-around technique that can be applied in the case of unidirectional signals. The serial capabilities of the ACIA can easily be monitored by employing a normally on (NON) SCOPE octal buffer and a normally off (NOFF) three-statable buffer to electronically wrap an output around to an input. During non-test operations, the SCOPE buffer (2A4 → 2Y4 in UP7) is on and the three-state 74LS125 buffer is off. When wrap-around tests are performed the upper nibble of UP7 is three-stated and the wrap-around buffer is active. This prevents signal contention problems from the edge connector while these tests are performed.

In both serial electronic wrap-around cases (primary and secondary), the 74LS125's output enable was controlled from a single LCA output signal, while a SCOPE buffer output enable controlled that buffer and three others in the same nibble.

To permit primary area ACIA test operations, certain nonessential signals had to be grouped together in the same nibble as the ACIA's SCOPE buffered Rx/D signal. This was necessary because the other three nibble signals would be three-stated as well during these test operations. Obviously the PACIA* and BDRATE signals had to be located elsewhere--they must both be active during ACIA terminal testing.

Eventually, it was determined that the reset signals PRST* and FRMERST* could be deactivated during serial wrap-around tests because it was assumed that no reset would be generated either on or off-board (or accepted) while the primary ACIA is talking to itself. However, to prevent the primary area from being reset during these test operations,

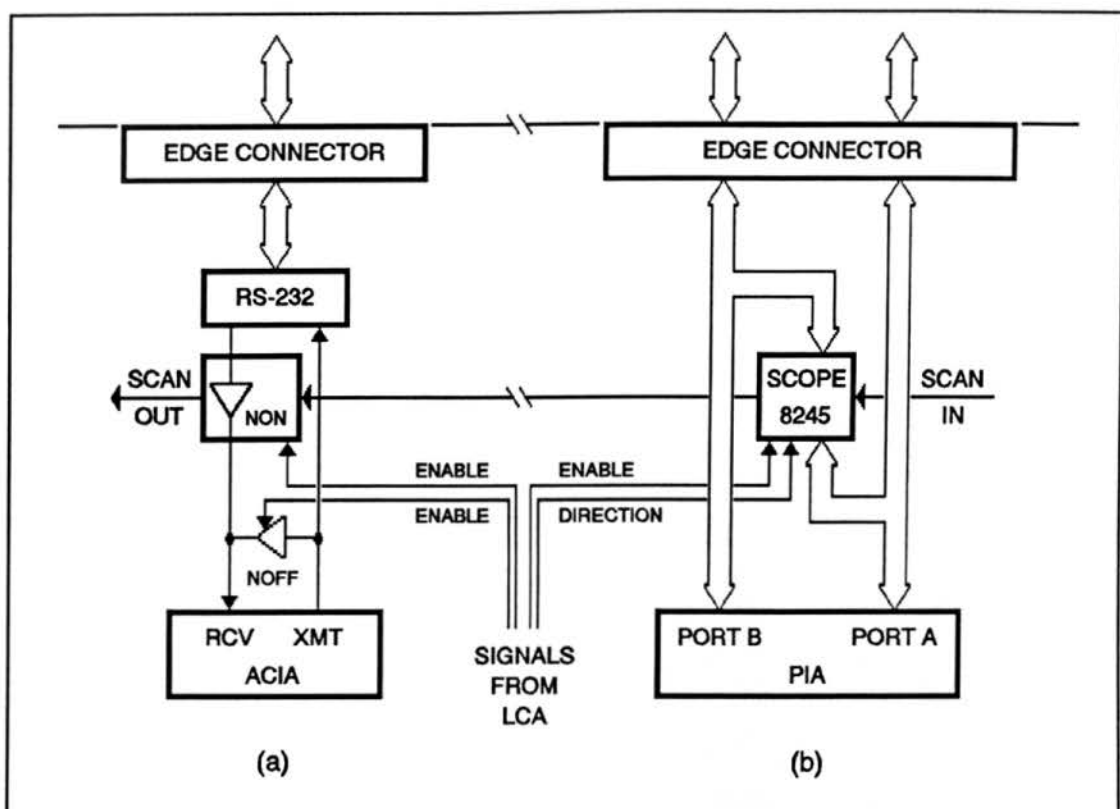


Figure 18. Uni- and Bi-directional Signal Wrap-Around Techniques.

a pull-up resistor on the buffered side of the primary LCA octal was added.

The 74LS125 three-state buffer was used on the DFT design as the normally off (NOFF) TxD → RxD wrap-around buffer instead of a SCOPE buffer, because the microprocessor does not have access to its primary area test bus—only the secondary area test bus. This buffer is controlled by a WRAPCTRL signal from the LCA. Although the SCOPE buffer's output enable (PLCAB2) could still be controlled by the microprocessor, no additional benefit would be realized by using a scannable wrap-around buffer if the host microprocessor was the test processor. Only when off-board testing via ASSET was being performed could benefits of this type be achieved.

A similar wrap-around technique was performed on the secondary ACIA. However, the placement of the SCOPE buffer did make a difference in these tests. Since the secondary scan chain was under host microprocessor control, the ACIA transmit and

receive lines could be directly controlled and/or observed by making the wrap-around buffer scannable. Therefore, the placement of the SCOPE/74LS125 buffers was exactly the opposite to that of the primary ACIA, i.e. a SCOPE octal buffer (1A4 → 1Y4 in UP10) was used to perform the NOFF wrap-around connection and the NON buffer was a 74LS125 buffer. An LCA LBUFCTRL signal controlled the 74LS125 NON buffer, while the LCA octal control signal SACIAB1 (US10's lower output enable) controlled the NOFF SCOPE wrap-around buffer.

Another reason the primary area wrap-around technique was not used in the secondary area was because the LCA's secondary octal cannot be three-stated due to the large number of chip selects that it buffers (five in all). If there was a set of four signals that could be grouped together (including the wrap-around buffer) that could be three-stated then the same technique could be applied to the secondary ACIAs as well. In addition, the testability benefits mentioned above are derived when the secondary wrap-around buffer is a SCOPE buffer.

4.4.2.2 Parallel Wrap-Around Test Structures

Parallel I/O in the form of Motorola's PIA can also be used in an electronic wrap-around fashion to test neighboring components or itself by employing a similar technique via three-stateable buffer packages (see Figure 18b). Ports A and B are bidirectional so they can each be configured to read and write at separate times to perform a self-diagnosis. Furthermore, SCOPE octal buffers are utilized, permitting boundary-scan test techniques and providing a moderate amount of self-check on the electronic wrap-around buffers.

SPIAB1 controls the direction of the wrap-around transceiver (US11), while SPIAB2 controls the its output enable. Unlike the ACIA wrap-around tests, no connector can be present for the PIA wrap-around tests to ensure that signal contentions are avoided. Therefore, the test software prompts the user to remove any external device connections.

The 8-bit size of these ports and their bidirectional nature made the addition of such a self-test capability fairly simple. However, the PIA also has four I/O port control lines or handshake lines. Two lines (CA1 and CB1) are unidirectional, and two are bidirectional (CA2 and CB2). The diagram in Figure 19 shows how visibility and control of these lines was achieved with scannable buffers.

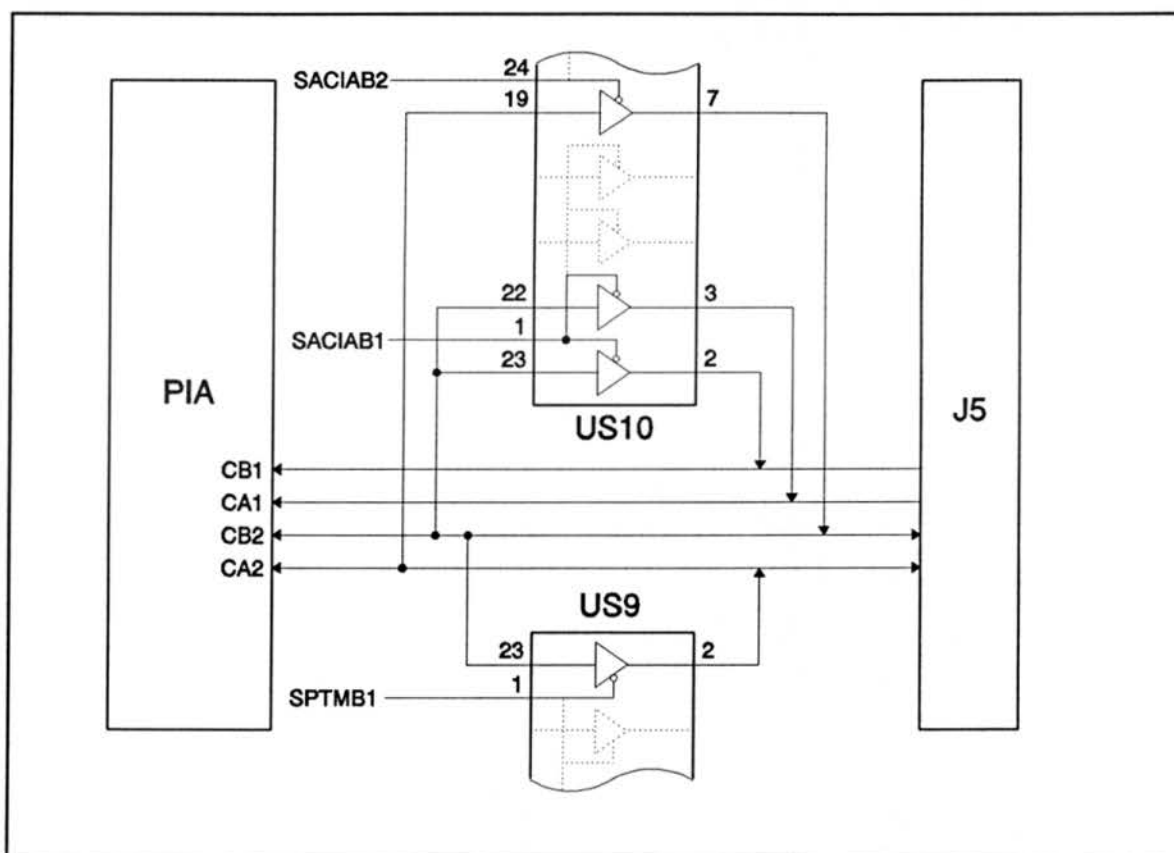


Figure 19. PIA Control Line Wrap-Around Techniques.

US9 and US10 permit either CA2 or CB2 to control the other three handshake lines. If CB2 is configured as an output, then the assertion of SSCIAB1 enables CB2 to be

wrapped around to CA1 and CB1. Furthermore, if CA2 is configured as an input, SPTMB1 enables a SCOPE buffer that will create a CB2 → CA2 connection. Similarly, CA2 can be configured as an output and CB2 can be configured as an input. If SACIAB2 and SACIAB1 are both asserted, then CA2 can control CB2, CA1, and CB1.

Care must be taken to prevent certain potential signal contention problems. For example, CA2 and CB2 should never both be configured as outputs or inputs during test mode at the same time. If these bidirectional control lines are configured as outputs, then the assertion of either SACIAB2 or SPTMB1 would cause a US10/PIA signal contention in one case, and a US9/PIA contest in the other. If both are configured as inputs, then no wrap-around tests can be performed. This of course, is the responsibility of the test software engineer (Lebak).

However, certain hardware design measures were also necessary. To avoid signal contention problems, it became necessary to separate certain electrical test connections from one another. The ten separate SCOPE peripheral test control signals and the functions they perform are listed in Table 15.

It was especially important to control the PIA wrap-around control signals independently. For example, the CB2 to CA2 buffer connection must be controlled independent of the CA2 to CB2 connection. Otherwise, the PIA bidirectional signal configured as an output would conflict with its corresponding bidirectional signal's wrap-around. For example, if SACIAB2 and SPTMB1 (the CA2 → CB2 and CB2 → CA2 output enables) were controlled from the same source, then when CA2 controlled CB2, CB2 signals would get passed back to CA2 via a US9 buffer. This of course would produce yet another signal contention.

Table 15. SCOPE Octal Peripheral Test Control Signals.

Signal Mnemonic	NON/NOFF (WakeUp)	Signal Name and Description
PLCAB1	NON (ON)	PRIMARY LCA BSR1 CONTROL - Active enables primary status/control signals (PACIA*, PRIRAM*, PRIROM*, BDRATE).
PLCAB2	NON (OFF)	PRIMARY LCA BSR2 CONTROL - Active enables primary status/control signals (PRST*, Primary ACIA: RxD & TxD, and FRMERST*).
SPTMB1	NOFF (OFF)	SECONDARY PTM BSR1 CONTROL - Enables PIA port B's control of PTM inputs (PIA-CB2 → PIA-CA2, PIA-B2 → PTM-C3*, PIA-B3 → PTM-G3*).
SPTMB2	NOFF (OFF)	SECONDARY PTM BSR2 CONTROL - Enables PIA port B's control of PTM inputs (PIA-B4 → PTM-C2*, PIA-B5 → PTM-G2*, PIA-B6 → PTM-C1*, PIA-B7 → PTM-G1*).
SACIAB1	NOFF (OFF)	SECONDARY ACIA BSR1 CONTROL - ACIA/PIA wrap-around enable (CB2 → CB1 & CA1, secondary ACIA: TxD → RxD).
SACIAB2	NOFF (OFF)	SECONDARY ACIA BSR2 CONTROL - Wrap-around enable for secondary I/O (CA2 → CB2).
SPIAB1		SECONDARY PIA BSR1 CONTROL - Enable for PIA data wrap-around direction (L: B → A; H: A → B).
SPIAB2	NOFF (OFF)	SECONDARY PIA BSR2 CONTROL - Enable for PIA data port wrap-around.
SLCAB1	NON (ON)	SECONDARY LCA BSR1 CONTROL - Enable for the secondary chip selects (SACIA* and SECROM*).
SLCAB2	NON (ON)	SECONDARY LCA BSR2 CONTROL - Enable for the secondary chip selects (SECRAM*, SPIA*, and SPTM*).

4.4.2.3 Programmable Timer Test Structures

After being certified as fault-free, the PIA can be used to help test its neighboring component--the PTM. Each PTM timer features three control lines (G*, C*, and O), two of which are inputs, and one which is an output (see the "Timer IC" section on page 31).

As indicated by Figure 20, the PIA can control the gate (G*) and clock (C*) inputs for each timer through port B's upper six bits (B2 - B7) and the SCOPE buffer US9. Furthermore, two timer output lines (O2 and O3) can be monitored by two US10 scannable inputs (pins 16 and 15 respectively). Unfortunately, these buffer outputs were not connected to any device inputs.

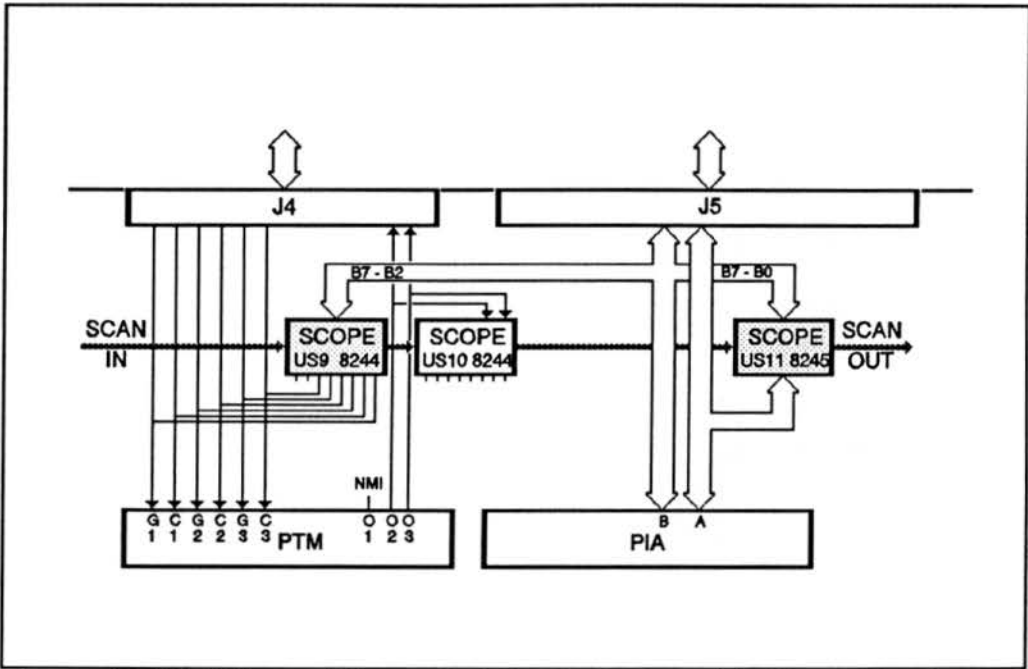


Figure 20. Visibility and Control of the Programmable Timer Signals.

Thus, timer control can be achieved by port B on the PIA or by boundary-scan control of US9, but the second and third timer's output can only be verified by boundary-scan tests. The author regrets any inconvenience that this may have caused the software engineer. However, it was felt that a US10 output-to-peripheral input connection was not possible without additional test component support.

Timer one's output (O1) is used to generate a non-maskable interrupt (NMI*), as such no hardware test support was provided for it because no on-board test components were available to control this wired-OR connection. This signal is tested via software like the

other peripheral interrupt request lines. (See section 4.2.3 entitled *System Area Tests* for more information.)

The PIA and PTM test structures were designed to be non-intrusive, i.e. the test structures could observe and control the peripheral's signals during test operations, yet they would not be relied upon by the peripheral to support any of its functions. This is important, because the test structures are in place to test the system components, not to provide additional support for them. A test component failure should not impact the peripheral's ability to perform any of its functions. In both cases, the peripheral signals could reach the edge connectors without passing through any SCOPE octal buffers.

4.4.3 Test LED

The DFT system contains two LEDs--a green LED to indicate the presence of a 5 volt power supply, and a red LED to indicate primary area test status. Initially this status LED was to mean indicate a "test mode" versus "normal mode" of operation--perhaps via a bi-colored LED. Later, it was determined that this LED could be used to communicate simple test status information prior to the "primary ACIA terminal connection" verification. This became necessary since the ACIA terminal connection is used to display future test status of other system components and it must be certified as fault-free prior to performing these functions.

A primary objective for determining how the LED circuitry would be implemented involved utilizing existing on-board components. The LCA was used to provide firmware controlled status information to the user. However, standard LEDs require more current (about 16 mA are required to light an LED) than the LCA could source so a method of

driving this component needed to be found¹⁰. The buffer also served to protect the LCA (a high dollar component) from thermal stress due to high output currents.

The driver of choice is the high-voltage output open-collector 7407 buffer. However, all six of these buffers were already in use buffering test connector signals and providing wired-OR connections, so an alternative current driver had to be found. The other available on-board buffer--the 74LS125 had two unused gates so it seemed the logical choice. The circuit in Figure 21 shows how an active-low LCA STATUS signal was used to control the test LED circuitry.

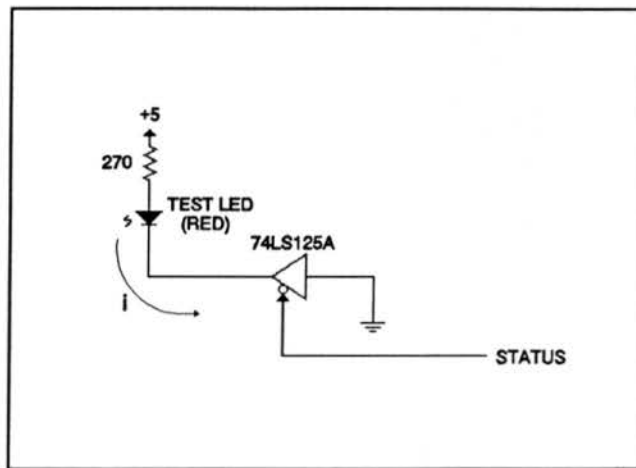


Figure 21. DFT Test LED Connections.

Since the 74LS125A sources only 2.6 mA maximum when high (i.e. $I_{OH} = -2.6$ mA), it became obvious that this buffer could not directly drive an LED [TI 1988]. However, the allowable amount of *sink* current is 24 mA when low (i.e. $I_{OL} = 24$ mA) so the circuit in Figure 21 was designed to sink no more than 16 mA when the 74LS125A output enable

¹⁰ A high efficiency LED requires only around 2 mA.

is asserted¹¹. The simple resistance calculation is shown below in (4). A 270 Ω resistor pack (RPAC) might not be designed to dissipate the amount of heat that this resistor would require ($P = I^2R \approx 70$ mW); so a discrete resistor was used instead.

$$V_{CC} - I_{LED} \cdot R - V_{LED} = 0, \quad R = \frac{V_{CC} - V_{LED}}{I_{LED}}, \quad R = \frac{4.3 \text{ V}}{16 \text{ mA}} = 268.75 \approx 270 \text{ } \Omega$$

where $V_{CC} = 5 \text{ V}$, $V_{LED} = 0.7 \text{ V}$, $I_{LED} = 16 \text{ mA}$ nominal

(4)

Similarly, +5 volt power is indicated by a green LED and resistor circuit that is activated when the microprocessor system receives its +5 volts. This resistance calculation is the same as the one shown above. A power indicator eliminates system power as one of the probable causes of a component or system failure--one has only to look to see if the system is powered.

4.5 Test Software Support

A major test goal of both board designs was to diagnose and isolate failures to the component level so that faulty socketable ICs could be replaced. The test process proceeds in a boot-strapping manner with tested devices aiding in the test of remaining board components. For example, the scannable buffer in Figure 18b is also connected in a controllable fashion to input lines on the PTM. Therefore, the PIA can be used to help test the PTM once the PIA is certified fault-free.

Off-board test software consists of 35 to 50 Kbytes of ASSET C++ routines to exercise boundary-scan testing of board components. Unlike the off-board tests, on-

¹¹ It was assumed that a voltage drop across the 74LS125A would decrease this current even more. However, the 74LS125A design specification of 24 mA should be more than enough to sink the required 16 mA of current.

board tests consist of 12 Kbytes of mostly non-boundary-scan tests coded entirely in assembly language. This software includes isolation routines for the secondary area, and partial isolation for the primary area. The 32K x 8 test ROM still has 12 Kbytes of available space for future enhancements [Lebak 1991].

The limited address space of the MC6809 system as well as the addressing limitations of the DFT design (in terms of compatibility with the existing systems) required that a page-mapped ROM be used in the system area. The 32K x 8 ROM is mapped into eight 4K x 8 pages. One page (with the page-mapping firmware) is left in memory at all times. This page-mapping hardware is contained entirely within the LCA.

Functional testing of all primary area board components (with the exception of the microprocessor) was performed via a dedicated test connector. The primary reason for not testing the microprocessor was due to the lack of a test model and test technique [Lebak 1991]. However, functional testing was performed on all of the secondary area, including the edge connector.

Chapter 5 DFT Costs and Savings

The overall economic study of the benefits and costs of DFT is beyond the scope of this thesis for two reasons, most DFT maintenance benefits will not be seen for some time, and the costs versus savings portion of this project is the responsibility of Mr. Asay. Therefore, this author will not attempt to convey all of the costs associated with DFT--only the costs associated with the design phase. Unfortunately from that perspective, DFT techniques may appear to be costly having little information on the long-term benefits.

Although a DFT design is more complex than a traditional board design, long-term maintenance benefits may outweigh the initial larger capital cost. For example, the DFT design's microprocessor self-diagnoses and isolates faults in the secondary area that would require an expensive and invasive bed-of-nails tester in the case of the traditional design.

Unfortunately, the DFT design represents three times the component cost of a traditional design (fundamentally due to the comparatively high cost of the SCOPE octals and the LCA). This extra cost may be partially justified by examining the effects of the design on the failure rates of the two boards. The additional DFT test components also increase this system's overall failure rate making it greater than that of the traditional design. However, the DFT hardcore area is more reliable than the traditional design since the hardcore area is isolated from the secondary area on the DFT board. The DFT design is also more likely to self-diagnose the cause of any system failures. Finally, any expansion on the DFT design increases the failure probability in the secondary area. Expansion in the traditional design increases failure probability in the hardcore area--which is untestable by the host microprocessor [Assay, Lenhert, and Gordon 1991].

A breakdown of relative test times also indicates the advantages of DFT. Total test time of the DFT system is about 2.3 seconds. This is split between the primary area which takes approximately 1.0 second to test, and the secondary area which takes approximately 1.3 seconds to test. The test time for components that have inherent test capabilities associated with them (e.g. electronic wrap-arounds or 1149.1 connections) is about 0.1 second! This means that 96% of the test time is spent testing those parts which had no test support (i.e. RAM and ROM) [Lebak 1991]!

Based on these test time results and the board area comparisons, a lower DFT cost could have been achieved if a completely arbitrary choice of hardware components was possible. A scannable board constructed from 100% scannable components would have resulted in a reduction in software complexity, hardware component cost, and hardware failure rates by reducing the number of test-dedicated components on the board. For example, if a 32-bit scannable microprocessor like Motorola's 68040 was chosen, four scannable buffers around the microprocessor could have been removed. Furthermore, if the LCA and TBC were scannable, the octal IC count would have been decreased by as much as six--i.e. a reduction of over 10% of the total test area¹². Placing test in the RAM and ROM would have decreased their test time (and the overall test time) greatly. (On a small system this time is insignificant, but on a larger design it could get out of hand.)

Although the DFT system seems to have a high testability component overhead, *only* 7.9% of the component pins are dedicated to perform test functions. This is because the IEEE 1149.1 standard required only serial pin connections.

¹² The Xilinx 4000 series LCAs are IEEE 1149.1 conformable.

The traditional design featured PAL22V10s in place of the DFT system's more complex LCA. In addition, the traditional system used 74LS244s and 74LS245s in place of the BCT8244s and BCT8245s and removed the double buffering on the edge connector. Full address decoding was performed by the PALs although five signals were left unused. For the DFT system, A0 and A1 were not routed to the LCA due to pin limitations so full address decoding was not performed.

Although the DFT system represents 35 components when fully populated (compared to the traditional system's 19 components), power requirements for the two systems differ by less than 1.25 Watts. This can be attributed to the fact that BCT devices were used in place of less power conservative LS devices (see Hardware Reference Manual). It may also be interesting to note that the LCA and its serial PROM required only 110.5 mW--just a fraction of the power required by the PALs (1.89 W)¹³.

Some of the additional costs the author found with respect to the DFT system included a more complex and lengthy PWB layout process. Routing the board became a problem primarily due to the number of different bus types present (partitioned busses). If single buffered address and data lines were available (as is the case in the traditional design), then routing would be much simpler due to the fact that ICs could get to these signals easier if they ran everywhere. In the DFT design, the partitioned signal busses result in less places for ICs to "tie-into" the bus creating real-estate and length minimization problems. Furthermore, additional partitioning logic could result in changing a two-layer board into a multi-layer board--increasing the cost of a DFT system even further.

¹³ There are probably low-power PAL versions available.

Ironically, this routing problem caused more potential manufacturing fault defects since the interconnect density was so high. To successfully route the entire system on a two layer board 10 mil wide traces were necessary. (For more information see the PWB Specification section.) Therefore, to help prevent solder bridging faults and other stuck-at faults, a solder mask was used, which of course increased the cost of fabrication.

Most likely a comparable 9.00" x 6.50" traditional board may not have to use the small traces, vias, and component pad spacing that was used on the DFT system, because there are all less signals to route. Furthermore, the system busses (address, data, and control) are unpartitioned so there are more places that the components can reach these busses.

Chapter 6 Future Work and Lessons Learned

After both the traditional and DFT PWB designs have been fabricated, the testability analysis study will concentrate on comparing statistics obtained from the two designs. Statistics can be used to determine such things as user preference for one of the designs or relative failure rates. In addition, and perhaps most importantly, an economic study of the increased costs and savings of producing chips, boards, and systems with embedded test features is ongoing.

A design goal of no hardware jumpers proved to limit system memory configurations, but provided more reliability and removed areas that would have been untestable. However, hindsight is 20/20, and there are a few areas that need to be addressed.

For example, PTM tests are much more complex than they need to be. Observation of the PTM's timer signals requires the use of scan tests (and thus the TBC), instead of leveraging off of the PIA. These timer output lines could have been wrapped back into pins on the PIA in some fashion, so the more complex (from software) scan tests could have been avoided. Timer control is achieved by either port B on the PIA or by boundary-scan control of US9, but the second and third timer's output can *only* be verified by boundary-scan tests.

The LCA could have been used to generate the serial port clocks since it provides frequency generation support via an external crystal oscillator. Thus, the large 0.600" MC14411 DIP IC could have been removed from the DFT design--thereby reducing component cost and the test-to-core component real-estate. Furthermore, the removal of the A1 from the LCA eliminated the LCA's ability to distinguish between an NMI* or a RESET*.

If the design were started today, there would be a greater selection of scannable devices from which to choose--AT&T's BSM for example instead of TI's TBC. These additional choices would most likely have affected the DFT versus traditional design statistics. In addition, the functional and mechanical requirements affected the choices of components and thus the cost of testing the system.

Chapter 7 Conclusions

One of the primary reasons two designs were constructed was to provide an environment for a testability analysis study. The goal was to determine when an investment in testability provided more cost savings than it consumed. Obviously, increased DFT capital costs can be tolerated less on a small design like the DFT microprocessor system because they take up a large initial investment in area, parts, etc. which cannot be easily amortized over the board (especially in the short term). DFT benefits become more apparent as a product ages (because test and repair time decreases). The electrical engineering department at KSU expects to recover all the capital costs of design for test over the product's lifetime (provided it is long enough).

Unfortunately, at this point, references to design comparisons can only be made, i.e. it may be some time before the benefits of DFT can be realized. However, DFT statistics alone cannot convey the importance of DFT--they seem very biased on a system the size of the DFT system. Instead, direct comparisons between the DFT and traditional designs should be made. These comparisons showed an insignificant difference in power requirements and relatively small test-pin to component pin overhead (7.9%).

In addition, a lower DFT cost could have been achieved if a completely arbitrary choice of hardware components was possible, but the system design was constrained in size, PWB layers, and functionality with a 10-year old design. These constraints were primarily economic constraints. Although they decreased KSU's microprocessor lab costs (by being used after the project is over), they made the DFT system test component statistics decidedly biased.

The testability analysis study and subsequent DFT design represented a unique opportunity to design-in test from the start. Utilizing LCA functions and board level

partitioning facilitated the test process by isolating components not involved in test and subsequently reducing test complexity. I/O capabilities of on-board components were exercised to facilitate testability by reducing the need for additional test components. Although weighing individual functionality in terms of a device's test contribution required a longer design cycle, the reduction in test overhead may well be worth the initially larger design cost. Electronic wrap-arounds provided a large degree of test capability for a small amount of additional hardware.

On-board BIST capabilities are an improvement over traditional testing methods in that the system can remain in the field during the test. This reduces service time for failures in the secondary area since the host processor is the test processor as well. Failures in the primary area are easily diagnosed and isolated through the use of an ANSI/IEEE 1149.1 standard test fixture and TI's ASSET system. This new testable design at Kansas State University permits both internal and external test processing capability while minimizing on-board design overhead.

References and Bibliography

The following alphabetic references and bibliography were invaluable during the course of this project.

References

- [Abramovici, Breuer and Friedman 1990] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, W. H. Freeman and Company, New York, New York, 1990, Chapter 9.
- [Andrews 1982] *Programming Microprocessor Interfaces for Control and Instrumentation*, Prentice Hall, Englewood Cliffs, New Jersey, 1982.
- [Asay, Gordon, and Lenhert 1991] T. Asay, D. Gordon, D. Lenhert, *The Economics of Designing Testability Into An Existing Microprocessor Board*, Proceedings of the International Conference on the Economics of Design For Test, September, 1991.
- [Gordon 1991a] D. Gordon, *Microprocessor Board Design for Built-In Self-Test using SCOPE™ Octals*, Microcomputer Applications, Volume 10, No. 1, pp. 1-4, 1991.
- [Gordon 1991b] D. Gordon, LCA PROGRAMMING SCHEMATICS AND SIMULATION RESULTS, provided to the author by D. Gordon, 1991.
- [HP 1990] *Boundary-Scan Tutorial and BSDL Reference Guide*, Hewlett-Packard, December 1990.
- [IEEE 1990] *Standard Test Access Port and Boundary-Scan Architecture*, IEEE Std. 1149.1-1990, Test Technology Technical Committee of the IEEE Computer Society, Piscataway, New Jersey, May 21, 1990.
- [IEEE 1991] *Supplement to Standard Test Access Port and Boundary-Scan Architecture*, IEEE Std 1149.1-1990, P1149.1a/D5, Test Technology Technical Committee of the IEEE Computer Society, Piscataway, New Jersey, July 14, 1991.
- [Lebak 1991] J. Lebak, *Development of On-Board and Off-Board Tests For a Microprocessor System with Boundary-Scan*, Master's Thesis, Kansas State University, May 1991.
- [Lenhert and Gordon 1990] D. Lenhert and D. Gordon, *EECE 631 Reference Manual*, Engineering Copy Center, Kansas State University, Manhattan, Kansas, August 1990.

- [Motorola 1982] *M68MM17 Monoboard Microcomputer Micromodule 17*, Motorola Semiconductor Products Inc., Phoenix, Arizona, Document M68MM17(D2), June 1982.
- [Motorola 1983] *Motorola 8-bit Microprocessor & Peripheral Data Book*, Motorola Microprocessor Division, Austin, Texas, 1983.
- [Motorola 1989a] *Motorola Telecommunications Data Book*, Series C, Motorola Semiconductor Division, 1989.
- [Motorola 1989b] *Motorola Fast and LS TTL Data Book*, Series G, Motorola Semiconductor Division, 1989.
- [NEC 1991] *Memory Products Data Book*, NEC Electronics, Document No. 60105, 1991.
- [Omaton 1989a] *SCHEMA II+ User's Manual*, Version 2.21, 1989.
- [Omaton 1989b] *SCHEMA-PCB User's Manual*, Version 3.11, 1989.
- [Teradyne 1991] Testability Presentation at Honeywell ATSD, Phoenix, October, 1991.
- [TI 1988] *The TTL Data Book*, Texas Instruments, 1988.
- [TI 1989] *Testability Primer*, Texas Instruments, Richardson, Texas, Document SSYA002, 1989.
- [TI 1991] *Advanced Logic and Bus Interface Logic*, Texas Instruments Semiconductor Group, Richardson, Texas, 1991.
- [Xilinx 1991] *The Programmable Gate Array Data Book*, XILINX, San Jose, California, 1991.

Bibliography

- [AMD 1984] *Programmable Array Logic Data Book*, Advanced Micro Devices, 1984.
- [Horowitz and Hill] *The Art of Electronics*, Horowitz and Hill, Cambridge, 1980.
- [Motorola 1985] *Limited Distance Modem*, Application Note AN946, Motorola, 1985.
- [Paulson and Gordon 1991] M. Paulson and D. Gordon, *A Microprocessor Board Hardware Implementation for Built-In Self-Testability*, Proceedings of the ISMM Conference on Engineering and Industrial Applications of Microcomputers, Long Beach, CA, December 16-18, 1991.
- [TI 1990a] *ASSET: Scan-Based Diagnostics*, Hardware Reference Version 1.0, Texas Instruments, Dallas, Texas, 1990.

[TI 1990b] *ASSET: Scan-Based Diagnostics*, Programmer's Reference Version 1.0, Texas Instruments, Dallas, Texas, 1990.

Appendix A: Nomenclature

1. Signal Name Nomenclature

This section defines the terminology that is used throughout this document to facilitate the user's understanding of both signal and IC test functions. Prefixes and suffixes are used to indicate functionality between systems, signals, and ICs. Prefixes denote the signal's relative system location, while suffixes indicate whether the signal is active-low or active-high.

Signal names are listed in uppercase with a trailing asterisk denoting an active-low signal--e.g., PODHERE* is an active-low signal. The only exceptions, are the bus driver control lines (octal buffers and transceivers) and the test status indicator (STATUS). Since SCOPE octal buffers and transceivers could be interchanged, a transceiver's active-high direction pin would be inappropriately represented by an active-low signal name, so the trailing asterisk was omitted. Furthermore, the method in which the test LED was going to be used was not determined at the time the hardware design was finalized.

For both newly designed systems, the first letter of a signal name depicts that signal's relative board location. For the Design For Testability (DFT) system, this letter specifies what partitioned section of the board contains that particular signal. For example, one can determine by inspection whether a signal is a primary area signal, secondary area signal, edge connector signal, signal from the LCA that is neither a primary or secondary signal, or a microprocessor signal. Figure 21 illustrates these signal prefixes and the naming conventions used by both the DFT and traditional microprocessor systems.

This naming protocol caused a few problems on the DFT system with the LCA sourced ROM and RAM chip selects. In those instances the abbreviations "PRI" and

"SEC" were used to indicate a primary or secondary area signal, respectively. For example, the chip select for the secondary RAM was named SECRAM* instead of the more confusing SRAM*.

Table 16. Signal Name Prefix Descriptions For Both Microprocessor Systems.

Signal Prefix	Traditional Microprocessor System Prefix Description	Design For Testability System Prefix Description
B	Buffered (from/to μ P)	Not Applicable
E	Expansion Edge Connector Signal	Expansion Edge Connector Signal
L	Not Applicable	LCA Only Signal (not a Primary or Secondary Area Signal)
M	Unbuffered Microprocessor Signal	Unbuffered Microprocessor Signal
P	Not Applicable	Primary Area Signal
S	Not Applicable	Secondary Area Signal

DFT system bus control lines use an "afb" syntax, where "a" represents the board area ("P", "S", or "L"), "f" represents the signal's functional name, and "b" represents either Boundary-Scan Cell (BSC) one or two (the control lines for the octals) unless the control line is common to both BSCs--e.g. PADDR. Signal names like SDATA4B2 indicate a secondary area BSC number 2 control signal for a data transceiver whose relative location is number four in the secondary scan chain.

In all cases, an 8-character length signal naming practice was adhered to, so the software designer could use same signal names in the cross-assembler for writing test programs. The whole advantage of a consistent signal naming convention becomes clear when one knows the signal's function, its origin, and where it goes, just by the name.

The test IC's reference designators also indicate their relative board location. A "Uax" naming convention was adopted, where "U" still indicates an IC, "a" indicates the

relative board area ("P"rimary, or "S"econdary), and "X" indicates that components relative scan location. For example, US8 is the eighth component in the secondary scan chain. This method of illustrating component scan chain locations was chosen over the use of a silk screen for two reasons. First, the components are not placed on the PWB in a scan chain order. Instead, they are placed in locations that minimizes their interconnect signal lengths. Second, a consistent naming convention would facilitate component descriptions and enable Lebak to use these names in his test software routines.

References to the IEEE 1149.1 standard are presented using the same as notation as the standard. Signal names are listed in uppercase; register names use a two or three letter acronym (as defined by the standard); TAP controller states are indicated by italicized mixed-case text; and finally, instructions are shown in italicized uppercase text.

LCA dual function pins were labeled "I/O", unless the dual function capabilities of that signal were used. For example, the LDC pin is labeled LDC-I/O since the LDC signal is used during LCA configuration, and the I/O portion is used after configuration.

To facilitate schematic readability, signals leaving one sheet and entering another are given an exiting grid location in the one case, and an entering location in the other. For example, In all cases, the signals enter and exit on the sheet edges, i.e. column 1 or 8. Three or more signals with common functions were grouped into busses so as not to "clutter" up the pages.

Since additional work will have to be done to produce the traditional system PWB artwork, a standard file naming convention was adopted to distinguish between DFT and traditional design files. The first three characters of a file's name indicates whether it is a Design For Testability (DFT) or Traditional Design (TRD) system file. For example, all of the DFT schematics are named DFTBOARD.P0x (where x is one of five schematic

pages, i.e. the numbers from 1 to 5) [Omaton 1989a]. PWB files are denoted by a ".JOB" file extension [Omaton 1989b].

2. Glossary of Terms/Acronyms

1149.1: See IEEE 1149.1.

Assert(ed): To place a signal in its active state. For example, in a positive logic system an active-low signal is asserted by a logic zero value.

ASSET (Advanced System Support for Emulation and Test): A boundary-scan test development system upon which users can write, develop, and debug boundary-scan tests. ASSET is available from Texas Instruments, and runs on an IBM-PC under DOS. ASSET was used by Lebak to develop and run the off-board tests.

Auxiliary Board: One of two boards that are electrically connected together via a card cage backplane to form KSU's current microprocessor system. This board is a KSU-designed wire-wrapped board that consists of an additional PIA and buffered expansion edge connector signals taken from the backplane.

BIST (Built-In Self-Test): The ability of a system or component to test itself.

Controllability: The ability to establish a specific signal value at a node in a circuit.

DFT (Design For Testability): Design practices that are intended to ensure that a component or system is testable, i.e. test structures are "designed in" from the beginning.

DFT Design: A microprocessor system that incorporates both structured and Ad hoc DFT practices to provide as much signal visibility and control as possible (given obvious economic constraints).

IEEE 1149.1: This standard defines a test architecture that can be incorporated into digital devices to simplify the test process by providing a means of controlling and observing values at the device's pins.

JTAG (Joint Test Action Group): A consortium of electronic companies that originally conceived the now standard IEEE 1149.1 digital test bus architecture.

LCA (Logic Cell Array): An FPGA from Xilinx, that was used in the DFT system to provide a myriad of functions. These functions include performing peripheral and memory chip select decoding, clock frequency division, controlling bus driver enables, and supporting on-board test operations.

LCA Firmware: Programmable logic in the LCA that is configured via software. This software provides support for on-board test functions and the general operation of the microprocessor system. (See LCA above.)

Main Board: One of two boards that are electrically connected together via a card cage backplane to form KSU's current microprocessor system. This board is a PWB

that houses the microprocessor and most of the system peripherals. (See MM17 or MM19 below.)

MM17 or MM19: The main microprocessor board in KSU's current microprocessor system used in their EECE 631 microprocessor design course. Both the DFT and traditional systems maintain functional compatibility with KSU's system by incorporating the same core components.

Negate(d): To place a signal in its inactive state. For example, in a positive logic system an active-low signal is negated by a logic high value.

NOFF (Normally Off): A buffer that is three-stated during normal system operation to avoid signal contention problems. Under certain test conditions the buffer may drive a signal, i.e. it may be turned on.

NON (Normally On): A buffer that drives a signal during normal system operation. Under certain test conditions the buffer may be three-stated to avoid signal contention problems, i.e. it may be turned off.

Observability: The ability to determine a specific signal value at a node in a circuit.

PAL (Programmable Array Logic): A fuse programmable device that was incorporated into the traditional design to perform the chip selects and control bus driver operations.

Predictability: The ability to obtain a known output response given an input stimulus.

Primary Area: One of the partitioned DFT system areas. Host microprocessor self-tests could not be performed on this area because this area comprises the minimum set of system components that are essential to proper system operation.

PWB (Printed Wiring Board): The printed circuit board that contains the artwork for all of the DFT system signals on a single two-layer (component and non-component side) through-hole board.

ROM Firmware: Software routines stored in the primary (DFT system) or system ROM (Traditional system). This software includes the system monitor program (SUPERmon), the KSU mass storage program (Mother Hen), and in the case of the DFT system--the on-board test routines and test-support software.

SCOPE (System Controllability/Observability Partitioning Environment): A family of ICs from Texas Instruments that are not only IEEE 1149.1 compliant, but implement additional test instructions not found in the standard. A total of 19 SCOPE octal buffers, transceivers, and edge-triggered D flip-flops were incorporated into the DFT system design.

Secondary Area: One of the partitioned DFT system areas. This area can be tested by the host microprocessor. Components found in this area are not essential to the operation of the system.

Stable State: A reference to one of six states the TAP controller can be placed in as defined by the IEEE 1149.1 standard. When the TAP controller remains in the same state for two or more TCKs, that state is said to be stable.

TAP (Test Access Port): The four, or optionally five-wire serial test bus as defined by the IEEE 1149.1 standard.

TAP Controller: The 16-state state machine that controls boundary-scan test functions via the IEEE 1149.1 serial test bus. The TAP controller changes states on the rising edge of TCK based on the value of TMS (IEEE 1149.1 test bus signals).

TBC (Test Bus Controller): Texas Instrument's boundary-scan controller IC. This device was incorporated into the DFT system to provide on-board host microprocessor controlled test operations.

Temporary State: A reference to one of ten states the TAP controller can be placed in as defined by the IEEE 1149.1 standard. When the TAP controller cannot remain in the same state for two or more TCKs without first passing through another state, that state is said to be temporary.

Testability: A design characteristic that influences costs associated with testing.

Traditional Design (TRD): A microprocessor system that incorporates no inherent testability features. This system was used as a baseline microprocessor system design from which DFT system design results could be compared.

Visibility: See Observability.

Appendix B: EECE 631 Hardware Reference Manual

This reference manual describes the hardware characteristics of both the Design-For Testability (DFT) System and the traditional microprocessor systems in place at KSU. KSU uses these systems in an upper-level undergraduate and graduate-student microprocessor design class (EECE 631) to investigate microprocessor operation. They are used as a platform from which student hardware and software projects can be constructed.

The traditional system block diagram can be found in Figure 22. It depicts a typical microprocessor design with one set of unpartitioned busses. Standard low-power schottky TTL bus drivers control the bus signals, while Programmable Array Logic (PAL) devices are used to perform chip select decoding and control these bus driver octals. This system represents a design based on the traditional microprocessor system design approach which incorporates no on-board testability structures that could help the user diagnose faulty components.

Although, these systems are functionally equivalent (with respect to components, memory maps, and ROM-based software routines), the DFT system can diagnose component or system faults. It incorporates electronic wrap-around buffers, the IEEE 1149.1 test bus, partitioned busses, and firmware-based test control. Therefore, emphasis will be placed on the description of this system, since it is functionally more complex and can potentially help the user debug projects better. However, both system's features, components, and capabilities are described as well as important user information such as the system memory maps and the edge connector pinouts. The DFT system block diagram is shown in Figure 23.

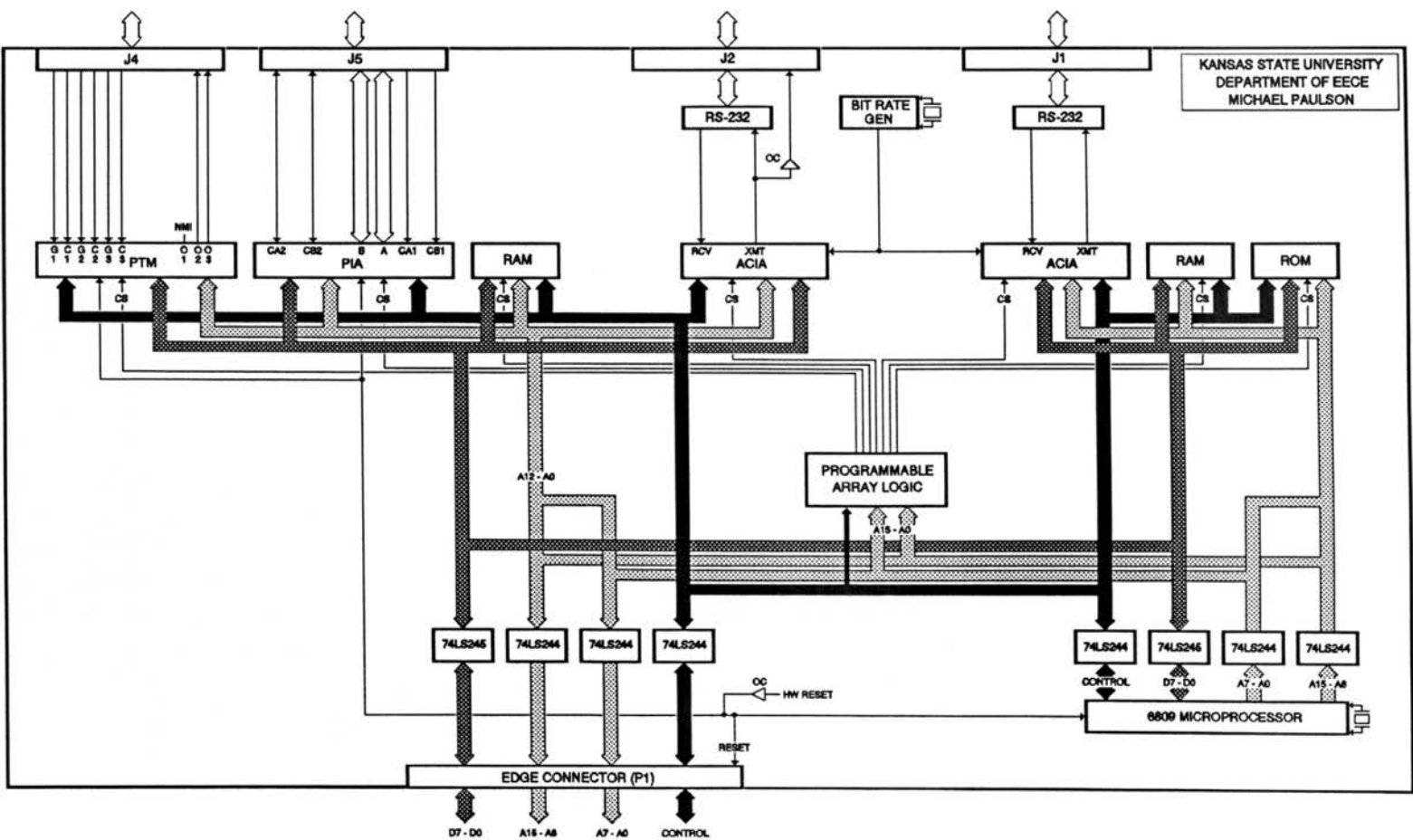


Figure 22. Traditional Microprocessor System Block Diagram.

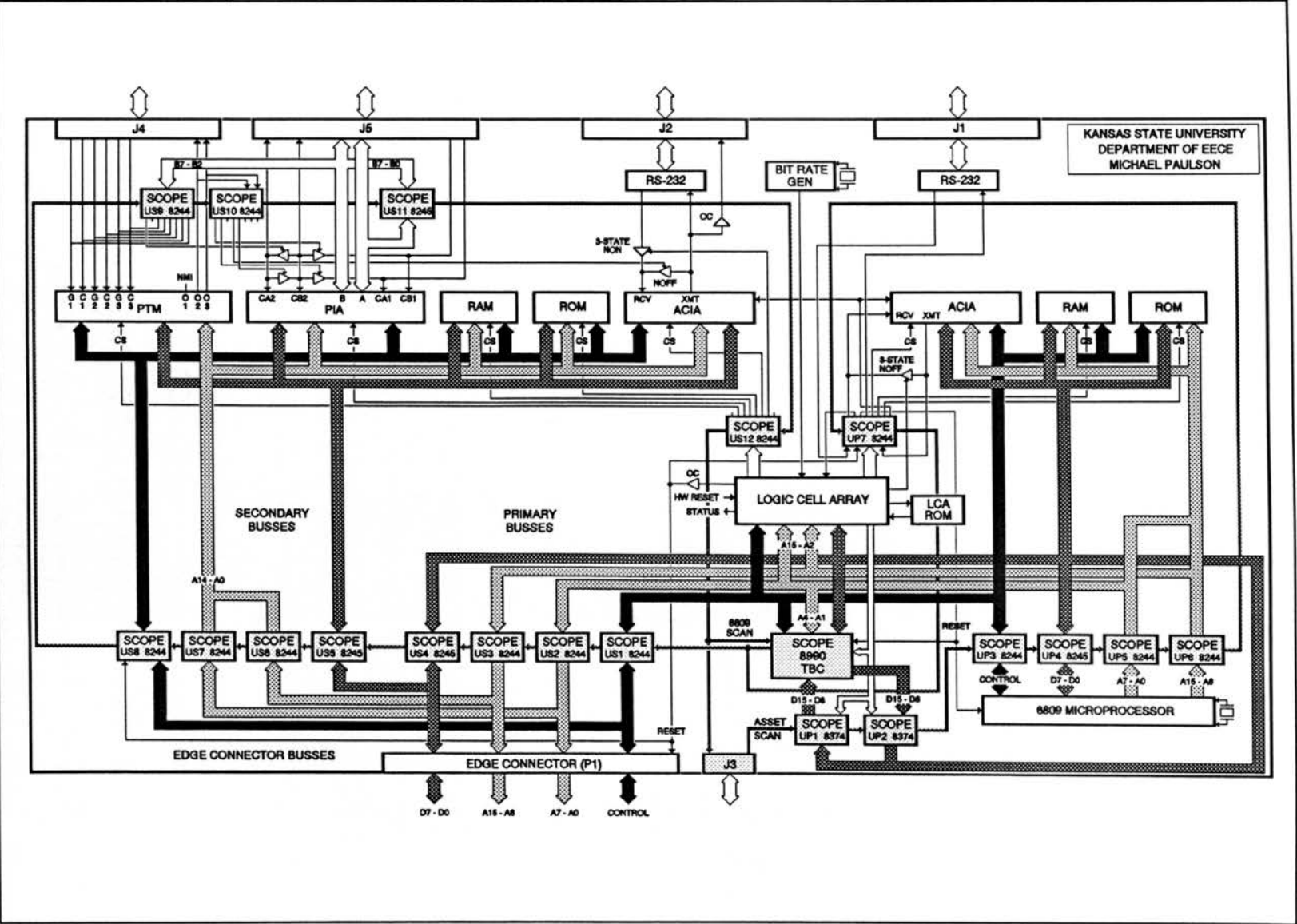


Figure 23. DFT Microprocessor System Block Diagram.

Each microprocessor system comprises two microprocessor boards that are electrically connected to one another via expansion edge connectors and a card cage backplane. The PWB that houses the microprocessor and most of the peripherals will be referred to as the main board, and the other board will be referred to as the auxiliary board.

The main microprocessor board from each system consists of a Motorola MC68B09 microprocessor, 8-bit addressable RAM and ROM, and components from Motorola's 6800 family of peripherals with no inherent testability features. This list of peripherals includes Motorola's Peripheral Interface Adapter (PIA) or MC68B21 which provides two 8-bit bidirectional parallel ports, and a MC68B50 or Asynchronous Communications Interface Adapter (ACIA) to provide serial I/O capabilities such as the terminal interface for input/output messages and the serial communication link to permanent mass storage for user programs and projects. Timed operations are controlled by the one of three programmable timers on the MC6840 or Programmable Timer Module (PTM).

The auxiliary board is a wire-wrapped board that provides the buffered expansion edge connector signals (taken directly from the card cage backplane), at the front of the card cage. Since all of the laboratory systems are placed in a card cage system, the expansion edge connector signals would be unavailable if the auxiliary board did not provide them. This gives the user access to all system interface signals.

Often projects and subsequent debugging sessions lead to component or system failures on the host microprocessor system. The traditional system has extremely limited self-test capability so students are unable to ascertain the origin of any system errors--their project or the lab equipment. By monitoring the design process as well as the use

of the microcomputers, a microprocessor system with Built-In Self-Testability (BIST) could provide an excellent environment for a testability analysis study.

This reference manual gives a brief introduction into DFT system test components and structures. A basic understanding of the *IEEE Std 1149.1* digital test standard (for more information consult the standard) was assumed. For information on the DFT versus traditional design laboratory procedures or the operation of the test routines consult Appendix B of Lebak's thesis, i.e. the *Student User's Manual for the Diagnostic Routines* [Lebak 1991].

1. DFT System Hardware and Software Components

BIST components include Texas Instrument's (TI) IEEE 1149.1 conformable Scan for Controllability Observability Partitioning Environment (SCOPE) octals, TI's boundary-scan Test Bus Controller (TBC part number SN74ACT8990), and an external boundary-scan test access port. This test port provided an external connection to an IEEE 1149.1 test bus via a dedicated edge connector. This connector, along with a software development system from TI known as the Advanced System Support For Emulation and Test (ASSET) may be used to control test operations via the scannable buffers on the board. ASSET provides the necessary IEEE 1149.1 vectors to control these scannable devices.

2. Test Control via LCA

To reduce hardware real-estate and provide a consistent approach to test control, a Field Programmable Gate Array (FPGA) was chosen to facilitate the test process. The FPGA chosen was a product from XILINX (3000 series) that programmed itself upon power-up via a serial ROM. This reprogram feature was used to effect a complete hardware reset of the entire system via a pushbutton reset switch and appropriate reset circuitry.

The LCA provided many advantages such as configuration registers in place of hardware jumpers. This eliminated the need to both move and remember where hardware jumpers were placed. Hardware jumpers were effectively replaced by software jumpers and all default settings can be reloaded on power-up via the LCA reprogramming cycle. In addition, certain signals may be controlled and/or observed through registers designed within the LCA. These include (but are not limited to) the enables for the buffers as well as the baud rate clock for two serial ports. The LCA also facilitated communication between TI's 16-bit TBC and Motorola's 8-bit microprocessor by serving as an interface to scannable flip-flops that controlled the upper and lower eight bits of the TBC's data bus.

At power-on all control signals have deterministic conditions. Buffers necessary for primary operation are enabled. Those buffers that are not necessary for primary operation and have the possibility of interfering with proper microprocessor operation are disabled. These buffers are later enabled via LCA firmware.

3. Board Partitioning

To facilitate testability, the microprocessor board was partitioned into two sections: a primary area (sometimes called the system or hardcore area) where self-testability could not be reliably achieved, and a secondary area on which microprocessor-controlled self-test could be performed [Gordon 1991]. The partitioning of the board was performed via TI's SCOPE octals that are connected in a boundary-scan path that permits component isolation and testing. In addition, they allow controllability and observability of the RAM, ROM, and peripherals.

Before testing the secondary area, the primary area must be certified as fault-free. The primary area includes the following: the microprocessor; a primary RAM and ROM; a Logic Cell Array (LCA) which provides chip select circuitry, facilitates host microprocessor testing control, and controls boundary-scan device buffer enables; a boundary-scan Test Bus Controller (TBC); and an Asynchronous Communications Adapter (ACIA) that permits serial I/O capabilities as well as off-board observability. (This ACIA informs the operator as to the results of the on-board tests.)

The buffer configuration for the primary area allows complete isolation of both the microprocessor and edge connector by electrically isolating the primary area from the secondary area while power-on self-tests are performed. This keeps failures in the secondary area from interfering with the primary area. Buffers reaching out from the primary area may be selectively enabled once their signals have been tested.

Once the primary area is certified as fault-free, the secondary area is tested using a combination of both boundary and non-boundary-scan test procedures [Lebak 1991]. The secondary area consists of a secondary RAM, ROM, and ACIA and all other peripherals such as the Programmable Timer Module (PTM), and a Peripheral Interface Adapter (PIA) that permits parallel I/O capabilities via two 8-bit bidirectional ports. The

on-board microprocessor controls all secondary test operations via the boundary-scan test bus controller or by the use of peripheral I/O wraparounds. A boundary-scan test connector provided an interface to TI's ASSET system that permitted diagnosis of faults in both the primary and secondary areas.

4. DFT Structures

The IEEE 1149.1 test bus was under host microprocessor control by utilizing TI's TBC as a primary area peripheral. TI's TBC controls all major boundary-scan functions such as the scan and execution instructions required of all IEEE 1149.1 conformable devices (TI's SCOPE octals). Host microprocessor control of the test bus allowed operational verification of secondary board areas such as the edge connector. Since the microprocessor boards (user and system) are all connected together in a card cage, they all contain card edge connectors [Motorola 1982]. Functional compatibility with the existing microprocessor system was a design requirement, so this edge connector was incorporated into the design. Due to the high failure rate of exposed edge connectors, it was deemed necessary to place isolating SCOPE buffers around the busses to and from this connector. Isolating buffers helped partition the board design by effectively separating the busses into primary and secondary areas (see Figure 8). This permitted full observation and control of the data, address, and control busses.

An edge connector dedicated to the scan testing of board interconnects provides access for off-board testing via ASSET and the boundary-scan chain. When the ASSET system connector is present, the primary and secondary scan chains that serve as partitioning logic are automatically linked into one long scan chain. This is done by utilizing a TOFF* input signal on the TBC that three-states all the IC's outputs. This

eliminated bus contention problems between the ASSET system and the on-board TBC [TI 1991]. This long scan chain allows us to supplement the testing of the primary-secondary interface (in the event that enhancement is needed).

In addition to boundary-scan test features, non-boundary-scan test features are incorporated utilizing existing on-board components in the test process. This reduced system test overhead. Traditional testing of I/O channels requires the use of mechanical wrap-arounds. In the DFT design, three-stateable electronic wrap-arounds connect controllable peripheral outputs to observable peripheral inputs under firmware control (at speed). One limitation of this technique is that an external device connected to these pins might drive the signal with a lower impedance than our wrap-around buffers. This would lead to an erroneous diagnosis of the system. To address this difficulty the test software contains prompts for the user to remove all external device connections in the event that this situation occurs.

Figure 24a illustrates such a wraparound technique that can be applied in the case of unidirectional signals. The serial capabilities of the ACIA can easily be monitored by employing a normally on (NON) SCOPE octal buffer and a normally off (NOFF) three-stateable buffer to electronically wrap an output around to an input. Parallel I/O in the form of Motorola's PIA can also be used in this fashion to test neighboring components or itself by employing a similar technique via three-stateable buffer packages (see Figure 24b). Ports A and B are bidirectional so they can each be configured to read and write at separate times to perform a self-diagnosis. Furthermore, SCOPE octal buffers are utilized, permitting boundary-scan test techniques and providing a moderate amount of self-check on the electronic wrap-around buffers.

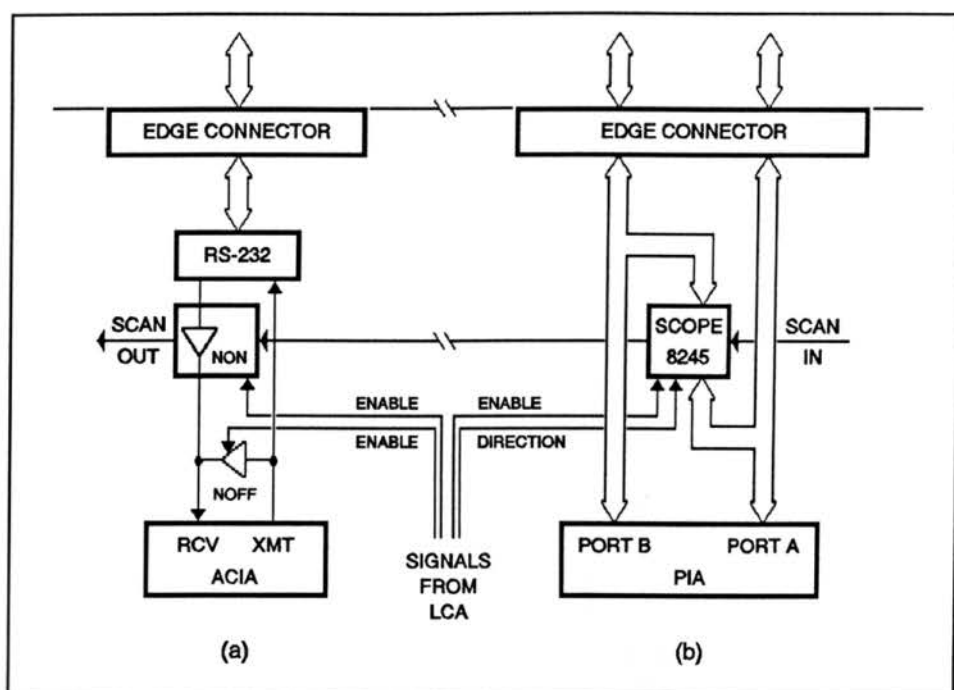


Figure 24. Uni- and Bi-directional Signal Wrap-Around Techniques.

On-board testing proceeds in a bottom-up approach. One of the first peripherals to be tested is the primary ACIA since it is used to help display error messages. Test status for this device is indicated by a test LED since normal diagnostic messages can not be sent to a terminal if the ACIA is dead.

5. Test Software Support

A major test goal of both board designs was to diagnose and isolate failures to the component level so that faulty socketable ICs could be replaced. The test process proceeds in a boot-strapping manner with tested devices aiding in the test of remaining board components. For example, the scannable buffer in Figure 24b is also connected in a controllable fashion to input lines on the PTM. Therefore, the PIA can be used to help test the PTM once the PIA is certified fault-free.

DFT and Traditional system features

- ▶ an advanced Motorola MC68B09 8-bit microprocessor:
 - Two accumulators configurable as either separate 8-bit registers or one 16-bit accumulator that provides support for:
 - 16-bit ALU-functions such as addition and subtraction, and operations like compare, load, and store.
 - 8 x 8-bit unsigned multiply
 - Two 16-bit stack pointers and index registers
 -
- ▶ Two separate 8-bit parallel I/O ports with four dedicated handshake lines.
- ▶ An asynchronous serial (RS-232 compatible) port with a strap-controlled terminal interface and a firmware controlled baud rate from 153.6 kHz to 9600 Hz on the DFT system, and 9600 baud on the traditional system.
- ▶ A serial port dedicated to communication with KSU's PDP-11 computer for mass storage of user programs and data.
- ▶ A programmable timer module with three separate 16-bit programmable counters or timers.
- ▶ Four sockets on the DFT design are devoted to memory. Two are for +5-volt-supplied 32K x 8 SRAM's, and two are for 32K x 8 EPROM's or other 28-pin JEDEC conformable Non-Volatile Memory (NVM). The traditional design incorporates one less ROM socket, the remaining three other sockets are for 8K x 8 sized memory.
- ▶ An 8 MHz crystal-controlled clock is internally divided by four in the microprocessor to produce two bi-phase system clocks (E and Q) that run at 2 MHz.
- ▶ A green power-on LED to indicate system power.

DFT-specific features

- ▶ A red test LED to indicate system self-test status prior to terminal verification, or to indicate the presence of the ASSET connector.
- ▶ A firmware-controlled Baud Rate Generator (BRG) for the user serial I/O port (see LCA below) on the DFT board.

- ▶ Fully buffered and partitioned address, data and control busses to facilitate the detection and isolation of system failures.
- ▶ Two serial boundary-scan test chains for the detection and isolation of system faults via Texas Instrument's (TI) IEEE 1149.1 conformable SCOPE octals.
- ▶ A test interface dedicated to providing communication for off-board boundary-scan testing of system components via TI's ASSET System.
- ▶ An on-board Test Bus Controller (TBC) that generates on-board boundary-scan vectors for testing the secondary system area.
- ▶ A Logic Cell Array (LCA) that contains firmware-controlled:
 - peripheral chip select decoding logic (programmable memory mapped I/O)
 - clock division circuitry for:
 - the Baud Rate Generator
 - TBC's Scan Clock
 - test registers that provide:
 - SCOPE octal output enable control
 - an on-board boundary-scan control interface between the microprocessor and the TBC
 - additional test features [Lebak 1991].
- ▶ Pushbutton hardware reset circuitry that initiates a full system reset by reconfiguring the system LCA and beginning a complete system self-test.

Table 17. Traditional and DFT System Specs. Adapted from [Motorola 1982].

Microprocessor System Characteristics	Specifications	
	Traditional	DFT
Microprocessor	MC68B09	
<u>Power Requirements</u> (without memory)	10.27 W +5 V & ± 12 V supply	11.5 W +5 V & ± 12 V supply
<u>Word Sizes of Signal Types</u>		
Data	8-bits, three-state TTL-compatible signals.	
Address	16-bits, three-state, TTL-compatible signals.	
Instruction	59 variable length instructions (up to five bytes in length)	
Addressing Modes	10 Address modes are available. They include: direct, relative, immediate, indexed, extended, register, inherent, extended indirect, indexed indirect, and long relative.	
Clock Frequency	8 MHz crystal controlled oscillator that is frequency divided by the μP to give a system clock of 2 MHz.	
<u>Memory</u>		
Address Range (with translated memory)	64K	> 64K (via bank selection)
Memory Types	8K x 8 SRAM and EPROM.	32K x 8 SRAM and EPROM
I/O Addressing	One PIA & PTM, & two ACIAs.	One PIA & PTM, & two ACIAs.
External Memory	At least one 44K block.	At least one 44K block.
<u>Interrupts</u>		
Internal	NMI* and IRQ* from the PTM. All other on-board peripherals can only request an ISR.	
External	IRQ*, FIRQ*, and NMI* are available at the expansion connector (P1).	
Operating Temperature	Commercial temperature range (component dependent).	

Microprocessor System Characteristics	Specifications	
	Traditional	DFT
<u>Physical Characteristics</u>		
Width x Depth	9.75" x 6.00"	
Thickness	Two layer PWB.	
<u>Interfaces</u>		
Expansion Edge Connector P1	Stanford Applied Engineering SAC-43D/1-2.	
Serial I/O Interface Connector J1	20-pin 3M type.	
Serial I/O Mass Storage Connector J2	20-pin 3M type.	
Test Interface Connector J3	Not available	14-pin 3M type.
Programmable Timer Interface Connector J4	20-pin 3M type.	
Parallel I/O Interface Connector J5	26-pin 3M type.	

Table 18. System Memory Maps. Adapted from [Lenhert and Gordon 1990].

Address (Hex)	Current / Traditional System Functions	Design-For-Testability System Function
0000-1FFF	User RAM	Secondary RAM
2000-CFFF	Unused	Unused
D000-D3FF	Mother Hen Program	Mother Hen Program
D400-D4BF	Memory Test Program	Memory Test Program
D4C0-D4DF	Memory Move Program	Memory Move Program
D4E0-D4FF	Memory Clear Program	Memory Clear Program
D500-D5FF	ACIA Test Program	ACIA Test Program
D600-D7FF	PIA Test Program	PIA Test Program
D800-DFFF	SUPERmon ROM	SUPERmon ROM
E000-E7FF	System RAM	Primary RAM
E800-EBFF	User I/O Area	User I/O Area
EC00-EC0F	Unused	Unused
EC10-EC13	System PIA* / Not Available*	Not Available*
EC14-EC15	System ACIA (Console)	Primary ACIA (Console)
EC16-EC17	Unused	Unused
EC18-EC1F	System PTM	Secondary PTM
EC20-EC53	Unused	Unused
EC54-EC55	System Aux. ACIA* / Not Available*	Not Available*
EC56-EC7F	Unused	Unused
EC80-ECA7	Not Available*	Primary TBC*
ECA8-ECFF	Unused	Unused
ED00-ED1F	Not Available*	Used by LCA*
ED20-EDBF	Unused	Unused
EDC0-EDC3	User PIA	Secondary PIA
EDC4-EEBF	Unused	Unused
EEC0-EEC1	Mother Hen (MH) ACIA	Secondary ACIA (MH)
EEC2-EEFF	Unused	Unused
EF00-EFFF	User I/O Area	User I/O Area
F000-FFFF	SUPERmon ROM	SUPERmon ROM

* denotes system differences.

Table 19. DFT System Connector Descriptions.

Reference Designator	# Pins	Description
J1	20	Primary Serial Interface (Terminal ACIA)
J2	20	Secondary Serial Interface (Mother Hen ACIA)
J3	14	Test Interface
J4	20	Programmable Timer Interface (PTM)
J5	26	Parallel Interface (PIA)
P1	86	Expansion Edge Connector

Table 20. System Connector Specifications.

Reference Designator	# Pins	Key Location	Pin Width	Ctr-to-Ctr Spacing
J1	20	pin 3/pin 5	0.050"	0.100"
J2	20	pin 5/pin 7	0.050"	0.100"
J3	14	pin 3/pin 5	0.050"	0.100"
J4	20	pin 7/pin 9	0.050"	0.100"
J5	26	pin 7/pin 9	0.050"	0.100"
P1	86	pin 19/pin 21	0.100"	0.156"

The following section shows the interface signals and their edge connector pin locations. The pin locations on these edge connectors are such that existing cables may be reused with the addition of the appropriate card edge connector. Information in Table 21 describes the dedicated serial terminal connection J1. The dedicated Mother Hen serial communication connector J2 is described in Table 22. The DFT system's

dedicated test connector signals (J3) are depicted in Table 23. Since this test connector is new, a column in Table 23 indicates which ASSET cable signals should be used in the fabrication of the off-board test cable. Information in Table 24 describes the PTM connector signals available to the user (J4), and the PIA interface signals on connector J5 are shown in Table 25. Finally, the expansion edge connector signal table can be found in Table 26.

The pin definitions for the LCA (DFT system) and PALs (traditional system) are also shown in this section as well. The LCA signal table is shown in a single four-page table--Table 27, while the PALs U15 and U10 are depicted in Table 28 and Table 29. Since full address decoding requires all 16 address lines, PAL U10 multiplexes the upper address byte from 8 → 3 lines and sends this information to the lower-address PAL U15. U15 now has enough information to fully decode all of the address lines so it is the PAL that sources the chip selects.

Table 21. Connector J1 Serial Interface Signals.

Connector Pin #	Signal Mnemonic	I/O/B	Signal Name and Description
1,4,8,9,10, 11,12,14, 15,16,17, 18,19,20	NC		NO CONNECTION, i.e. not used.
2	STRAPPOS	I	STRAPPABLE TO +12 V - NC or strappable to +12 V via a J6 solder connection.
3	PTXD12V	O	PRIMARY ACIA TRANSMIT DATA - Serial data transmit line.
5	PRXD12V	I	PRIMARY ACIA RECEIVE DATA - Serial data receive line.
6	STRAP5V	I	STRAPPABLE TO +5 V - NC or strappable to +5 V via a J6 solder connection.
7	PRTS12V		PRIMARY ACIA REQUEST TO SEND - Terminal Requests permission to transmit data line.
10	STRAPNEG	I	STRAPPABLE TO -12 V - NC or strappable to -12 V via a J6 solder connection.
13	GND	I	GROUND

Table 22. Connector J2 Serial Interface Signals.

Connector Pin #	Signal Mnemonic	I/O/B	Signal Name and Description
1,13	GND	I	GROUND
2,4,6,7,8,9, 10,11,12,14, 15,16,17,18, 20	NC		NO CONNECTION, i.e. not used
3	SRXD12V	I	SECONDARY ACIA RECEIVE DATA - Serial data receive line.
5	STXD12V	O	SECONDARY ACIA TRANSMIT DATA - Serial data transmit line.
19	MHTXD	O	MOTHER HEN TRANSMIT - An OC wire-ORed serial communication link for program storage.

Table 23. Connector J3 Test Interface Signals (DFT System Only).

Connector Pin #	ASSET Pod Pin #	Signal Mnemonic	I/O/B	Signal Name and Description
All even pins except pin 2	31	GND	I	GROUND - Connects ASSET ground and the microprocessor system ground so both are at the same voltage potential.
1	24	TMSR	O	TEST MODE SELECT RETURN - Maintains synchronization of test bus signals.
2	33	PODHERE*	I	ASSET POD CONNECTED - A ground line from ASSET is used to pull this signal down to indicate an ASSET pod connection.
3	26	TMS1	I	TEST MODE SELECT 1 - Broadcast secondary board area TMS line. Used to select TAP controller states.
5	32	UUTCLK	O	UNIT UNDER TEST CLK - Same function as J3-1.
7	34	LCLK	I	LOCAL CLOCK - Broadcast test clock (TCK) signal to all SCOPE octals.
9	30	TDO	I	TEST DATA OUTPUT - Used to send ASSET serial test instructions or data.
11	26	TMS0	I	TEST MODE SELECT 0 - Broadcast primary board area TMS line. Used to select TAP controller states.
13	28	TDI	O	TEST DATA INPUT - Used to receive serial information from the SCOPE octals.

Table 24. Connector J4 Timer Interface Signals.

Connector Pin #	Signal Mnemonic	I/O/B (Active)	Signal Name and Description
All even pins plus pin 19	GND	I	GROUND
1	G1*	I (L)	GATE INPUT 1 - Used as a trigger or clock gating to the Timer.
3	O1 (NMI*)	O (H)	TIMER OUTPUT 1 - PTM signal capable of driving up to two TTL loads.
5	C1*	I (L)	CLOCK INPUT 1 - Used to decrement the timer.
7	G2*	I (L)	GATE INPUT 2 - Same function as G1* on pin 1.
9	O2	O (H)	TIMER OUTPUT 2 - Same function as O1 on pin 3.
11	C2*	I (L)	CLOCK INPUT 2 - Same function as C1 on pin 5.
13	G3*	I (L)	GATE INPUT 3 - Same function as G1* on pin 1.
15	O3	O (H)	TIMER OUTPUT 3 - Same function as O1 on pin 3.
17	C3*	I (L)	CLOCK INPUT 3 - Same function as C3 on pin 5.

Table 25. Connector J5 Parallel Interface Signals.

Parallel Interface Pin Numbers, Names and Functions			
Connector Pin #	Signal Mnemonic	I/O/B	Signal Name and Description
1	A6	B	PORT A (Bit 6) - Data line from PA6 on the PIA. Bit 6 of Data Direction Register A (DDRA) controls whether this line will be an input (0) or an output (1).
2,4	+5	I	+5 V DC power supply.
3	A7	B	PORT A (Bit 7) - Same function as A6 on pin 1 except signal emanates from PA7, and is controlled by bit 7 of the DDRA.
5	B0	B	PORT B (Bit 0) - Data line from PB0 on the PIA. Bit 0 of Data Direction Register B (DDRB) controls whether this line will be an input (0) or an output (1).
6	CA1	I	CONTROL BIT 1, PORT A - First control bit of Port A.
7	B1	B	PORT B (Bit 1) - Same function as B0 on pin 5 except signal emanates from PB1, and is controlled by bit 1 of the DDRB.
8	CA2	B	CONTROL BIT 2, PORT A - Second control bit of Port A.
9	B2	B	PORT B (Bit 2) - Same function as B0 on pin 5 except signal emanates from PB2, and is controlled by bit 2 of the DDRB.
10	A0	B	PORT A (Bit 0) - Same function as A6 on pin 1 except signal emanates from PA0, and is controlled by bit 0 of the DDRA.
11	B3	B	PORT B (Bit 3) - Same function as B0 on pin 5 except signal emanates from PB3, and is controlled by bit 3 of the DDRB.
12	A1	B	PORT A (Bit 1) - Same function as A6 on pin 1 except signal emanates from PA1, and is controlled by bit 1 of the DDRA.

Parallel Interface Pin Numbers, Names and Functions			
Connector Pin #	Signal Mnemonic	I/O/B	Signal Name and Description
13	B4	B	PORT B (Bit 4) - Same function as B0 on pin 5 except signal emanates from PB4, and is controlled by bit 4 of the DDRB.
14	A2	B	PORT A (Bit 2) - Same function as A6 on pin 1 except signal emanates from PA2, and is controlled by bit 2 of the DDRA.
15	B5	B	PORT B (Bit 5) - Same function as B0 on pin 5 except signal emanates from PB5, and is controlled by bit 5 of the DDRB.
16	A3	B	PORT A (Bit 3) - Same function as A6 on pin 1 except signal emanates from PA3, and is controlled by bit 3 of the DDRA.
17	B6	B	PORT B (Bit 6) - Same function as B0 on pin 5 except signal emanates from PB6, and is controlled by bit 6 of the DDRB.
18	A4	B	PORT A (Bit 4) - Same function as A6 on pin 1 except signal emanates from PA4, and is controlled by bit 4 of the DDRA.
19	B7	B	PORT B (Bit 7) - Same function as B0 on pin 5 except signal emanates from PB7, and is controlled by bit 7 of the DDRB.
20	A5	B	PORT A (Bit 5) - Same function as A6 on pin 1 except signal emanates from PA5, and is controlled by bit 5 of the DDRA.
21	CB1	I	CONTROL BIT 1, PORT B - First control bit of Port B.
22,25	GND	I	GROUND
23	CB2	B	CONTROL BIT 2, PORT B - Second control bit of Port B.
24,26	NC		NO CONNECTION, i.e. not used

Table 26. Expansion Edge Connector P1 Interface Signals.

Expansion Edge Connector Pin Numbers, Names and Functions					
Connector Pin #	Motorola Pin #	Signal Mnemonic	I/O/B (Active)	Wired-OR	Signal Name and Description
1,2,3,4,5,6	A,1,B,2,C,3	EDGE5 (+5)	I		+5 V DC POWER - For system components. ¹⁴
7	D	IRQ*	I (L)	YES	INTERRUPT REQUEST - A level-sensitive signal used to initiate an Interrupt Service Request (ISR). The signal is latched by μ P on leading edge of E, and is recognized by the μ P after it completes execution of the current instruction. At that time, if the μ P's Condition Code Register (CCR) interrupt mask bit is not set, the μ P will execute the interrupt request.
8	4	EHLT*	I (L)	YES	HALT - Active to requests the halting of μ P activity. The signal is latched by μ P on leading edge of E, and is recognized by the μ P after it completes execution of the current instruction. The halt condition, indicated through BA [pin 25 (P)] and BS [pin 46 (23)], does not grant bus access. See BUSREQ* and BUSGNT.
9	E	NMI*	I (L)	YES	NONMASKABLE INTERRUPT - A level-sensitive signal used to initiate a nonmaskable ISR. The signal is latched by μ P on leading edge of E, and is recognized by the μ P after it completes execution of the current instruction. At that time, regardless of the state of the μ P's CCR interrupt mask bit, the μ P will execute the nonmaskable interrupt request.
10	5	ERST*	B (L)	YES	RESET - Allows the μ P to sense and generate a reset on the edge connector.
11,13,16,17, 18,20,28,29, 33,34,35,37, 38,47,48,49, 50,51,52,53, 54,55,56	F,H,8,K, 9,10,14,S, U,17,V,W, 19,B',24,C' 25,D',26,E' 27,F',28	NC			NO CONNECTION, i.e. pins are not used.
12	6	ER/W	O (H/L)		READ/WRITE - Signal generated by the μ P to indicate μ P read/write activity.
14	7	EQ	O (H)		QUADRATURE CLOCK - Bi-phase clock generated by the μ P. Negating MRDY (asserting MNRDY*) extends Q's low state. See E CLOCK below.

¹⁴ The signal mnemonic EDGE5 was chosen to differentiate signal name nets on the power supply entry-side of the ferrite bead in the PWB.

Expansion Edge Connector Pin Numbers, Names and Functions																				
Connector Pin #	Motorola Pin #	Signal Mnemonic	I/O/B (Active)	Wired-OR	Signal Name and Description															
15	J	EE	O (H)		E CLOCK - Bi-phase clock generated by the μ P. Together, the E and Q clocks indicate transitions in the μ P cycle. Signal can be extended in quarter-cycle increments through the use of the MRDY signal.															
19	L	EMEMCLK	O (H)		MEMORY CLOCK - Same function as E on pin 15 (J).															
21,22	M,11	-12	I		-12 V DC POWER - For serial interface circuitry.															
23	N	EDMA* (EBUSREQ*)	I (L)	YES	DIRECT MEMORY ACCESS/BUS REQUEST - Signal used to request off-board bus mastership. Signal must be active 40ns prior to leading edge of E, and is acknowledged through EBUSGNT pin 30 (15).															
24	12	EREFREQ*	I (L)	YES	REFRESH REQUEST - Indicates a refresh request for DRAM. See EREFGNT pin 26 (13).															
25	P	EBA	O (H)		BUS AVAILABLE - Used in conjunction with EBS to indicate μ P status not system status. <table><tr><td>BA</td><td>BS</td><td>μP State</td></tr><tr><td>0</td><td>0</td><td>Normal (running)</td></tr><tr><td>0</td><td>1</td><td>Interrupt acknowledge</td></tr><tr><td>1</td><td>0</td><td>Sync acknowledge</td></tr><tr><td>1</td><td>1</td><td>Halt or bus grant</td></tr></table>	BA	BS	μ P State	0	0	Normal (running)	0	1	Interrupt acknowledge	1	0	Sync acknowledge	1	1	Halt or bus grant
BA	BS	μ P State																		
0	0	Normal (running)																		
0	1	Interrupt acknowledge																		
1	0	Sync acknowledge																		
1	1	Halt or bus grant																		
26	13	EREFGNT	O (H)		REFRESH GRANT - Acknowledges a refresh request and indicates the refresh grant cycle. While EREFGNT is active, EBUSGNT is inactive. μ P is inactive during EREFGNT, as well as the following bus cycle, making refresh a two-cycle operation.															
27	R	EMRDY (EMNRDY*)	I (L)	YES	MEMORY NOT READY - Extends E high and Q low in quarter-cycle increments when active during the last quarter of the μ P cycle (E active, Q inactive). Signal may be held active for no longer than 10 μ s. Signal permits the system to work with slow memory modules.															
30	15	EBUSGNT	O (H)		BUS GRANT - Acknowledges a bus request and indicates the granting if the bus. Signal is inactive during the first 50 ns of every bus cycle, forcing all bus masters off the bus during this time. This avoids bus contention problems during transfer of bus mastership. Signal is held inactive during refresh grant cycles and during μ P internal refresh cycles.															
31,32	T,16	+12	I		+12 V DC POWER - For serial interface circuitry.															

Expansion Edge Connector Pin Numbers, Names and Functions					
Connector Pin #	Motorola Pin #	Signal Mnemonic	I/O/B (Active)	Wired-OR	Signal Name and Description
36	18	ECLK	O (H)		CLOCK - Same function as E CLOCK on pin 15 (J).
39,40,41,42, 43,44,81,82, 83,84,85,86	X,20,Y,21 Z,22,W',41, X',42,Y',43	GND	I		GROUND - For system components.
45	A'	FIRQ*	I (L)	YES	FAST INTERRUPT REQUEST - A level-sensitive signal used to initiate a fast ISR. The signal is latched by μ P on leading edge of E, and is recognized by the μ P after it completes execution of the current instruction. At that time, if the μ P's CCR interrupt mask bit is not set, the μ P will execute the fast interrupt request. This sequence is fast in the sense that it only stacks the return address and CCR.
46	23	EBS	O (H)		BUS STATUS - See EBA on pin 25 (P).
57	H'	ED3	B		DATA BUS (Bit 3) - One of eight data lines used to transmit data to and from the μ P and its peripherals.
58	29	ED1	B		DATA BUS (Bit 1) - Same function as data bus (Bit 3) on pin 57 (H').
59	J'	ED7	B		DATA BUS (Bit 7) - Same function as data bus (Bit 3) on pin 57 (H').
60	30	ED5	B		DATA BUS (Bit 5) - Same function as data bus (Bit 3) on pin 57 (H').
61	K'	ED2	B		DATA BUS (Bit 2) - Same function as data bus (Bit 3) on pin 57 (H').
62	31	ED0	B		DATA BUS (Bit 0) - Same function as data bus (Bit 3) on pin 57 (H').
63	L'	ED6	B		DATA BUS (Bit 6) - Same function as data bus (Bit 3) on pin 57 (H').
64	32	ED4	B		DATA BUS (Bit 4) - Same function as data bus (Bit 3) on pin 57 (H').
65	M'	EA14	O		ADDRESS BUS (Bit 14) - One of 16 address lines sourced by the μ P to select addressable memory locations in the system.
66	33	EA15	O		ADDRESS BUS (Bit 15) - Same function as address bus (Bit 14), pin 65 (M').
67	N'	EA13	O		ADDRESS BUS (Bit 13) - Same function as address bus (Bit 14), pin 65 (M').
68	34	EA12	O		ADDRESS BUS (Bit 12) - Same function as address bus (Bit 14), pin 65 (M').
69	P'	EA10	O		ADDRESS BUS (Bit 10) - Same function as address bus (Bit 14), pin 65 (M').

Expansion Edge Connector Pin Numbers, Names and Functions

Connector Pin #	Motorola Pin #	Signal Mnemonic	I/O/B (Active)	Wired-OR	Signal Name and Description
70	35	EA11	O		ADDRESS BUS (Bit 11) - Same function as address bus (Bit 14), pin 65 (M').
71	R'	EA9	O		ADDRESS BUS (Bit 9) - Same function as address bus (Bit 14), pin 65 (M').
72	36	EA8	O		ADDRESS BUS (Bit 8) - Same function as address bus (Bit 14), pin 65 (M').
73	S'	EA6	O		ADDRESS BUS (Bit 6) - Same function as address bus (Bit 14), pin 65 (M').
74	37	EA7	O		ADDRESS BUS (Bit 7) - Same function as address bus (Bit 14), pin 65 (M').
75	T'	EA5	O		ADDRESS BUS (Bit 5) - Same function as address bus (Bit 14), pin 65 (M').
76	38	EA4	O		ADDRESS BUS (Bit 4) - Same function as address bus (Bit 14), pin 65 (M').
77	U'	EA2	O		ADDRESS BUS (Bit 2) - Same function as address bus (Bit 14), pin 65 (M').
78	39	EA3	O		ADDRESS BUS (Bit 3) - Same function as address bus (Bit 14), pin 65 (M').
79	V'	EA1	O		ADDRESS BUS (Bit 1) - Same function as address bus (Bit 14), pin 65 (M').
80	40	EA0	O		ADDRESS BUS (Bit 0) - Same function as address bus (Bit 14), pin 65 (M').

Table 27. LCA Pin Signals.
136

LCA Pin Numbers, Names and Functions						
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	C/S	WakeUp	Signal Name and Description
1	GND	I				GROUND - One of two ground pins on the LCA.
2	WRAPCTRL	O (L)	NOFF	S	OFF	ELECTRONIC WRAP-AROUND CONTROL - Primary ACIA wrap-around control. Active forces primary ACIA Tx/D → Rx/D.
3	PLCAB2	O (L)	NON	S	ON	PRIMARY LCA BSR2 CONTROL - Active enables primary status/control signals (PACIA*, PRIIRAM*, PRIROM*, BDRATE).
4	LBDRATE	O		S		LCA BAUD RATE - Rx/C/TxC of both ACIAs (9600 Baud x16). Sources BDRATE.
5	LPRIROM*	O (L)		C	OFF	LCA PRIMARY ROM - Chip select for primary EPROM. Sources PRIROM*.
6	PBSA13	O (L)		C	OFF	PRIMARY BANK SELECT FOR A13 - Translated primary ROM A13.
7	PBSA12	O (L)		C	OFF	PRIMARY BANK SELECT FOR A12 - Translated primary ROM A12.
8	RDTBCHI	O (L)		C	OFF	READ TBC HIGH - Read the high-order data byte from the TBC (OE* of UP2).
9	WRFFHI	O (L)		C	OFF	WRITE FLIP-FLOP HIGH - Write the high-order data byte to TBC's "Write F/F" (CLK of UP1).
10	PDATAB2	O (L)	NON	S	ON	PRIMARY DATA BUS BSR2 CONTROL - Active enables μ P data bus buffer.
11	PADDR	O (L)	NON	S	ON	PRIMARY ADDRESS BUS CONTROL - Active enables both μ P address bus buffers.
12	PWRDWN*	I (L)	NOFF			POWER DOWN - Active disables LCA operation to conserve power. Signal negated by a PU resistor.
13	WRTBCALL	O (L)		C	OFF	WRITE ALL - Write TBC upper and lower Bytes (WR* of TBC and OE* of UP1).
14	SLCAB1	O (L)	NON	S	ON	SECONDARY LCA BSR1 CONTROL - Enable for the secondary chip selects (SACIA* and SECROM*).

LCA Pin Numbers, Names and Functions						
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	C/S	WakeUp	Signal Name and Description
15	SLCAB2	O (L)	NON	S	ON	SECONDARY LCA BSR2 CONTROL - Enable for the secondary chip selects (SECRAM*, SPIA*, and SPTM*).
16	LBUFCTRL	O (L)	NON	S	ON	LCA BUFFER CONTROL - Active enables secondary ACIA Mother Hen (MH) Rx/D.
17	LSACIA*	O (L)		C	OFF	LCA SECONDARY ACIA (MH) - Chip select for secondary ACIA. Sources SACIA*.
18	LSECROM*	O (L)		C	OFF	LCA SECONDARY ROM - Chip select for secondary EPROM. Sources SECROM*.
19	LSECRAM*	O (L)		C	OFF	LCA SECONDARY RAM - Chip select for secondary RAM. Sources SECRAM*.
20	LSPIA*	O (L)		C	OFF	LCA SECONDARY PIA - Chip select for secondary PIA. Sources SPIA*.
21	LSPTM*	O (L)		C	OFF	LCA SECONDARY PTM - Chip select for secondary PTM. Sources SPTM*.
22	Vcc	I				+5 V DC POWER - One of two Vcc pins on the LCA.
23	MAXFREQ	I				MAXIMUM FREQUENCY - The frequency generated by the baud rate generator prior to division by the LCA.
24	SADDR	O (L)	NON	S	OFF	SECONDARY ADDRESS BUS CONTROL - Enables 4 edge connector address buffers.
25	SDATA4B2	O (L)	NON	S	OFF	SECONDARY DATA BUS US4 BSR2 CONTROL - Enables primary ↔ edge data buffers.
26	SDATA5B2	O (L)	NON	S	OFF	SECONDARY DATA BUS US5 BSR2 CONTROL - Enables edge ↔ secondary data buffers.
27	SPTMB1	O (L)	NOFF	S	OFF	SECONDARY PTM BSR1 CONTROL - Enables PIA port B's control of PTM inputs (CB2 → CA2, PB2 → PTM-C3*, PB3 → PTM-G3*).

LCA Pin Numbers, Names and Functions						
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	C/S	WakeUp	Signal Name and Description
28	SPTMB2	O (L)	NOFF	S	OFF	SECONDARY PTM BSR2 CONTROL - Enables PIA port B's control of PTM inputs (PB4 → PTM-C2*, PB5 → PTM-G2*, PB6 → PTM-C1*, PB7 → PTM-G1*).
29	SACIAB1	O (L)	NOFF	S	OFF	SECONDARY ACIA BSR1 CONTROL - ACIA/PIA wrap-around enable (CB2 → CB1 & CA1, secondary ACIA: Tx/D → Rx/D).
30	STATUS	O (L)	NOFF	S	ON	STATUS - Indicates μ P system test status.
31	M1	I				MODE 1 - Used to establish an LCA configuration mode. Wired active via a PD resistor.
32	M0	I				MODE 0 - See Mode 1 above. Also pulled down.
33	M2-PQ	I (H)				MODE 2-PRIMARY Q CLOCK - (Mode 2 prior/during LCA configuration and primary Q clock after). See MODE 1 and E CLOCK for more information.
34	PODHERE*	I (L)				ASSET POD HERE - Negated via a PU resistor when pod not present. Pulled low by ASSET connector.
35	TOERST*	O (L)	NOFF	S?	OFF	RESET TO EDGE CONNECTOR - Allows μ P to generate a reset on the edge connector.
36	LDC-PLCAB2	O (L)	NON	S	OFF	LOW DURING CONFIGURATION-PRIMARY LCA BSR2 CONTROL - (LDC prior/after LCA configuration and enable after) Active enables primary status/control signals (PRST*, Primary ACIA: Rx/D & Tx/D, and FRMERST*).
37	FRMERST*	I (L)				RESET FROM EDGE CONNECTOR - Allows the μ P to sense a reset from the edge connector.
38	PBS	I (H)				PRIMARY BUS STATUS - Indicates μ P Status when used in conjunction with PBA. See P1 connector.
39	PBA	I (H)				PRIMARY BUS AVAILABLE - See above.
40	PR/W	I (H/L)				PRIMARY R/W - Indicates μ P read/write cycles.
41	PDMA*	I (L)				PRIMARY DMA - Indicates an external request to the μ P for bus mastership (i.e for off-board DMA access).
42	PHLT*	I (L)				PRIMARY HALT SIGNAL - Request μ P halt.

LCA Pin Numbers, Names and Functions						
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	C/S	WakeUp	Signal Name and Description
43	GND	I				GROUND - One of two ground pins on the LCA.
44	PMRDY	I (H)				PRIMARY MRDY - Used to extend the E and Q Clocks for Slower Memory Devices. See P1 connector.
45	PE	I (H)				PRIMARY E CLOCK - A bi-phase clock sourced by the μ P. Together the E and Q clocks are used to indicate transitions in the μ P cycle.
46	LPACIA*	O		S	H	LCA PRIMARY ACIA - Chip select for primary ACIA. Sources PACIA*.
47	PCTRLB1	O (L)	NON	S	ON	PRIMARY CONTROL BSR1 CONTROL - Status buffer (around μ P) enables (PQ, PE, PBS, PBA).
48	PCTRLB2	O (L)	NON	S	ON	PRIMARY CONTROL BSR1 CONTROL - Status Buffer (around μ P) enables (PR/W, PDMA*, PHLT*, PMRDY).
49	LPRI RAM*	O		S	H	LCA PRIMARY RAM - Chip select for primary RAM. Sources PRIRAM*.
50	PBSA14	O		S	H	PRIMARY BANK SELECT FOR A14 - Translated primary ROM A14.
51	SCTRLB1	O (L)	NON	S	OFF	SECONDARY CONTROL BSR1 CONTROL - Enable for edge connector control buffer (EQ, EE, EBS, EBA).
52	SCTRLB2	*O (L)	NON	S	OFF	SECONDARY CONTROL BSR1 CONTROL - Enable for edge connector control buffer (ER/W, PDMA*, PHLT*, PMRDY).
53	EBUSGNT	O (H)	NOFF		OFF	BUS GRANT - Edge device granted bus.
54	RESET*	I (L)	NOFF			LCA RESET - LCA reset hardwired to the pushbutton reset for LCA reconfiguration.
55	DONE-PG*	O (H)				LCA DONE-PROGRAM - (Program prior/during LCA configuration and DONE after) - LCA configuration will take place/LCA configuration is completed. $\downarrow$ forces LCA reconfiguration.
56	PA8	I				PRIMARY ADDRESS BUS (Bit 8) - One of 14 address lines from the μ P to the LCA. Used to select memory-mapped register locations within the LCA.
57	PA9	I				PRIMARY ADDRESS BUS (Bit 9) - Same function as PA8 on pin 56.
58	PA10	I				PRIMARY ADDRESS BUS (Bit 10) - Same function as PA8 on pin 56.

LCA Pin Numbers, Names and Functions						
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	C/S	WakeUp	Signal Name and Description
59	PA11	I				PRIMARY ADDRESS BUS (Bit 11) - Same function as PA8 on pin 56.
60	PA12	I				PRIMARY ADDRESS BUS (Bit 12) - Same function as PA8 on pin 56.
61	PA13	I				PRIMARY ADDRESS BUS (Bit 13) - Same function as PA8 on pin 56.
62	PA14	I				PRIMARY ADDRESS BUS (Bit 14) - Same function as PA8 on pin 56.
63	PA15	I				PRIMARY ADDRESS BUS (Bit 15) - Same function as PA8 on pin 56.
64	Vcc	I				+5 V DC POWER - One of two Vcc pins on the LCA.
65	PA7	I				PRIMARY ADDRESS BUS (Bit 7) - Same function as PA8 on pin 56.
66	PA6	I				PRIMARY ADDRESS BUS (Bit 6) - Same function as PA8 on pin 56.
67	TBCCLK	O		S		TBC CLOCK - Divided E clock sourced to the TBC.
68	PA5	I				PRIMARY ADDRESS BUS (Bit 5) - Same function as PA8 on pin 56.
69	PA4	I				PRIMARY ADDRESS BUS (Bit 4) - Same function as PA8 on pin 56.
70	PA3	I				PRIMARY ADDRESS BUS (Bit 3) - Same function as PA8 on pin 56.
71	PA2	I				PRIMARY ADDRESS BUS (Bit 2) - Same function as PA8 on pin 56.
72	DIN-PD0	I/B				DIN-PRIMARY DATA BUS (Bit 0) - (DIN prior/during LCA configuration and PD0 after). Before/During: Used for LCA configuration. After: One of eight data lines used to read/write from/to memory-mapped registers within the LCA.
73	RDTBCLO	O (L)		C	OFF	READ TBC LOW - Read the low-order byte (RD* of TBC and CLK of UP2) from the TBC.
74	CCK	O				LCA CONFIGURATION CLOCK - Clock Input to LCA's Serial ROM.
75	SPIAB2	O (L)	NOFF	S	OFF	SECONDARY PIA BSR2 CONTROL - Enable for PIA data port wrap-around.
76	SPIAB1	O (L)		S	H or L	SECONDARY PIA BSR1 CONTROL - Enable for PIA data wrap-around direction (L: B → A; H: A → B).
77	SACIAB2	O (L)	NOFF	S	OFF	SECONDARY ACIA BSR2 CONTROL - Wrap-around enable for secondary I/O (CA2 → CB2).

LCA Pin Numbers, Names and Functions						
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	C/S	WakeUp	Signal Name and Description
78	PD1	I/B				PRIMARY DATA BUS (Bit 1) - Same function as PD0 on pin 72.
79	PD2	I/B				PRIMARY DATA BUS (Bit 2) - Same function as PD0 on pin 72.
80	PD3	I/B				PRIMARY DATA BUS (Bit 3) - Same function as PD0 on pin 72.
81	PD4	I/B				PRIMARY DATA BUS (Bit 4) - Same function as PD0 on pin 72.
82	PD5	I/B				PRIMARY DATA BUS (Bit 5) - Same function as PD0 on pin 72.
83	PD6	I/B				PRIMARY DATA BUS (Bit 6) - Same function as PD0 on pin 72.
84	PD7	I/B				PRIMARY DATA BUS (Bit 7) - Same function as PD0 on pin 72.

Table 28. PAL U15 Pin Signals.

PAL U15 Pin Numbers, Names and Functions				
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	Signal Name and Description
1	BE	I (H)		BUFFERED E CLOCK - A bi-phase clock sourced by the μ P. Together the E and Q clock are used to indicate transitions in the μ P cycle.
2	BR/W	I (H/L)		BUFFERED R/W - Indicates μ P read/write cycles.
3	BA0	I		BUFFERED ADDRESS BUS (Bit 0) - One of 16 address lines from the μ P to the PALs. Used to fully decode the peripheral chip selects
4	BA1	I		BUFFERED ADDRESS BUS (Bit 1) - Same function as BA0 on pin 3.
5	BA2	I		BUFFERED ADDRESS BUS (Bit 2) - Same function as BA0 on pin 3.
6	BA3	I		BUFFERED ADDRESS BUS (Bit 3) - Same function as BA0 on pin 3.
7	BA4	I		BUFFERED ADDRESS BUS (Bit 4) - Same function as BA0 on pin 3.
8	BA5	I		BUFFERED ADDRESS BUS (Bit 5) - Same function as BA0 on pin 3.
9	BA6	I		BUFFERED ADDRESS BUS (Bit 6) - Same function as BA0 on pin 3.
10	BA7	I		BUFFERED ADDRESS BUS (Bit 7) - Same function as BA0 on pin 3.
11,13	NC			NO CONNECTION - NOT USED.
14	MUXA	I (H)		MULTIPLEXED ADDRESS LINE A - One of three high-order multiplexed (8 $\rightarrow$ 3) address lines emanating from PAL U10.
15	MUXB	I (H)		MULTIPLEXED ADDRESS LINE B - One of three high-order multiplexed (8 $\rightarrow$ 3) address lines emanating from PAL U10.
16	MUXC	I (H)		MULTIPLEXED ADDRESS LINE C - One of three high-order multiplexed (8 $\rightarrow$ 3) address lines emanating from PAL U10.
17	ROM*	O (L)	NOFF	ROM - Chip select for the ROM.
18	RAM1*	O (L)	NOFF	RAM 1 - Chip select for the system RAM.
19	ACIA1*	O (L)	NOFF	ACIA 1 - Chip select for the system ACIA.
20	RAM2*	O (L)	NOFF	RAM 2 - Chip select for the user RAM.
21	ACIA2*	O (L)	NOFF	ACIA 2 - Chip select for the Mother Hen ACIA.
22	PIA*	O (L)	NOFF	PIA - Chip select for the PIA.
23	PTM*	O (L)	NOFF	PTM - Chip select for the PTM.

Table 29. PAL U10 Pin Signals.

PAL U10 Pin Numbers, Names and Functions				
Pin #	Mnemonic	I/O/B (Active)	NON/NOFF	Signal Name and Description
1	BE	I (H)		BUFFERED E CLOCK - A bi-phase clock sourced by the μ P. Together the E and Q clock are used to indicate transitions in the μ P cycle.
2	BA8	I		BUFFERED ADDRESS BUS (Bit 8) - One of 16 address lines from the μ P to the PALs. Used to fully decode the peripheral chip selects
3	BA9	I		BUFFERED ADDRESS BUS (Bit 9) - Same function as BA8 on pin 2.
4	BA10	I		BUFFERED ADDRESS BUS (Bit 10) - Same function as BA8 on pin 2.
5	BA11	I		BUFFERED ADDRESS BUS (Bit 11) - Same function as BA8 on pin 2.
6	BA12	I		BUFFERED ADDRESS BUS (Bit 12) - Same function as BA8 on pin 2.
7	BA13	I		BUFFERED ADDRESS BUS (Bit 13) - Same function as BA8 on pin 2.
8	BA14	I		BUFFERED ADDRESS BUS (Bit 14) - Same function as BA8 on pin 2.
9	BA15	I		BUFFERED ADDRESS BUS (Bit 15) - Same function as BA8 on pin 2.
10, 11, 13	NC			NO CONNECTION - NOT USED.
14	EBUSGNT	O (H)	NOFF	BUS GRANT - Edge device granted bus.
15	BADDR	O (L)		BUFFERED ADDRESS BUFFER CONTROL - Enable for the μ P's address bus buffers.
16	BDATA	O (L)		BUFFERED DATA BUFFER CONTROL - Enable for the μ P's data bus buffer.
17	BCTRL	O (L)	NOFF	BUFFERED CONTROL BUFFER CONTROL - Enable for the μ P's control bus buffer.
18	EADDR	O (L)	NOFF	EDGE ADDRESS BUFFER CONTROL - Enable for the edge connector's address bus buffers.
19	EDATA	O (L)	NOFF	EDGE ADDRESS BUFFER CONTROL - Enable for the edge connector's data bus buffers.
20	ECTRL	O (L)	NOFF	EDGE ADDRESS BUFFER CONTROL - Enable for the edge connector's control bus buffers.
21	MUXA	O (H)		MULTIPLEXED ADDRESS LINE A - One of three high-order multiplexed (8 $\rightarrow$ 3) address lines going to PAL U15.
22	MUXB	O (H)		MULTIPLEXED ADDRESS LINE B - One of three high-order multiplexed (8 $\rightarrow$ 3) address lines going to PAL U15.
23	MUXC	O (H)		MULTIPLEXED ADDRESS LINE C - One of three high-order multiplexed (8 $\rightarrow$ 3) address lines going to PAL U15.

Appendix C: Traditional System Support Information

TRADITIONAL SYSTEM BILL OF MATERIALS

<u>ITEM</u>	<u>QUAN.</u>	<u>DESCRIPTION</u>	<u>REF. DES.</u>
1	24	CAPACITOR, .01 μ F	C1 - C23, C28
2	1	CAPACITOR, .1 μ F	C24
3	2	CAPACITOR, 18 pF	C26, C27
4	1	CAPACITOR, 6.8 μ F	C25
5	1	CRYSTAL, 1.8432 MHz	Y1
6	1	CRYSTAL, 8.00 MHz	Y2
7	1	FERRITE BEAD, 40 MHz	F1
8	1	LIGHT EMITTING DIODE, GREEN	L1
9	1	PAD JUMPER - ACTUALLY NOT CONNECTED	J6
10	1	RESISTOR PAC - BUS CONFIG. 7 Rs, 10 K	R2
11	1	RESISTOR PAC - BUS CONFIG. 7 Rs, 3300	R7
12	2	RESISTOR PAC - ISOLATED CONFIG. 3 Rs, 3300	R3, R9
13	1	RESISTOR PAC - ISOLATED CONFIG. 5 Rs, 3300	R6
14	2	RESISTOR, 1000	R4, R5
15	1	RESISTOR, 10 M	R1
16	1	RESISTOR, 270	R8
17	1	SWITCH, SPST PUSH BUTTON	SW1
18	1	7407 - TTL HEX BUFFER WITH HIGH VOLTAGE OUTPUT	U7
19	6	74LS244 - LS TTL OCTAL BUFFER WITH TS OUT	U2, U11, U17, U19 - U21

TRADITIONAL SYSTEM BILL OF MATERIALS

<u>ITEM</u>	<u>QUAN.</u>	<u>DESCRIPTION</u>	<u>REF. DES.</u>
20	2	74LS245 - LS TTL OCTAL BUS TRANSCEIVER WITH TS OUTJ9, U18	
21	1	2764 - UV ERASABLE PROM 8KX8	U5
22	2	6264 - 8KX8 FAST STATIC RAM	U3, U6
23	1	6821-PERIPHERAL I/F ADAPTER	U1
24	2	6850-ASYNC COMM I/F ADAPTER	U8, U13
25	1	6809 - 8 BIT MICROPROCESSOR	U14
26	1	MC14411 - BIT RATE GENERATOR	U16
27	1	MC145406 - EIA-232-D DRIVER/RECEIVER	U12
28	1	6840 - PROGRAMMABLE TIMER MODULE	U4
29	1	14-PIN EDGE CONNECTOR	J3
30	3	20-PIN EDGE CONNECTOR	J1, J2, J4
31	1	26-PIN EDGE CONNECTOR	J5
32	1	86-PIN EXPANSION EDGE CONNECTOR	P1
33	2	22V10-PAL-REGISTERED OUTPUTS	U10, U15

TRADITIONAL SYSTEM COMPONENT INDEX

<u>REF.DES.</u>	<u>DESCRIPTION</u>	<u>SCHEMATIC PAGE</u>
C1	DCAP	05
C10	DCAP	05
C11	DCAP	05
C12	DCAP	05
C13	DCAP	05
C14	DCAP	05
C15	DCAP	05
C16	DCAP	05
C17	DCAP	05
C18	DCAP	05
C19	DCAP	05
C2	DCAP	05
C20	DCAP	05
C21	DCAP	05
C22	DCAP	05
C23	DCAP	05
C24	DCAP	05
C25	DCAP	05
C26	DCAP	01
C27	DCAP	01
C28	DCAP	02
C3	DCAP	05
C4	DCAP	05
C5	DCAP	05
C6	DCAP	05
C7	DCAP	05
C8	DCAP	05
C9	DCAP	05
F1	DFERRBD	05
J1	^ ^ CONN20	01,05
J2	^ ^ CONN20	03,05
J3	^ ^ CONN14	05
J4	^ ^ CONN20	03,05
J5	^ ^ CONN26	03,05
J6	PADJUMP	05
L1	DLED	05
P1	^ ^ CONN86	01,04,05
R1	DRES	01
R2	DRPAC8	03,05
R3	DRPACI6	01,03,04
R4	DRES	02
R5	DRES	02
R6	DRPACI10	01
R7	DRPAC8	01
R8	DRES	05
R9	DRPACI6	03
SW1	DPB	02

<u>REF.DES.</u>	<u>DESCRIPTION</u>	<u>SCHEMATIC PAGE</u>
U1	6821	03
U10	PAL22V10	02
U11	74LS244	01
U12	MC145406	01,03,05
U13	6850	01
U14	6809	01
U15	PAL22V10	02
U16	MC14411	01
U17	74LS244	01
U18	74LS245	04
U19	74LS244	04
U2	74LS244	02
U20	74LS244	04
U21	74LS244	04
U3	6264	04
U4	6840	03
U5	2764	01
U6	6264	01
U7	7407	02,03,05
U8	6850	03
U9	74LS245	01
Y1	DXTAL	01
Y2	DXTAL	01

TRADITIONAL SYSTEM SIGNAL INDEX

<u>SIGNAL NAME</u>	<u>SCHEMATIC PAGE</u>
+12	01,03,05
+5	01,02,03,04,05
-12	01,03,05
A0	03
A1	03
A2	03
A3	03
A4	03
A5	03
A6	03
A7	03
ACIA1*	01,02
ACIA2*	02,03
B0	03
B1	03
B2	03
B3	03
B4	03
B5	03
B6	03
B7	03
BA	01,02
BA0	01,02,03,04
BA1	01,02,03,04
BA10	01,02,04
BA11	01,02,04
BA12	01,02,04
BA13	01,02
BA14	01,02
BA15	01,02
BA2	01,02,03,04
BA3	01,02,04
BA4	01,02,04
BA5	01,02,04
BA6	01,02,04
BA7	01,02,04
BA8	01,02,04
BA9	01,02,04
BADDR	01,02
BBA	02,04
BBS	02,04
BCTRL	02
BD0	01,03,04
BD1	01,03,04
BD2	01,03,04
BD3	01,03,04
BD4	01,03,04

SIGNAL NAMESCHEMATIC PAGE

BD5	01,03,04
BD6	01,03,04
BD7	01,03,04
BDATA	01,02
BDMA*	02,04
BDRATE	01,03
BE	01,02,03,04
BHLT*	02,04
BMRDY	02,04
BQ	02,04
BR/W	01,02,03,04
BS	01,02
C1*	03
C2*	03
C3*	03
CA1	03
CA2	03
CB1	03
CB2	03
DMA*	01,02
E	01,02
EA0	04
EA1	04
EA10	04
EA11	04
EA12	04
EA2	04
EA3	04
EA4	04
EA5	04
EA6	04
EA7	04
EA8	04
EA9	04
EADDR	02,04
EBA	04
EBS	04
EBUSGNT	02,04
ECTRL	02,04
ED0	04
ED1	04
ED2	04
ED3	04
ED4	04
ED5	04
ED6	04
ED7	04
EDATA	02,04
EDGE5	05

SIGNAL NAMESCHEMATIC PAGE

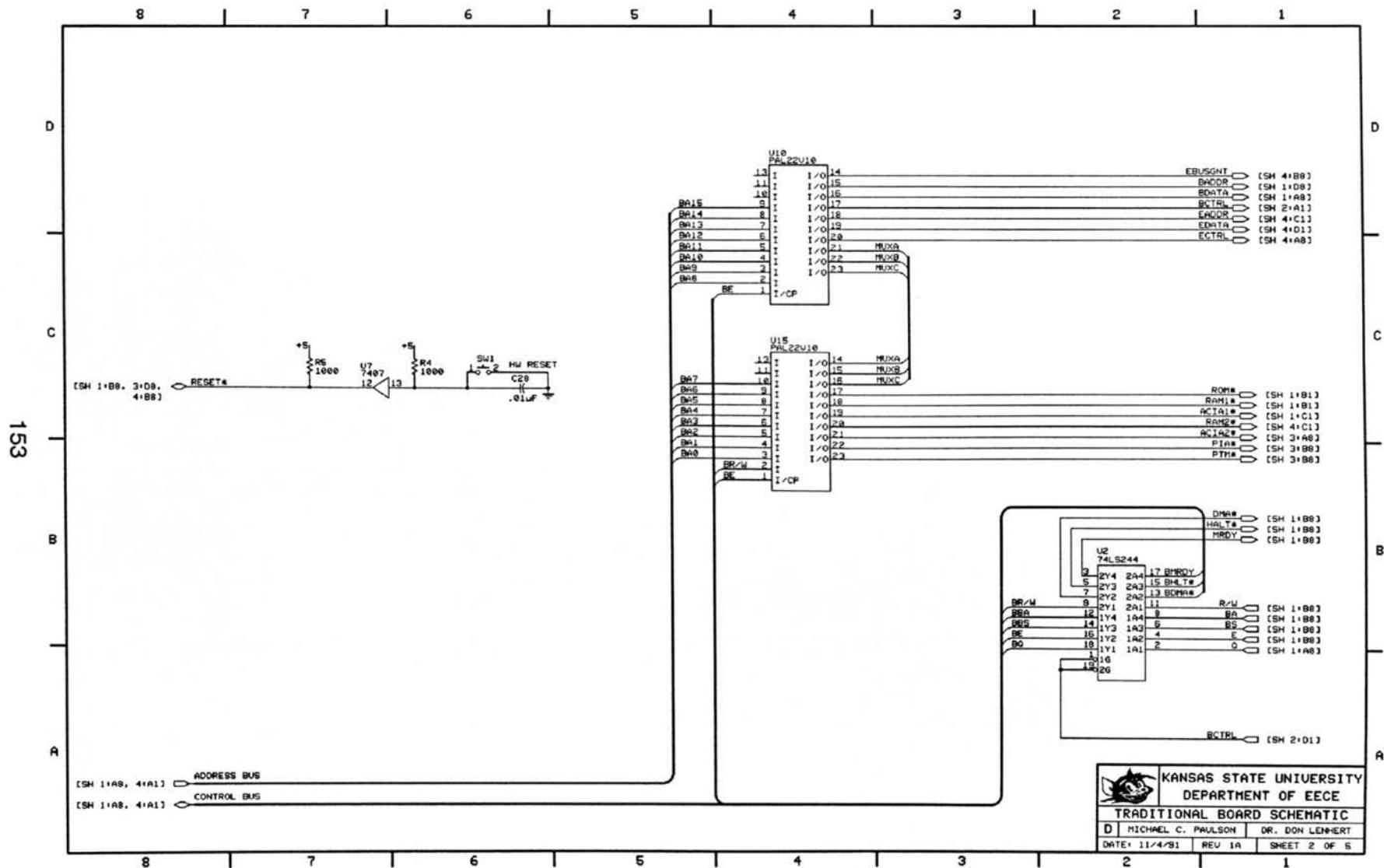
EDMA*	04
EE	04
EHLT*	04
EMRDY	04
EQ	04
ER/W	04
FIRQ*	01
G1*	03
G2*	03
G3*	03
GND	01,02,03,04,05
HALT*	01,02
IRQ*	01,03
MA0	01
MA1	01
MA10	01
MA11	01
MA12	01
MA13	01
MA14	01
MA15	01
MA2	01
MA3	01
MA4	01
MA5	01
MA6	01
MA7	01
MA8	01
MA9	01
MD0	01
MD1	01
MD2	01
MD3	01
MD4	01
MD5	01
MD6	01
MD7	01
MHTXD	03
MRDY	01,02
MUXA	02
MUXB	02
MUXC	02
NMI*	01,03
O2	03
O3	03
PIA*	02,03
PRTS12V	01
PRXD12V	01
PTM*	02,03

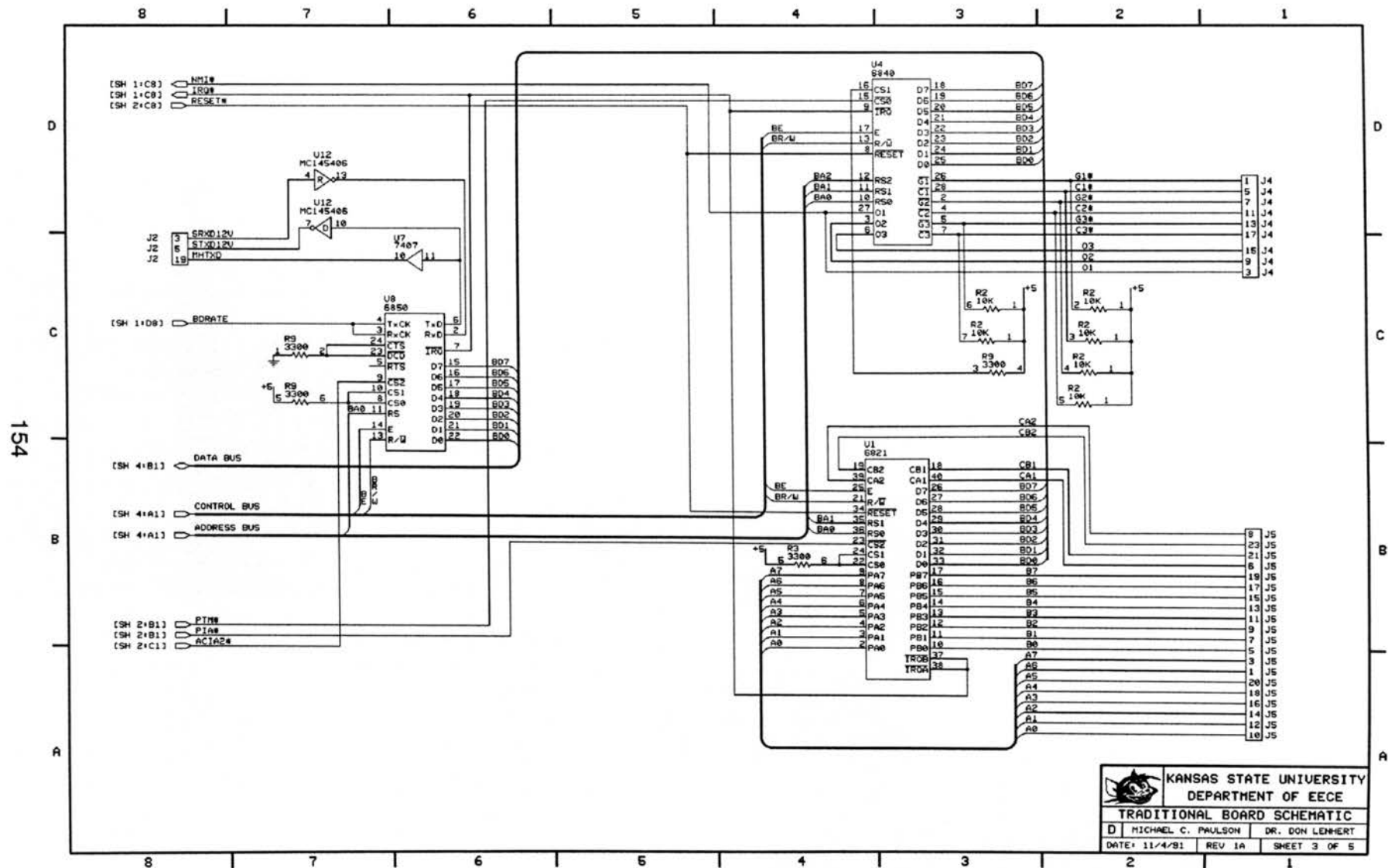
SIGNAL NAMESCHEMATIC PAGE

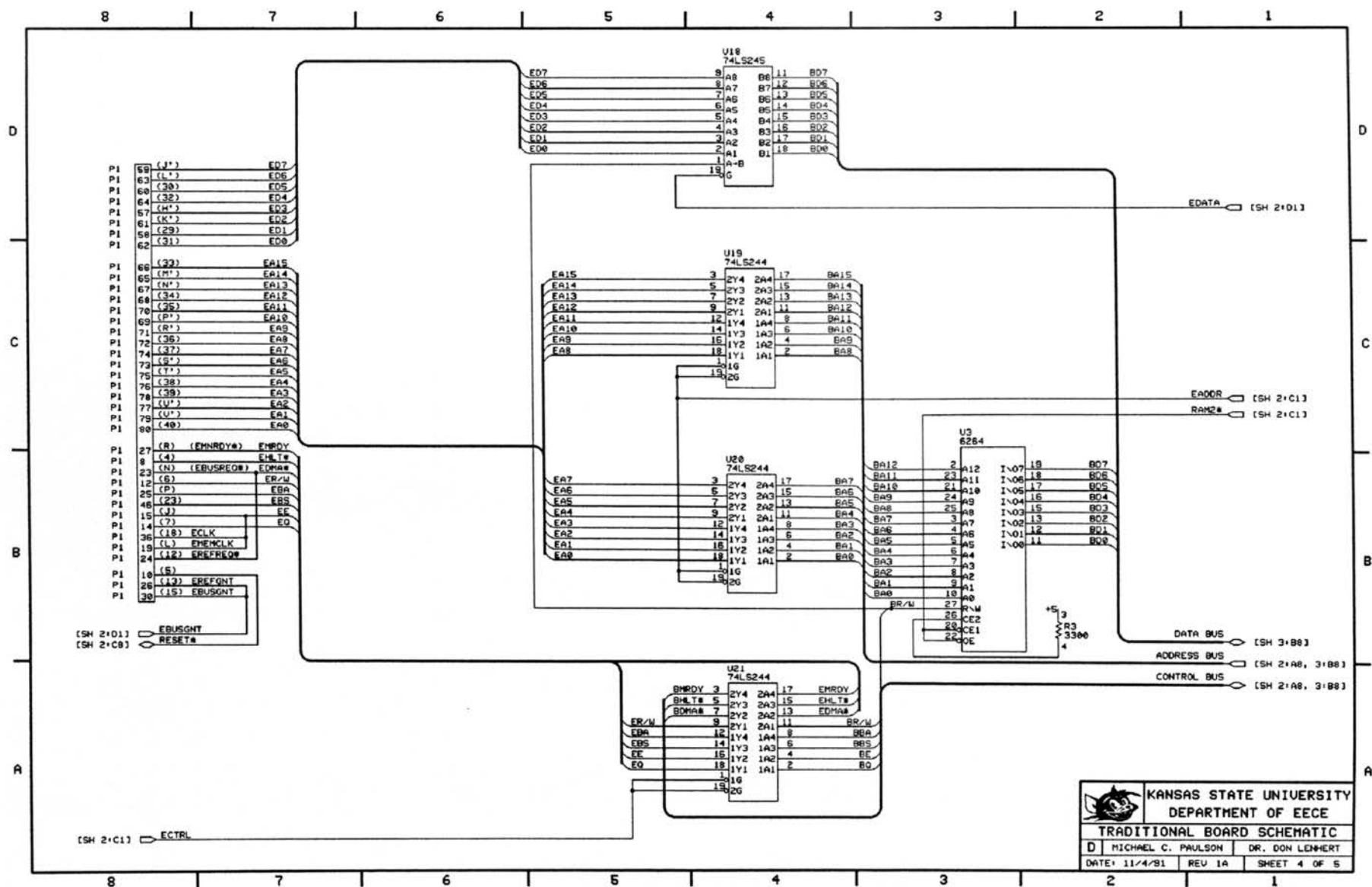
PTXD12V	01
Q	01,02
R/W	01,02
RAM1*	01,02
RAM2*	02,04
RESET*	01,02,03,04
ROM*	01,02
SRXD12V	03
STRAP5V	05
STRAPNEG	05
STRAPPOS	05
STXD12V	03

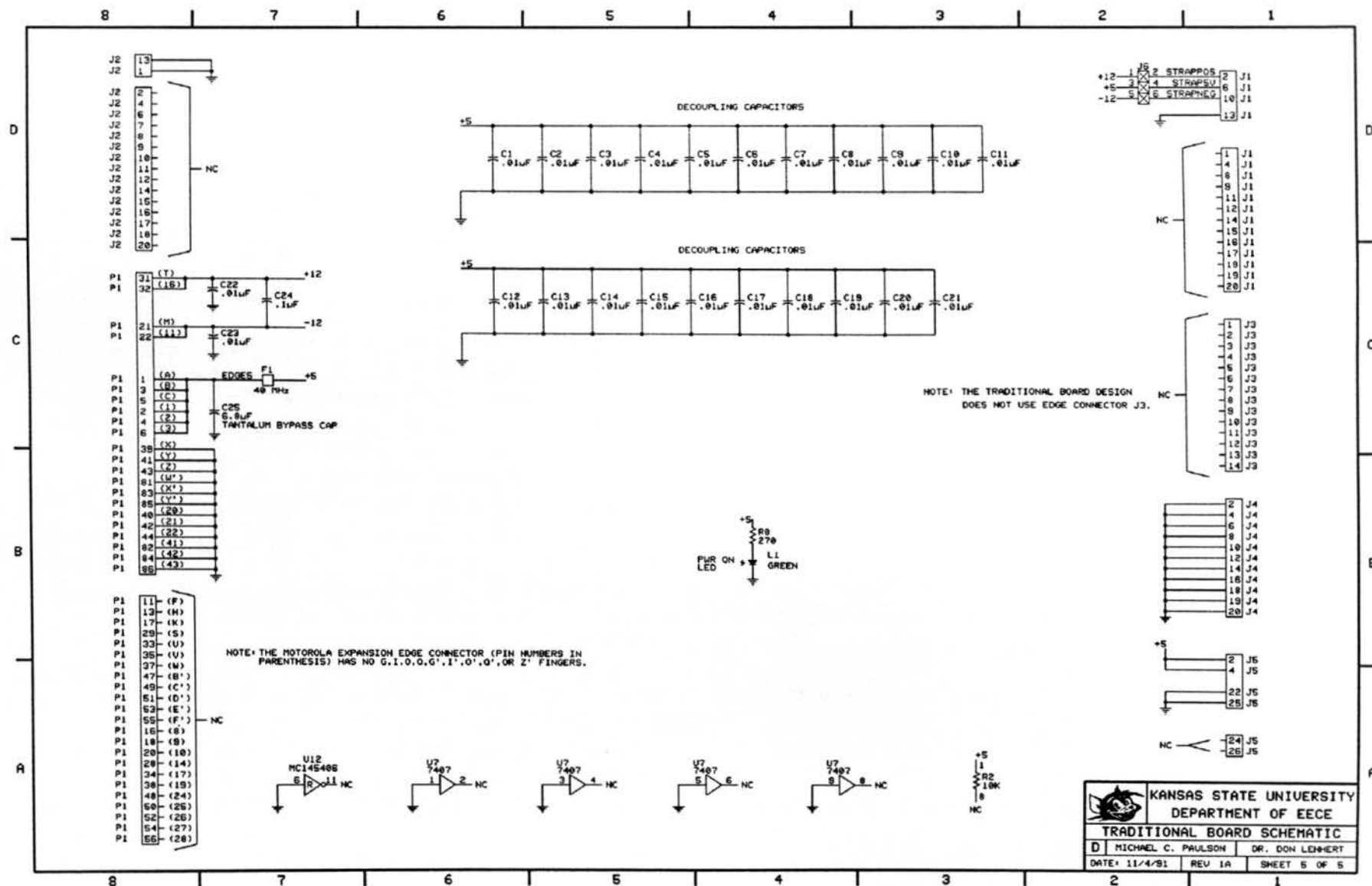


Appendix D: Traditional Board Schematics









Appendix E: DFT System Support Information

DESIGN FOR TEST SYSTEM BILL OF MATERIALS

<u>ITEM</u>	<u>QUAN.</u>	<u>DESCRIPTION</u>	<u>REF. DES.</u>
1	32	CAPACITOR, .01 μ F	C1 - C17, C19 - C22, C24, C26, C30, C31, C33 - C36, C38 - C40
2	4	CAPACITOR, .02 μ F	C18, C23, C25, C27
3	1	CAPACITOR, .1 μ F	C28
4	2	CAPACITOR, 18pF	C29, C32
5	1	CAPACITOR, 6.8 μ F	C37
6	1	CRYSTAL, 1.8432 MHz	Y1
7	1	CRYSTAL, 8.00 MHz	Y2
8	1	FERRITE BEAD, 40 MHz	F1
9	1	LIGHT EMITTING DIODE, GREEN	L1
10	1	LIGHT EMITTING DIODE, RED	L2
11	1	PAD JUMPER - ACTUALLY NOT CONNECTED	J6
12	2	RESISTOR PAC - BUS CONFIG. 5 Rs, 3300	R9, R15
13	1	RESISTOR PAC - BUS CONFIG. 7 Rs, 470	R5
14	1	RESISTOR PAC - BUS CONFIG. 7 Rs, 10 K	R2
15	1	RESISTOR PAC - BUS CONFIG. 7 Rs, 3300	R16
16	1	RESISTOR PAC - ISOLATED CONFIG. 3 Rs, 3300	R8
17	1	RESISTOR PAC - ISOLATED CONFIG. 5 Rs, 3300	R12
18	1	RESISTOR, 10 M	R13

DESIGN FOR TEST SYSTEM BILL OF MATERIALS

<u>ITEM</u>	<u>QUAN.</u>	<u>DESCRIPTION</u>	<u>REF. DES.</u>
19	2	RESISTOR, 270	R11, R14
20	7	RESISTOR, 3300	R1, R3, R4, R6, R7, R10, R17
21	1	SWITCH, SPST PUSH BUTTON	SW1
22	1	7407 - TTL HEX BUFFER WITH HIGH VOLTAGE OUTPUT	U7
23	1	74LS125 - LS TTL QUADRUPLE BUS DRIVER WITH TS OUTPUTS	U11
24	2	27256 - UV ERASABLE PROM 32KX8	U2, U5
25	1	6821-PERIPHERAL I/F ADAPTER	U1
26	2	6850-ASYNC COMM I/F ADAPTER	U8, U13
27	1	6809 - 8 BIT MICROPROCESSOR	U14
28	13	BCT8244 - SCOPE SCAN DEVICE WITH OCTAL BUFFER	UP3, UP5 - UP7, US1 - US3, US6 - US10, US12
29	4	BCT8245 - SCOPE SCAN DEVICE W/ OCTAL TRANSCEIVER	UP4, US4, US5, US11
30	1	ACT8990 - SCOPE TEST BUS CONTROLLER	U9
31	1	XC3042 - XILINX LOGIC CELL ARRAY	U15
32	1	XC1736 - LCA SERIAL CONFIGURATION PROM	U10
33	1	MC14411 - BIT RATE GENERATOR	U16
34	1	MC145406 - EIA-232-D DRIVER/RECEIVER	U12
35	1	6840 - PROGRAMMABLE TIMER MODULE	U4
36	1	14-PIN EDGE CONNECTOR	J3

DESIGN FOR TEST SYSTEM BILL OF MATERIALS

<u>ITEM</u>	<u>QUAN.</u>	<u>DESCRIPTION</u>	<u>REF. DES.</u>
37	3	20-PIN EDGE CONNECTOR	J1, J2, J4
38	1	26-PIN EDGE CONNECTOR	J5
39	1	86-PIN EXPANSION EDGE CONNECTOR	P1
40	2	BCT8374 - SCOPE SCAN DEVICE W/ OCTAL EDGE-TRIG D LATCH	P1, UP2
41	2	62256-CMOS-32,768-WORD BY 8-BIT LSI STATIC RAM	U3, U6

DESIGN FOR TEST SYSTEM COMPONENT INDEX

<u>REF.DES.</u>	<u>DESCRIPTION</u>	<u>SCHEMATIC PAGE</u>
C1	DCAP	05
C10	DCAP	05
C11	DCAP	05
C12	DCAP	05
C13	DCAP	05
C14	DCAP	05
C15	DCAP	05
C16	DCAP	05
C17	DCAP	05
C18	DCAP	05
C19	DCAP	05
C2	DCAP	05
C20	DCAP	05
C21	DCAP	05
C22	DCAP	05
C23	DCAP	05
C24	DCAP	05
C25	DCAP	05
C26	DCAP	05
C27	DCAP	05
C28	DCAP	05
C29	DCAP	01
C3	DCAP	05
C30	DCAP	05
C31	DCAP	05
C32	DCAP	01
C33	DCAP	05
C34	DCAP	05
C35	DCAP	05
C36	DCAP	05
C37	DCAP	05
C38	DCAP	05
C39	DCAP	02
C4	DCAP	05
C40	DCAP	05
C5	DCAP	05
C6	DCAP	05
C7	DCAP	05
C8	DCAP	05
C9	DCAP	05
F1	DFERRBD	05
J1	^ ^ CONN20	01,05
J2	^ ^ CONN20	03,05
J3	^ ^ CONN14	02,05
J4	^ ^ CONN20	03,05
J5	^ ^ CONN26	03,05

<u>REF.DES.</u>	<u>DESCRIPTION</u>	<u>SCHEMATIC PAGE</u>
J6	PADJUMP	05
L1	DLED	05
L2	DLED	05
P1	^ ^ CONN86	01,04,05
R1	DRES	04
R10	DRES	02
R11	DRES	05
R12	DRPAC10	01
R13	DRES	01
R14	DRES	05
R15	DRPAC6	02,03,05
R16	DRPAC8	01,05
R17	DRES	02
R2	DRPAC8	03
R3	DRES	03
R4	DRES	01
R5	DRPAC8	02,05
R6	DRES	04
R7	DRES	04
R8	DRPAC16	02,03
R9	DRPAC6	02,03
SW1	DPB	02
U1	6821	03
U10	XC1736	02
U11	74LS125	01,03,05
U12	MC145406	01,03,05
U13	6850	01
U14	6809	01
U15	3030/42	02
U16	MC14411	01
U2	27256	04
U3	62256	04
U4	6840	03
U5	27256	01
U6	62256	01
U7	7407	02
U8	6850	03
U9	ACT8990	02
UP1	BCT8374	02
UP2	BCT8374	02
UP3	BCT8244	02
UP4	BCT8245	01
UP5	BCT8244	01
UP6	BCT8244	01
UP7	BCT8244	02
US1	BCT8244	04
US10	BCT8244	03
US11	BCT8245	03
US12	BCT8244	03

<u>REF.DES.</u>	<u>DESCRIPTION</u>	<u>SCHEMATIC PAGE</u>
US2	BCT8244	04
US3	BCT8244	04
US4	BCT8245	04
US5	BCT8245	04
US6	BCT8244	04
US7	BCT8244	04
US8	BCT8244	04
US9	BCT8244	03
Y1	DXTAL	01
Y2	DXTAL	01

DESIGN FOR TEST SYSTEM SIGNAL INDEX

<u>SIGNAL NAME</u>	<u>SCHEMATIC PAGE</u>
+12	01,03,05
+5	01,02,03,04,05
-12	01,03,05
A0	03
A1	03
A2	03
A3	03
A4	03
A5	03
A6	03
A7	03
B0	03
B1	03
B2	03
B3	03
B4	03
B5	03
B6	03
B7	03
BA	01,02
BDRATE	01,02,03
BS	01,02
C1*	03
C2*	03
C3*	03
CA1	03
CA2	03
CB1	03
CB2	03
DMA*	01,02
DONE	02
E	01,02
EA0	04
EA1	04
EA10	04
EA11	04
EA12	04
EA13	04
EA14	04
EA15	04
EA2	04
EA3	04
EA4	04
EA5	04
EA6	04
EA7	04

<u>SIGNAL NAME</u>	<u>SCHEMATIC PAGE</u>
EA8	04
EA9	04
EBA	04
EBS	04
EBUSGNT	02,04
ED0	04
ED1	04
ED2	04
ED3	04
ED4	04
ED5	04
ED6	04
ED7	04
EDGE5	05
EDMA*	04
EE	04
EHLT*	04
EMRDY	04
EQ	04
ER/W	04
ERST*	02,04
FIRQ*	01
FRMERST*	02
G1*	03
G2*	03
G3*	03
GND	01,02,03,04,05
HALT*	01,02
IRQ*	01,03
LBDRATE	02
LBUFCTRL	02,03
LCLK	02
LPACIA*	02
LPRIRAM*	02
LPRIROM*	02
LSACIA*	02,03
LSECRAM*	02,03
LSECROM*	02,03
LSPIA*	02,03
LSPTM*	02,03
MA0	01
MA1	01
MA10	01
MA11	01
MA12	01
MA13	01
MA14	01
MA15	01
MA2	01

SIGNAL NAMESCHEMATIC PAGE

MA3	01
MA4	01
MA5	01
MA6	01
MA7	01
MA8	01
MA9	01
MAXFREQ	01,02
MD0	01
MD1	01
MD2	01
MD3	01
MD4	01
MD5	01
MD6	01
MD7	01
MHTXD	03
MRDY	01,02
NMI*	01,03
O2	03
O3	03
PA0	01,02,04
PA1	01,02,04
PA10	01,02,04
PA11	01,02,04
PA12	01,02,04
PA13	01,02,04
PA14	01,02,04
PA15	01,02,04
PA2	01,02,04
PA3	01,02,04
PA4	01,02,04
PA5	01,02,04
PA6	01,02,04
PA7	01,02,04
PA8	01,02,04
PA9	01,02,04
PACIA*	01,02
PADDR	01,02
PBA	02,04
PBS	02,04
PBSA12	01,02
PBSA13	01,02
PBSA14	01,02
PCTRLB1	02
PCTRLB2	02
PD0	01,02,04
PD1	01,02,04
PD2	01,02,04

SIGNAL NAMESCHEMATIC PAGE

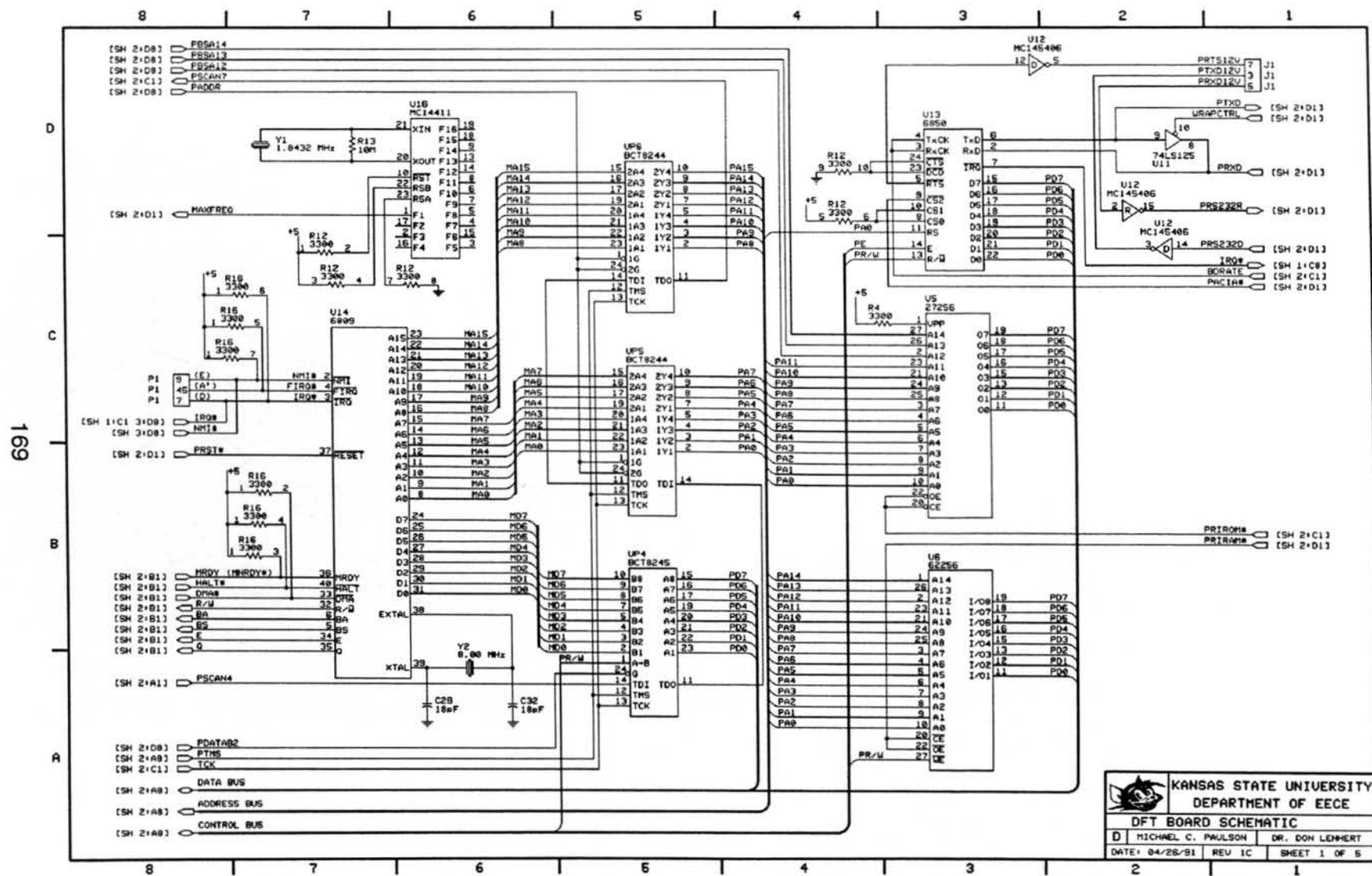
PD3	01,02,04
PD4	01,02,04
PD5	01,02,04
PD6	01,02,04
PD7	01,02,04
PDATAB2	01,02
PDMA*	02,04
PE	01,02,04
PHLT*	02,04
PLCAB1	02
PLCAB2	02
PMRDY	02,04
PODHERE*	02
PQ	02,04
PR/W	01,02,04
PRIRAM*	01,02
PRIROM*	01,02
PRS232D	01,02
PRS232R	01,02
PRST*	01,02
PRTS12V	01
PRXD	01,02
PRXD12V	01
PSCAN4	01,02
PSCAN7	01,02
PTMS	01,02
PTXD	01,02
PTXD12V	01
Q	01,02
R/W	01,02
RDTBCHI	02
RDTBCLO	02
SA0	03,04
SA1	03,04
SA10	04
SA11	04
SA12	04
SA13	04
SA14	04
SA2	03,04
SA3	04
SA4	04
SA5	04
SA6	04
SA7	04
SA8	04
SA9	04
SACIA*	03
SACIAB1	02,03

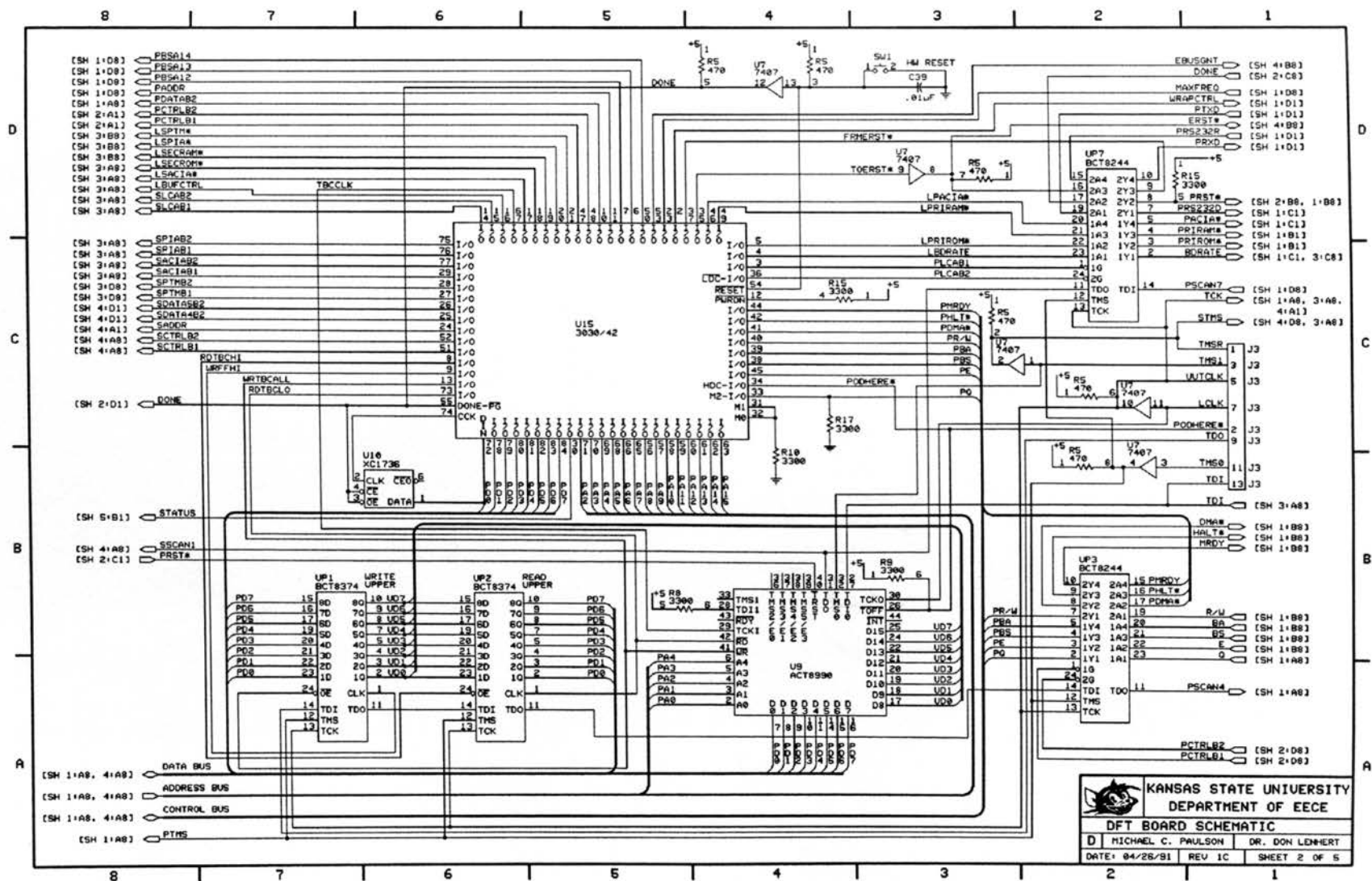
SIGNAL NAMESCHEMATIC PAGE

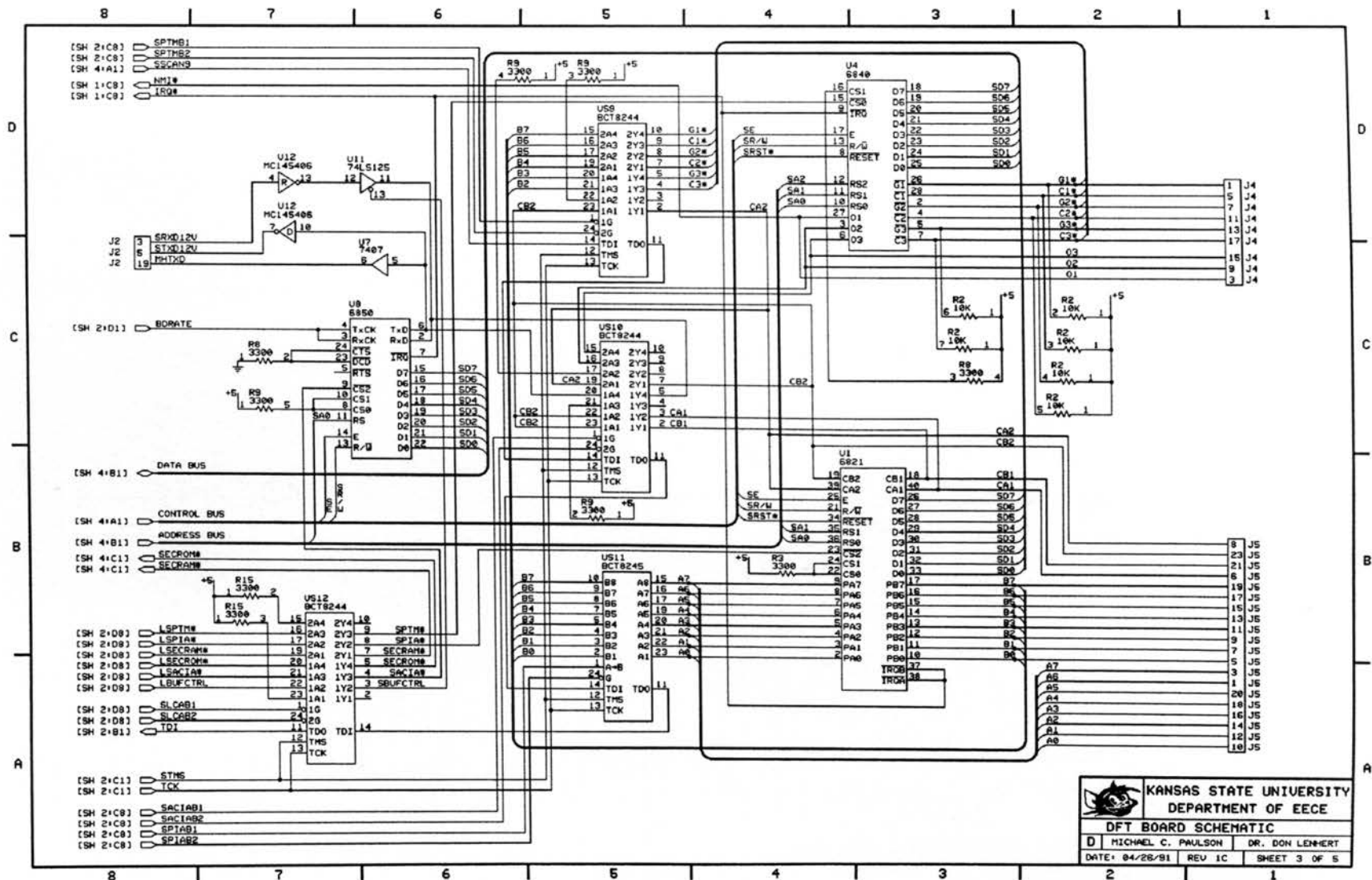
SACIAB2	02,03
SADDR	02,04
SBUFCTRL	03
SCTRLB1	02,04
SCTRLB2	02,04
SD0	03,04
SD1	03,04
SD2	03,04
SD3	03,04
SD4	03,04
SD5	03,04
SD6	03,04
SD7	03,04
SDATA4B2	02,04
SDATA5B2	02,04
SE	03,04
SECRAM*	03,04
SECROM*	03,04
SLCAB1	02,03
SLCAB2	02,03
SPIA*	03
SPIAB1	02,03
SPIAB2	02,03
SPTM*	03
SPTMB1	02,03
SPTMB2	02,03
SR/W	03,04
SRST*	03,04
SRXD12V	03
SSCAN1	02,04
SSCAN9	03,04
STATUS	02,05
STMS	02,03,04
STRAP5V	05
STRAPNEG	05
STRAPPOS	05
STXD12V	03
TBCCLK	02
TCK	01,02,03,04
TDI	02,03
TDO	02
TMS0	02
TMS1	02
TOERST*	02
UD0	02
UD1	02
UD2	02
UD3	02
UD4	02

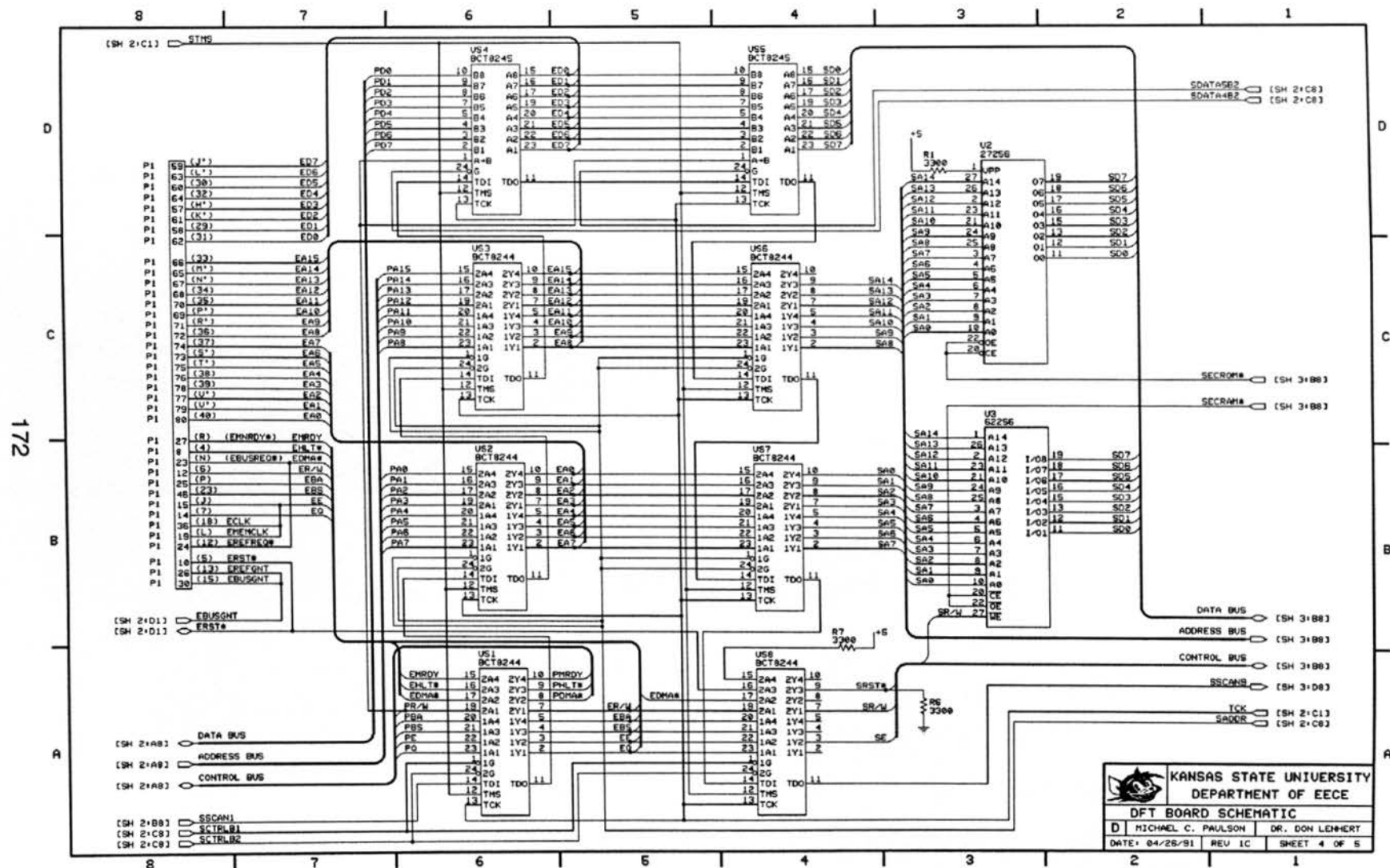
SIGNAL NAMESCHEMATIC PAGE

UD5	02
UD6	02
UD7	02
WRAPCTRL	01,02
WRFFHI	02
WRTBCALL	02









Appendix G: Timing Diagrams & Equations for Design Verification

Since two microprocessor systems with completely different interface logic were designed, two completely separate timing analyses would have to be performed. However, aside from the interface logic, the core components and microprocessor are the same. Therefore, these timing analyses can be combined if care is taken to note certain propagation delay differences. Furthermore, since each design encompass three different peripheral types as well as RAM and ROM, separate analyses would have to be performed for each of these cases as well.

Fortunately, the PTM, PIA, and ACIAs are all from the same family of components and all have identical timing diagrams and parameters. Furthermore, the memory read diagrams for the EPROM and RAM were combined since they were so similar. To indicate memory parameter differences, the EPROM timing parameter names are shown in parenthesis next to their SRAM counterparts.

The DFT system analysis was separated into two DFT board area timing analyses; one with primary area memories and a peripheral (the ACIA), and one with secondary board area components. This separation was necessary because peripherals in the secondary area are double buffered, while those in the primary are not. Since the traditional design is not partitioned, only one analysis case exists. Therefore, a total of three timing analysis cases are considered--each with a different set of interface logic delays. Since only the number and lengths of these delays changed from one system or board area analysis to another, the equations were written using the "worst-case" timing delays of the three.

Therefore, not only was a single figure constructed to indicate all possible microprocessor/peripheral (or memory) interfaces, but only four diagrams are used to

convey all necessary timing information¹⁵. Peripheral read and write cycles required two separate diagrams, as did the on-board memory.

Figure 25 represents the generic drawing of both microprocessor systems--the two DFT board areas and the traditional system are portrayed in this one figure. The following discussion will explain these details and the timing parameters and equations derived from this drawing and the read and write timing diagrams found in Figure 26, Figure 27, Figure 28, and Figure 29.

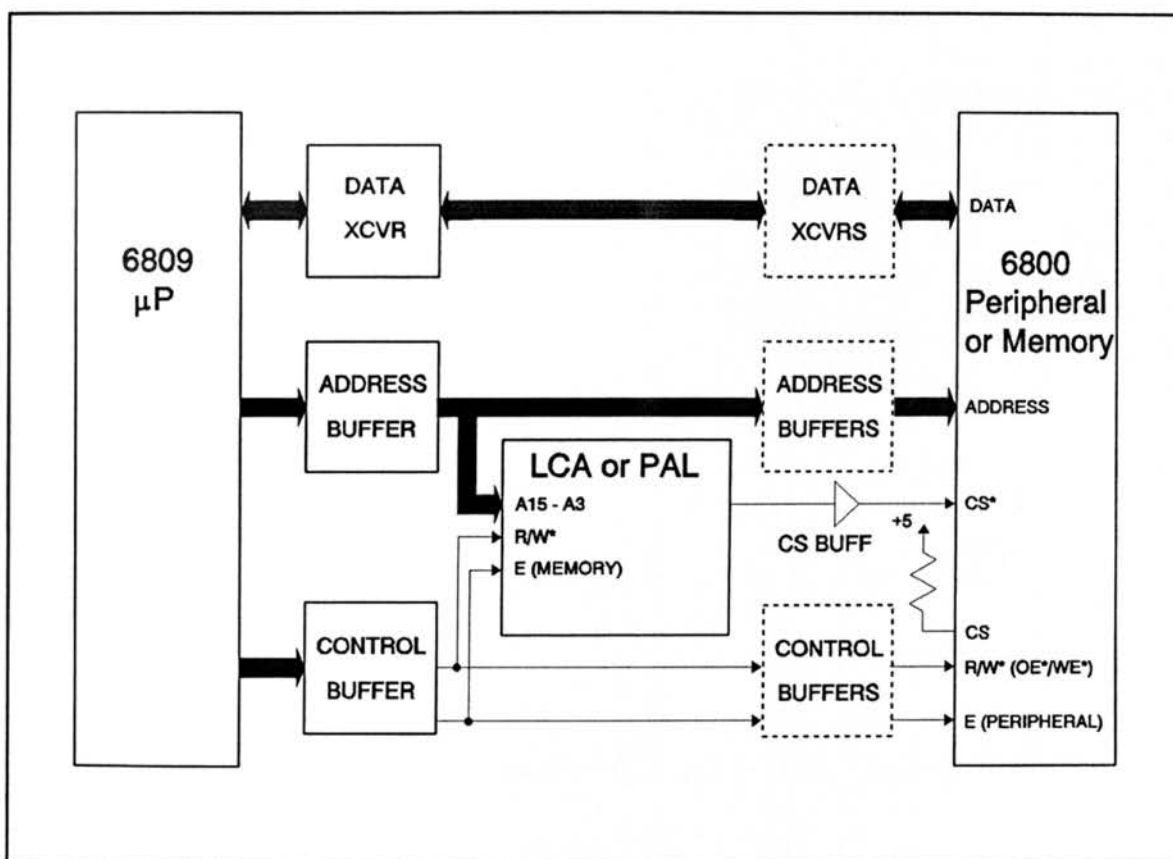


Figure 25. Generic System Block Diagram for Timing Analysis.

The solid address, data, and control bus drivers are used to indicate both the DFT's primary area as well as the traditional board's buffered area. The bus drivers indicated

¹⁵ The timing diagrams were modified versions of ones graciously provided to the author by Dr. D. Gordon.

by a dotted line represent the DFT's secondary board area. For primary area or traditional board timing analysis, they are not present.

To provide all this system information in one diagram, the interface signals represented different read/write system signal interface lines depending on the type of analysis performed (e.g. peripheral or memory and read or write cycle). "A" represents the register select lines on the peripherals and the address bus for the memory. "CS" and "CS*" represent all the active-high and active-low 68xx chip select signals. Finally, "R/W*" represents the R/W* line on the Motorola peripherals (obviously) and the OE* or WE* for the ROM and RAM. Table 30 describes the peripheral signal representation in more detail.

Table 30. General Peripheral Signal Descriptions.

	PIA (6821)	PTM (6840)	ACIA (6850)
R	RS1 (A1), RS0 (A0)	RS2 (A2), RS1 (A1), RS0 (A0)	RS (A0)
CS	CS1, CS0	CS1	CS1, CS0
CS*	CS2*	CS0*	CS2*

The following nomenclature is used to describe the timing analysis in this section:

- Peripheral or memory timing parameters are designated with a capital T.
- Processor timing parameters are designated with a lowercase t.
- Delays for the interface logic (buffers, transceivers, LCA, or PALs) are designated with an uppercase T and an appropriate subscript indicating its function (e.g. T_A is the propagation delay through the address buffers).
- Subscript case is not important, i.e. $T_{sdr} = T_{SDR}$.

Furthermore, Table 31 describes this terminology with respect to the actual components used. These analyses assume that:

- the host processor is a 2 MHz Motorola 6809, i.e. an MC68B09;
- the 68xx peripherals are from the same 68Bxx family;
- programmable logic delays are fixed; and
- NEC's μ PD43256A and μ PD28C256 SRAM and EPROM memory parameters were used in the memory timing analysis [NEC 1991].

Table 31. Interface Delay Terminology.

T_{SB}	= Propagation delay through BCT8244 SCOPE buffer.
T_{SX}	= Propagation delay through BCT8245 SCOPE transceiver.
T_B	= Propagation delay through 74LS244 buffer.
T_X	= Propagation delay through 74LS245 transceiver.
T_A	= Propagation delay through address buffer(s).
T_D	= Propagation delay through data transceiver(s).
T_C	= Propagation delay through control buffer(s).
T_{LCA}	= Propagation delay through LCA.
T_{PAL}	= Propagation delay through PAL.

Table 32. Interface Delays.DFT Board (BCT buffers and transceivers)

$$T_{LCA} = 35.8_{(max)}$$

Primary

$$T_A = T_R = T_{SB} = 9.6_{(max)}, 1.6_{(min)}$$

$$T_D = T_{SX} = 9.4_{(max)}, 1.6_{(min)}$$

$$T_C = 2T_{SB} + T_{LCA} = 19.2_{(max)} + 35.8 = 55_{(max)}, 0_{(min)}$$

Secondary

$$T_A = T_R = 3T_{SB} = 28.8_{(max)}, 4.8_{(min)}$$

$$T_D = 3T_{SX} = 28.2_{(max)}, 4.8_{(min)}$$

$$T_C = 2T_{SB} + T_{LCA} = 19.2_{(max)} + 35.8 = 55_{(max)}, 0_{(min)}$$

Traditional Board (LS buffers and transceivers)

$$T_{PAL} = 35_{(max)}$$

$$T_A = T_R = T_B = 30_{(max)}, 0_{(min)}$$

$$T_D = T_X = 40_{(max)}, 0_{(min)}$$

$$T_C = T_B + T_{PAL} = 30_{(max)} + 35_{(max)} = 65_{(max)}, 0_{(min)}$$

The LCA delays were determined by simulation since the Xilinx data book specifies many different delays with respect to the Input/Output Blocks (IOB) and the Configurable Logic Blocks (CLB) [Xilinx, 1991]. In addition, there are three possible LCA speed grades so the slowest (50 MHz) version was used [Gordon 1991b].

Write Cycle - 6800 Peripheral (See Figure 26)¹⁶

This is the simplest case of the timing equations: the MC6809 is receiving all signals. Thus, one equation must be written for each demand the peripheral makes on the processor.

$$1. T_{as} = P_{WEL} - t_{avs} - t_r + t_{aq} - T_A$$
$$T_{A(max)} \leq 40 \text{ ns}$$

This equation is easily satisfied for all three $T_{A(max)}$ cases.

$$2. T_{ah} = T_A + t_{ah}$$
$$T_{ah(min)} \leq 20 + T_{R(min)}$$

This equation is easily satisfied, since $T_{ah(min)} = 10 \text{ ns}$, so any $T_{R(min)}$ value would work.

$$3. T_{CS} = P_{WEL} - t_{avs} - t_r + t_{aq} - T_C$$
$$T_{CS(min)} \leq 25$$

Since $T_{CS(min)}$ is 40 ns, the equation is violated.

$$4. T_{CH} = T_C^* + t_{ah}$$
$$T_{CH(min)} \leq 20$$

Since $T_{CH(min)}$ is 10 ns, the equation is satisfied.

$$5. T_{dhw} = T_D + t_{dhw}$$
$$T_{dhw(min)} \leq 30 + T_{D(min)}$$

Since $T_{dhw(min)} = 10 \text{ ns}$, any $T_{D(min)}$ value will work.

$$6. T_{dsw} = t_{cyc} - t_f - t_{avs} - t_r - t_{ddq} - T_D$$
$$T_{dsw(min)} \leq 225 - T_{D(max)}$$

¹⁶ "*" indicates that a minimum propagation delay of zero (0) was assumed.

"***" indicates that the maximum propagation delay for the possible timing cases was chosen.

Since $T_{dsw(min)} = 60$ and the worst-case $T_{D(max)}$ for all three cases is 40 ns, this equation is satisfied.

To summarize, the "Peripheral Write Cycle" equations are satisfied for all parameters but the peripheral chip select time. Since the peripheral and processor are from the same family this parameter may still be met.

Read Cycle - 6800 Peripheral (See Figure 27)

The peripheral read cycle is slightly more complicated than the write cycle. In a read cycle, the processor both sends and receives the data from the peripheral, so the processor's setup and hold times must be considered as well. The first four equations are identical to the write cycle equations since the peripheral's timing parameters are considered.

$$\begin{aligned} 1. T_{as} &= P_{WEL} - t_{avs} - t_r + t_{aq} - T_A \\ T_{as(min)} &\leq 80 - T_{A(max)} \end{aligned}$$

This equation is easily satisfied for all three $T_{A(max)}$ cases, since $T_{as(min)} = 40$ ns.

$$\begin{aligned} 2. T_{ah} &= T_A + t_{ah} \\ T_{ah(min)} &\leq 20 + T_{R(min)} \end{aligned}$$

This equation is easily satisfied, since $T_{ah(min)} = 10$ ns, so any $T_{A(min)}$ value would work.

$$\begin{aligned} 3. T_{CS} &= P_{WEL} - T_C - t_{avs} - t_r + t_{aq} \\ T_{CS(min)} &\leq 25 \end{aligned}$$

Like the peripheral write cycle, $T_{CS(min)}$ is 40 ns, so the equation is violated.

$$\begin{aligned} 4. T_{CH} &= T_C^* + t_{ah} \\ T_{CH(min)} &\leq 20 \end{aligned}$$

Since $T_{CH(min)}$ is 10 ns, the equation is satisfied.

Finally, the *processor's* data setup and hold times must be considered.

$$\begin{aligned} 5. t_{dhr} &= T_{dhr} + T_D^* \\ t_{dhr(min)} &\leq 20 \end{aligned}$$

Since $t_{dhr(min)} = 10$ ns, this equation is satisfied.

$$\begin{aligned} 6. t_{dsr(min)} &= P_{WEH} - T_{ddr(max)} - T_{D(max)} \\ 40 &\leq 70 - T_{D(max)} \end{aligned}$$

Since the worst-case $T_{D(max)}$ is 40 ns for the traditional system, this equation is violated. For the DFT system, $T_{D(max)}$ is 28.8 ns, so it is *just* met for the DFT system.

The "Peripheral Read Cycle" equations are not all satisfied. Once again, the peripheral chip select setup time is violated for both systems. In addition, the traditional system also violates a processor read data setup time. These equation violations may or may not cause interface problems.

Write Cycle - SRAM Memory (See Figure 28)

Unlike the peripheral equations, this analysis represents a single SRAM-to-processor timing analysis. Once again, three cases are considered--the DFT system's primary SRAM, its secondary SRAM, and a traditional design's SRAM. One equation was written for each demand the memory makes on the processor. A total of eight equations were written.

$$1. T_{wc} = T_A + t_{ah} + t_{cyc} - t_{avs} - t_r + t_{aq} - T_A$$
$$T_{wc(min)} \leq 390$$

This equation is satisfied since $T_{wc(min)} = 150$ ns.

$$2. T_{cw} = P_{WEH} \text{ (by inspection)}$$
$$T_{cw(min)} \leq 220$$

Since $T_{cw(min)}$ is 100, the equation is satisfied.

$$3. T_{aw} = T_R + t_{ah} + t_{cyc} - t_{avs} - t_r + t_{aq} - T_A$$
$$T_{aw(min)} \leq 360$$

Since $T_{aw(min)} = 100$ ns, this equation is satisfied.

$$4. T_{as} = T_R - T_A$$
$$T_{as(min)} \leq T_{R(min)} - T_{A(max)}$$

Technically, $T_{A(max)} = 30$ ns and $T_{R(min)}$ is zero (0), but since this is a physical system and since $T_{as(min)}$ is 0, this equation may be met.

$$5. T_{wp} = T_R + t_{ah} + t_{cyc} - t_{avs} - t_r + t_{aq} - T_R$$
$$T_{wp(min)} \leq 390$$

Since $T_{wp(min)} = 90$, the equation is satisfied.

$$6. T_{wr} = T_A + t_{ah} - T_C$$
$$T_{C(min)} \leq 20$$

$T_{C(min)}$ is specified to be zero on the traditional design so the equation is met. However, no minimum value is specified for $T_{C(min)}$ on the DFT

system. If T_{LCA} is considered to be zero, then this equation is met for both DFT timing analyses.

$$7. T_{dw} = T_C^* - t_f + t_{cyc} - t_{avs} - t_r - t_{ddq} - T_D$$

$$T_{dw(min)} \leq 185$$

This equation is satisfied since $T_{dw(min)} = 35$ ns.

$$8. T_{dh} = T_D^* + t_{dwh} - T_C$$

This equation is satisfied since $T_{dh(min)}$ is specified to be zero.

The "SRAM Write Cycle" equations are completely satisfied for the all cases. However, certain minimum logic delays were assumed again.

Read Cycle - Memory (SRAM and EPROM--Figure 29)

The memory read cycles are more complicated than any of the previous timing analyses. The read timing diagram in Figure 29 represents both the SRAM and EPROM timing relationships. Therefore this section actually comprises both memory read cases. As was the case in the other read cycle, the processor both sends and receives the data from the memory, so the processor's setup and hold times must be considered to make an accurate timing analysis. The first equation is for the SRAM only, but the remaining four equations pertain to both memories. In the first two, memory is the constraining factor, while the remaining equations are processor-based equations.

$$\begin{aligned} 1. T_{rc} &= t_{cyc} \text{ (by inspection)} \\ T_{rc(min)} &\leq 500 \end{aligned}$$

Since $T_{RC(min)} = 150 \text{ ns}$, this equation is satisfied.

$$\begin{aligned} 2. T_{OE} &= t_{cyc} - T_D - t_{dsr} - t_f - t_{avs} - t_r + t_{aq} - T_R \\ t_{dsr(max)} &\leq 275 \end{aligned}$$

Although $t_{dsr(max)}$ is not specified it is most likely less than 275 ns.

$$\begin{aligned} 3. T_{aa}^{**} (T_{acc}^{**}) &= t_{cyc} - t_f - T_D - t_{dsr} - T_A + t_{aq} - t_{avs} - t_r \\ t_{dsr(min)} &\leq 81.2 \end{aligned}$$

Since $t_{dsr(min)} = 40 \text{ ns}$, this equation is satisfied.

$$\begin{aligned} 4. T_{acs} (T_{ce}) &= P_{WEH} - T_D^* - t_{dsr} - T_C^* \\ T_{C(min)} &\leq 20 \end{aligned}$$

$T_{C(min)}$ is specified to be zero on the traditional design so the equation is met. However, no minimum value is specified for $T_{C(min)}$ on the DFT system. If T_{LCA} is considered to be zero, then this equation is met for both DFT timing analyses as well.

$$\begin{aligned} 5. T_{chz} &= t_{dhr} - T_D^* - T_C^* + t_f \\ 10 \leq T_{CHZ(min)} &- 20 \end{aligned}$$

This equation is satisfied for any $T_{CHZ(min)}$ value of 30 ns or more. No minimum value is given for $T_{CHZ(min)}$ but the max is 30 ns.

The "Memory Read Cycle" equations do not constrain the interface logic any more than it has already been constrained. Once again, certain minimum and maximum logic delays were assumed. In particular, no T_{LCA} minimum delay is known (or at least not specified). It was assumed to be zero to satisfy some of these system equations.

Figure 27. MC6809/6800 Peripheral Read Timing Diagram.

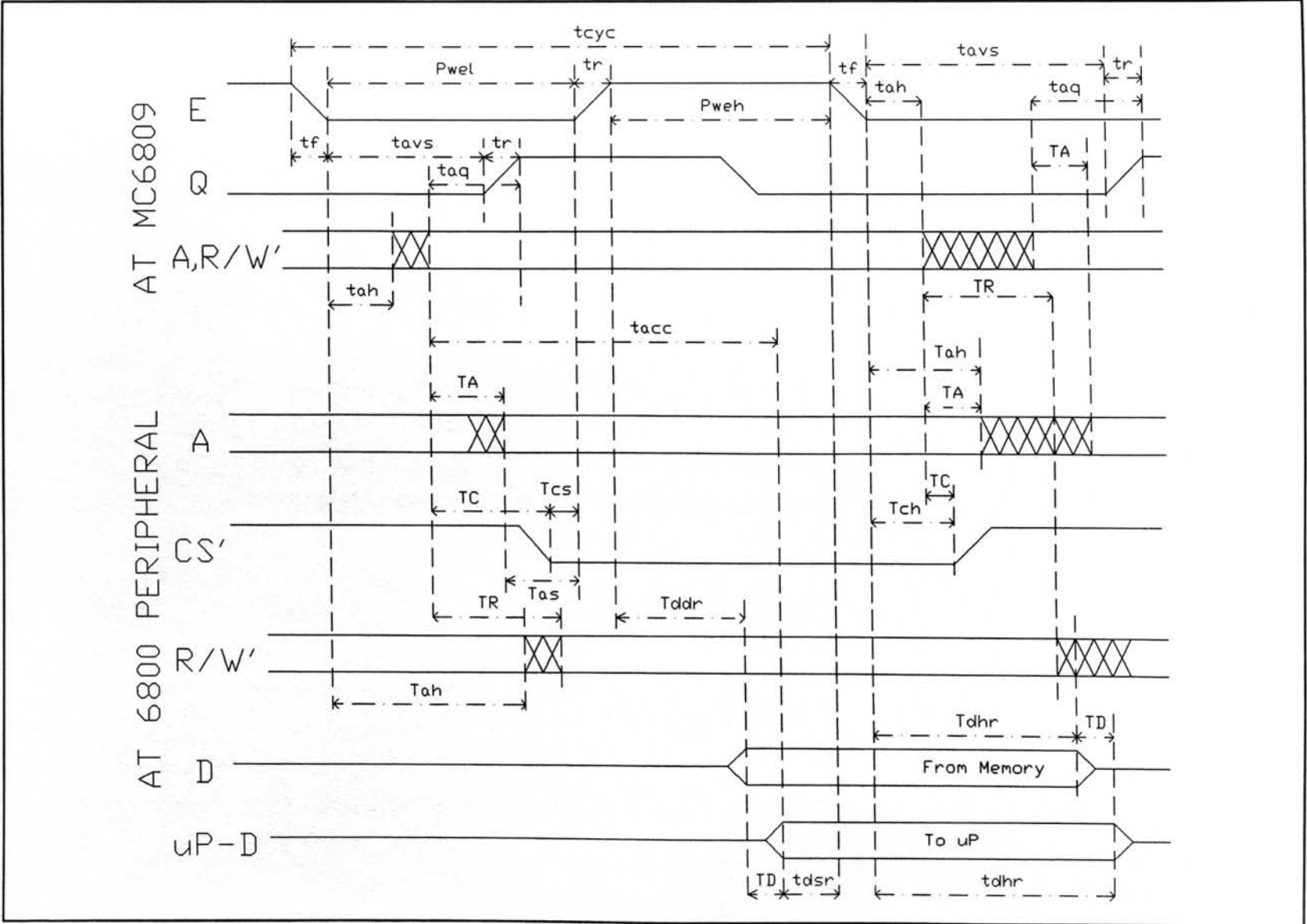
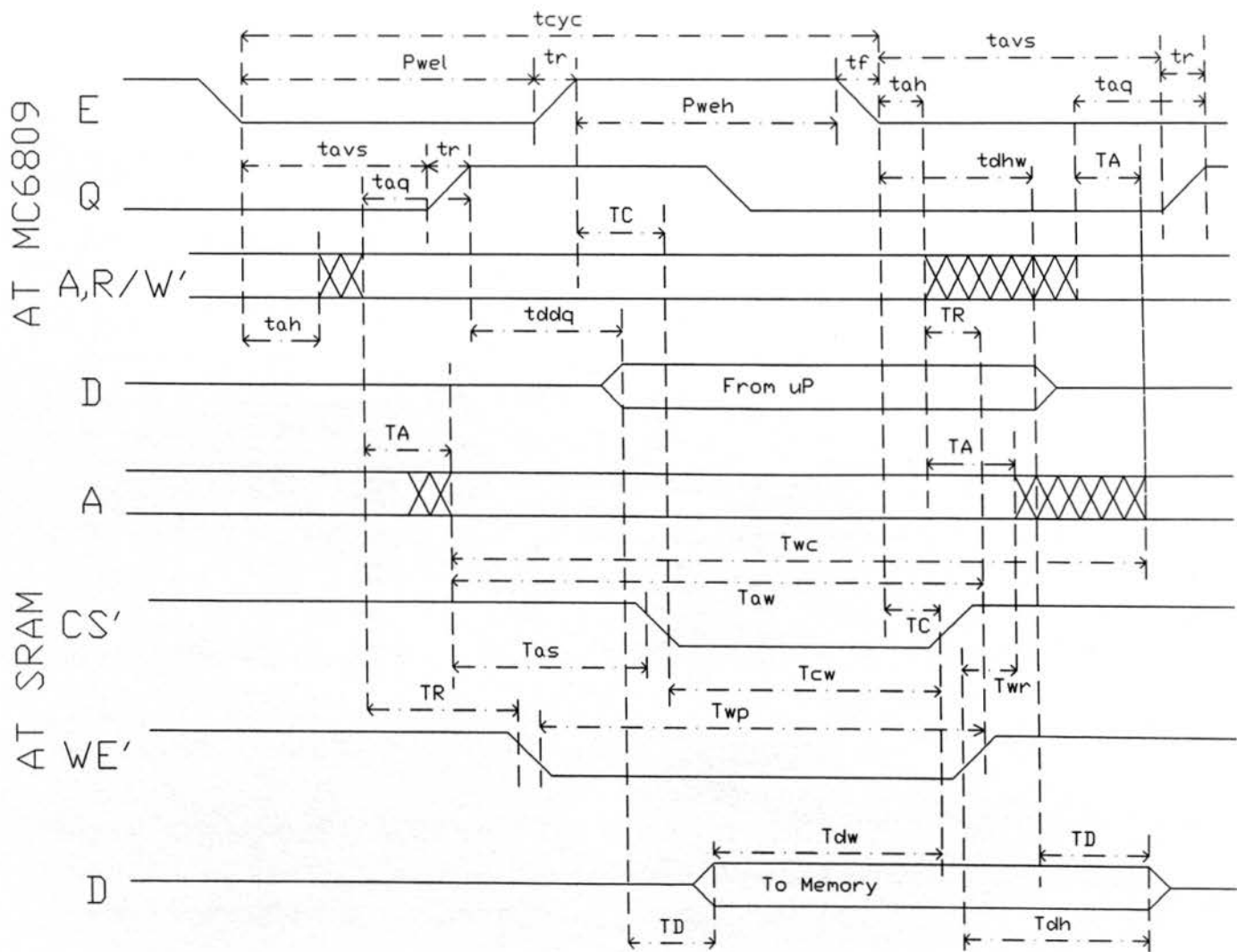


Figure 28. MC6809/SPAM Write Timing Diagram.



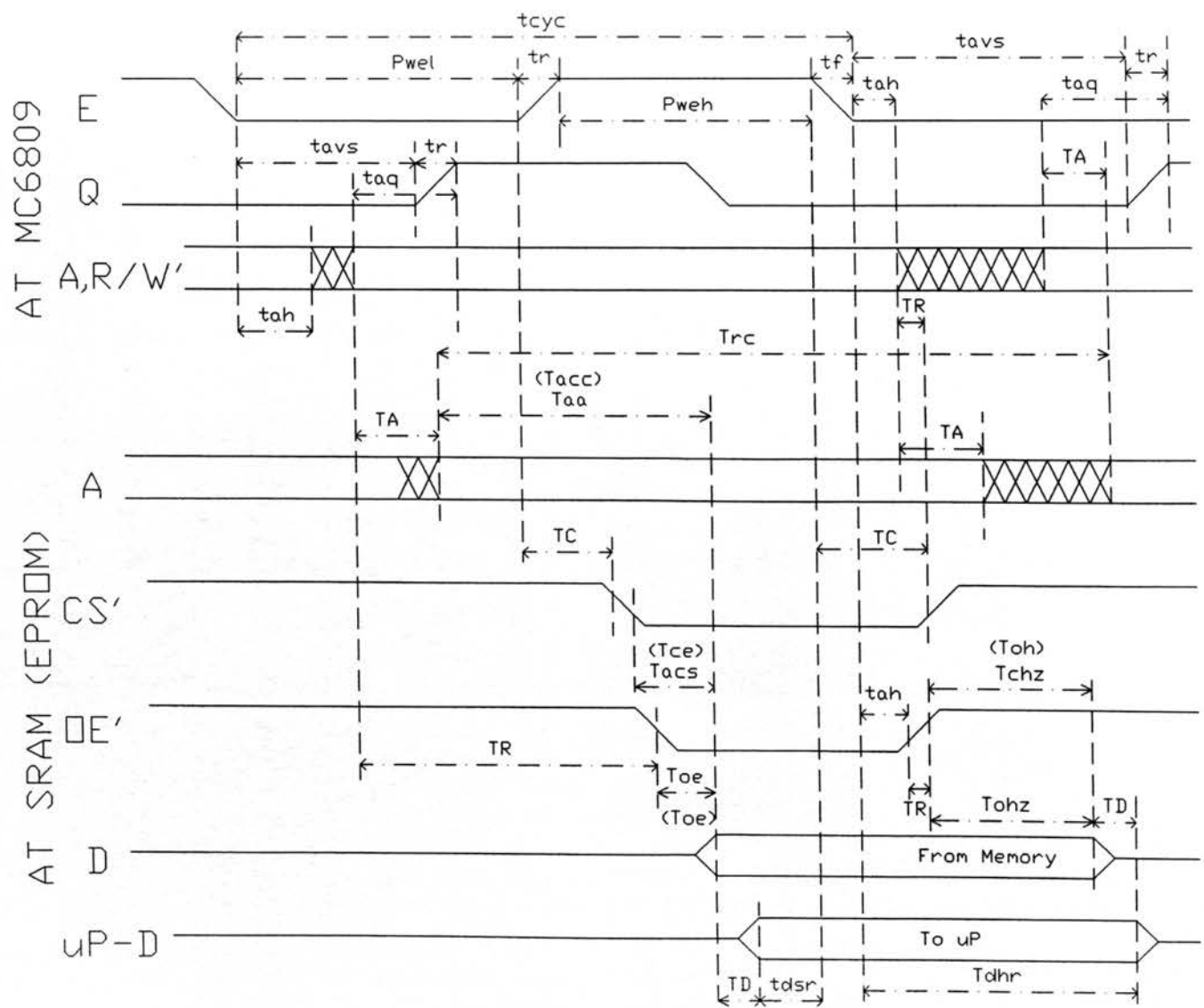


Figure 29. MC6809/SRAM (EPROM) Read Timing Diagram.

Appendix H: DFT PWB Specifications, Layout, and Artwork

This section shows the PWB information necessary to both fabricate the DFT board and produce a similarly fabricated traditional PWB board. The PWB artwork specifications that were used in laying out the DFT PWB system are shown in Table 33, and the PWB dimensions are shown in Figure 30.

The DFT PWB artwork can be found on pages 194, 195, 196, and 197. The component layout or assembly drawing is shown in Figure 31. This figure (minus the board outline) also doubles for the silk screen. The plots shown in Figure 32 and Figure 33 are the component and noncomponent sides of the DFT microprocessor system. Finally, a combined plot showing all of the system features (except the silk screen information) is shown in Figure 34.

A comparable 9.75" x 6.00" traditional board may not have to use the small traces, vias, and component pad spacing that was used on the DFT system, because there are less signals to route. Furthermore, the system busses (address, data, and control) are unpartitioned so there are more places that the components can reach these busses.

The PWB measurements are shown with respect to bottom left corner because that is the corner from which SCHEMA-PCB takes its measurements. An x,y coordinate system is used to show the exact PWB board outline and a "ruler" indicating a resolution of 0.001" (a mil) was placed next to the top and bottom side of the board to facilitate measurements. Edge connector pins, key spacing, and other important measurements are also indicated including a 1/4" component border denoted by a dotted line.

Table 33. PWB Artwork Specifications.

PWB Artwork	Size (mils)
Dimensions	9750 x 6000
Number of Layers	2 (component and noncomponent)
Signal Width	10 mils
Air-Gap Spacing	10 mils
Vcc/GND Width	50 mils (except on the board edges--the top is 150 mils, and bottom is 75 mils)
+12/-12 Width	30 mils
Via Drill Hole Diameter	28 mils
Via Pad Diameter	38 mils
Pad Drill Hole Diameter	35 mils
Pad Diameter	50 mils

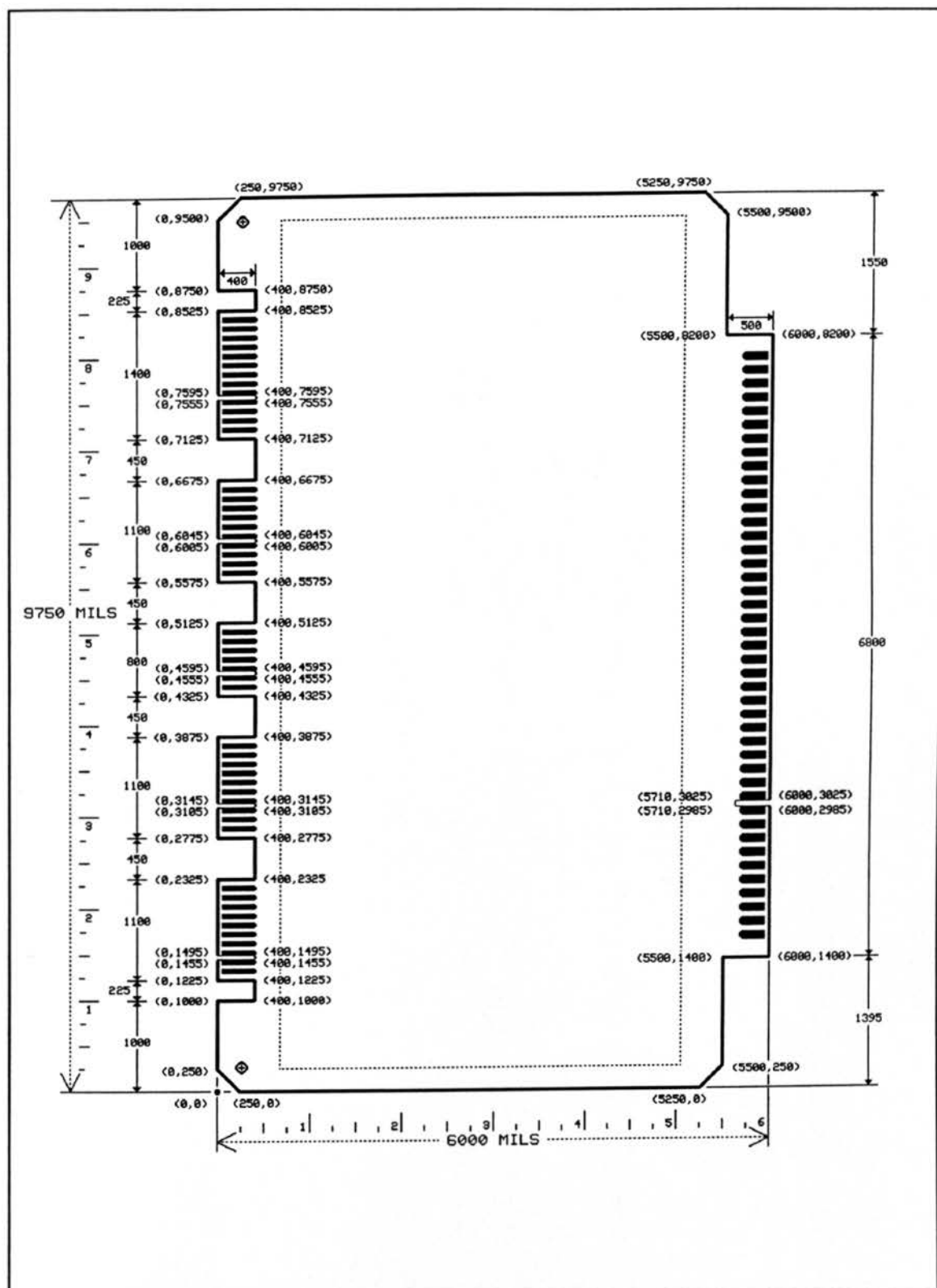
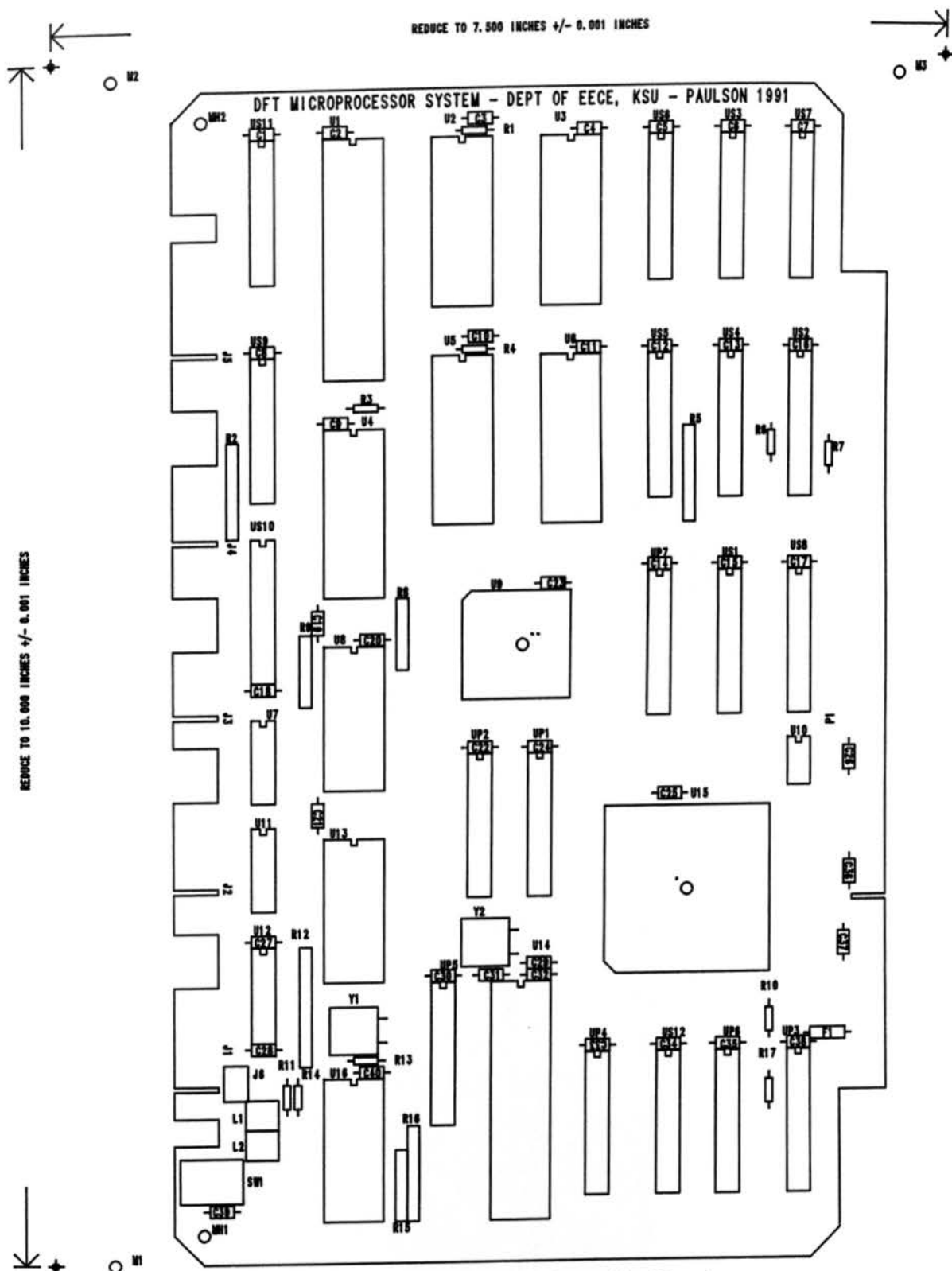


Figure 30. PWB Dimensions in mils (thousands of an inch).



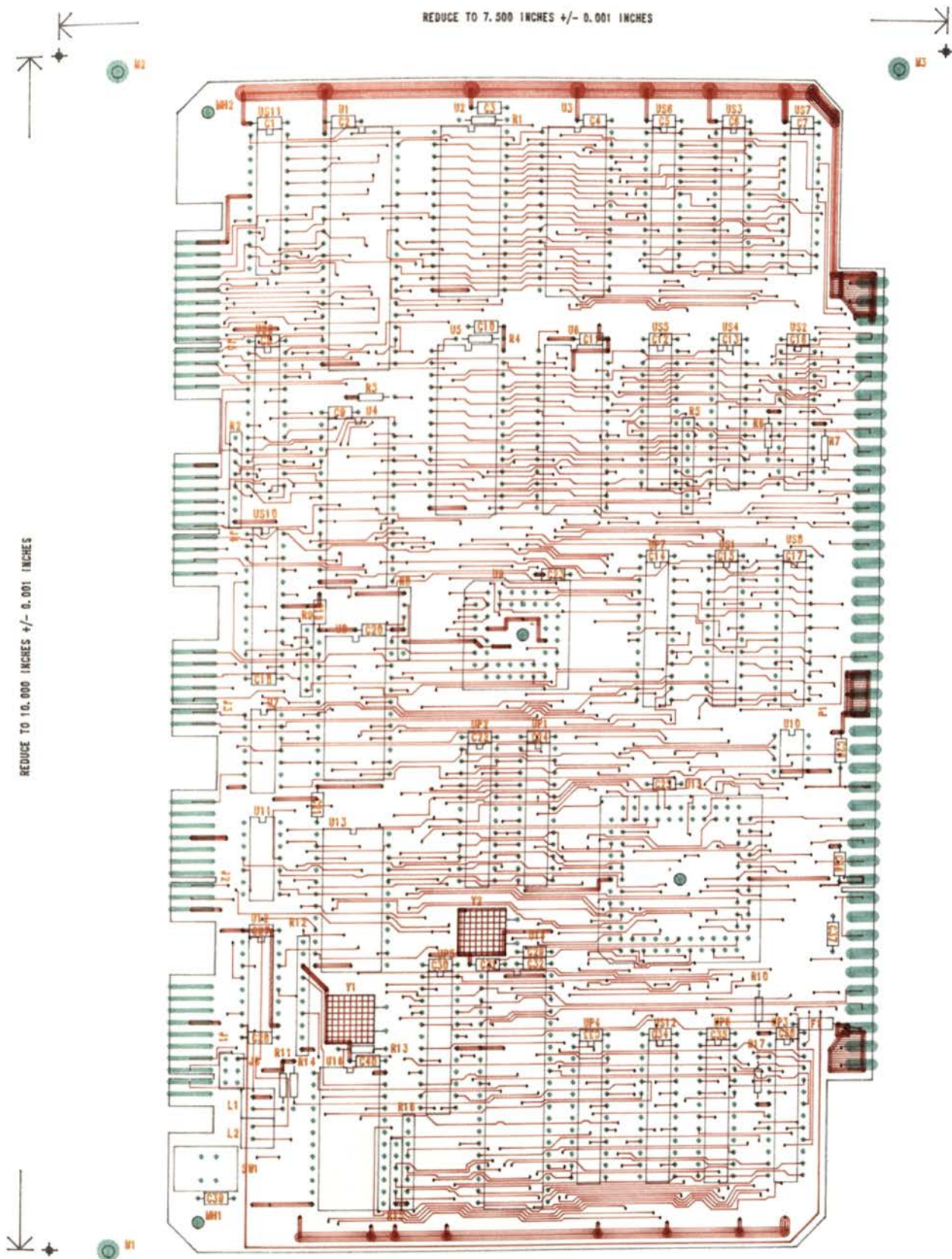


Figure 32. DFT System PWB Artwork, Layer 1 (Component Side).

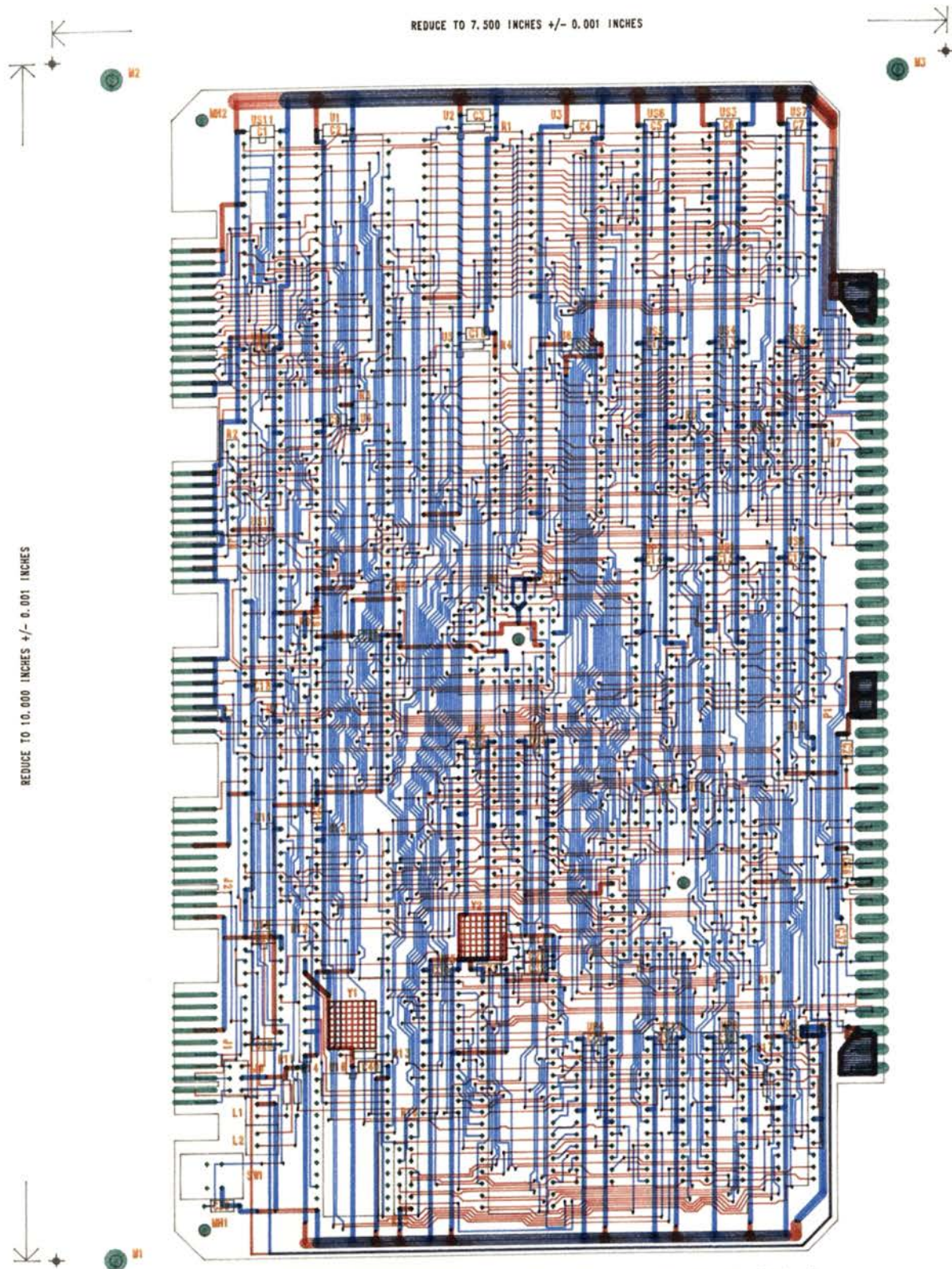


Figure 34. DFT System PWB Artwork, All Features (Both Sides).

Appendix I: SCHEMA II+ Library Instructions

This appendix describes the steps necessary to add a device to a SCHEMA II+ library. Letters in CAPS indicate a screen prompt, important point, or a Schema command. Primary and secondary commands are listed together with a hyphen between them. For example, commands to draw an object in Schema would be listed as DRAW-OBJECT. This means one would first select the primary command DRAW, then the secondary command OBJECT. To facilitate the explanation of these instructions, a fictitious part will be constructed.

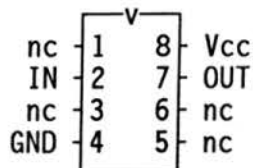
General Notes:

- These directions assume "some" prior SCHEMA II+ experience.
 - Items in **bold** are important! Library instructions will be shown in a monospace font and important notes will be displayed in this font.
 - Important keys are shown in brackets like <enter>, <esc> or <space>.
 - Some instructions require C GRID (cursor grid) to be on and some require C GRID to be off. These instances will be pointed out.
 - When the exact size of the device you're creating is important (such as in a board layout), the V GRID (visual dot grid) on the screen can help. **Dots** are placed **200 mil apart** (i.e. 200 thousandths of an inch apart).
 - In places where the use of a mouse is specified, the cursor keys on the keyboard could be used for **finer resolution**. See the primary command KEYS in the Schema manual for more information. (The cursor coordinate window can also help with placement.)
 - The steps are numbered from 0 to 16 (step 0 is optional). Each step may have several sub-steps.
 - Computer Engineers are allowed to start all numbering at zero.
0. (Optional step) Create a NEW library.
- a. From the Schema Manager, cursor down to SCHLIB and press <enter>.
 - b. Now cursor down to CREATE LIBRARY 2 and press <enter>.
 - c. SCHLIB prompts you with "Enter Size of Library to be created - S, M, or L", respond with S <enter>.

- d. SCHLIB then prompts with "Enter Library name with Directory and Extension (.LBR)", respond with \LIB\your_library_name.LBR then <enter>.
- e. Press <esc>, then <enter> to get back into the Schema Manager.

1. Determine the pinout of the device (from databook).
example pinout in databook:

74xxx - QUAD NOTHING GATE



2. Decide where pins should go on "your device" (flow-through design). A "functional block diagram" is usually provided in the device manufacturer's databook--this can prove invaluable here.

example:



3. To add a part to a library, you must first get to the OBJECT EDITOR. After getting into Schema draw (SCHDRAW), select FILE-ED OBJ, then press <enter>.
4. Determine the chip size (in Schema) of the device to be added. Be sure to leave enough room for text inside the rectangle (chip) describing the pins. Also, adjacent functional pin groups can easily be identified later if they are separated by more than 200 mil.
Note: Power and ground will **not** be placed inside the chip, so **do not** leave room inside the chip for these pins. (See an example IC if you are unsure about this step.)
 - a. Make sure C GRID is on (cursor grid is on initially when you enter EDIT OBJECT). The cursor should "jump" or move in discrete increments between the dots on the screen. F10 toggles the cursor grid on or off.
 - b. To make the chip outline select BODY-LINE.
 - c. Now use the mouse to draw a chip outline.

example:



5. Add pins outside the chip.
 - a. The C GRID should still be on.
 - b. Select BODY-HDR (body header).
 - c. Use the mouse to add pins to the chip. Leave at least 200 mil between adjacent pins.

Note: Reselecting BODY-HDR produces other types of pins (headers) that can be used—including vertical pins.

example:



6. Add text for pins. The diagram in Figure 35 shows an example of BODY-TEXT placement. This figure is a small portion of the top left-hand corner of a **completed** device whose three-stated pin number 23 is called A7. Ignore the text outside the chip outline for now--it will be explained in a later step.
 - a. Make sure C GRID is off (F10 toggles the cursor grid on or off). The cursor movement should now be smooth and continuous.
 - b. Select BODY-TEXT.
 - c. Type in the appropriate pin name. (Active-low signals can be constructed by placing a BODY-LINE over the top of the text later.)
 - d. Use the mouse to move the text next to the pin inside the chip.
 - e. Repeat steps b through d until finished with all pins.

example:

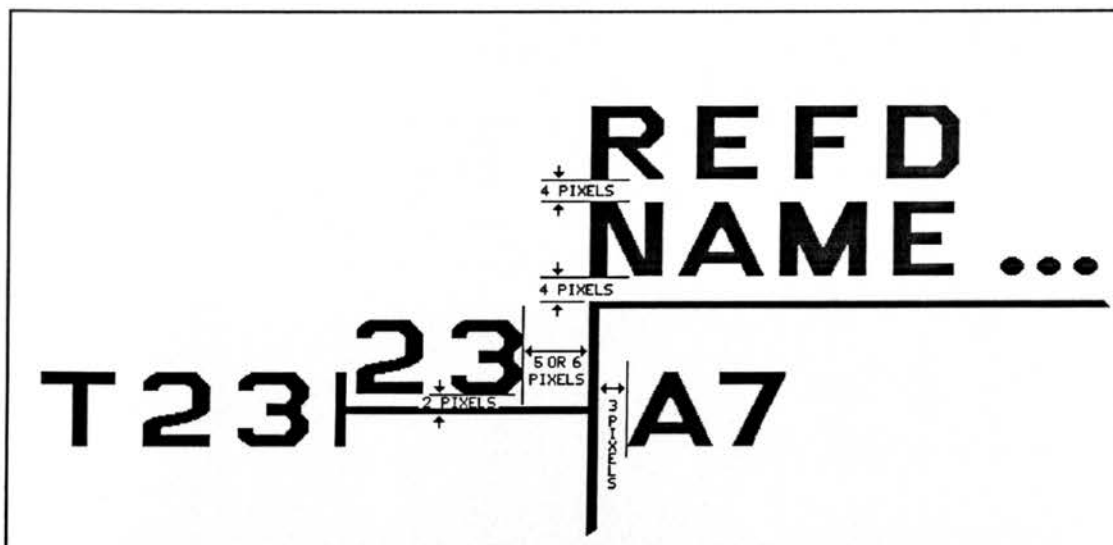
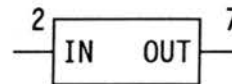


Figure 35. Suggested Placement of SCHEMA II+ Library Part Text.



7. Add pin numbers (outside the chip next to the pins).
 - a. Once again make sure C GRID is off and refer to Figure 35 on page 200 for placement.
 - b. Select BODY-TEXT again.
 - c. Type in the appropriate pin number.
 - d. Use the mouse to move the number next to the outside of the chip. (Numbers should be right justified.)
 - e. Repeat steps b through d until finished with all pins.

example:



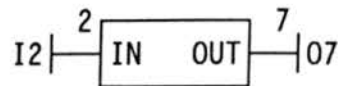
8. Add pin definitions such as input, output, bidirectional, etc. to the device. This is necessary for the Schema Post-Processor (SCHPOST). A "signal description" is usually provided in the device manufacturer's databook--this can prove invaluable here.
 - a. Now make sure that C GRID is on.
 - b. Select BODY-DF PI.
 - c. Reselecting BODY-DF PI changes the label (box) with respect to the pin (i.e. ■- or -■).
 - d. Make sure that ■- is selected for labels on the left-side or top of the chip, and -■ is selected for labels on the right-side or bottom of the chip. (Vertical labels are created by selecting BODY-DF PI and entering "<space>" prior to the pin definition.)
 - e. A pin definition is designated by a specific pin letter, followed by a pin number.

The following are legal pin definition letters:

I	Input
O	Output
T	Three-state Output (for data/addr buses, etc.)
B	Bidirectional
C	Open Collector
@	Absolute value
X	Don't Care

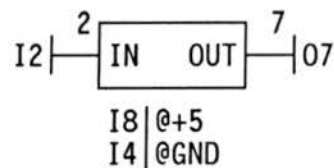
- f. Type in the appropriate **uppercase** pin definition letter and number (for example I2), then <enter>.
- g. Use the mouse to move the pin definition (I2|) next to the pin.
- h. Repeat steps b through g until finished with all pins.

example:



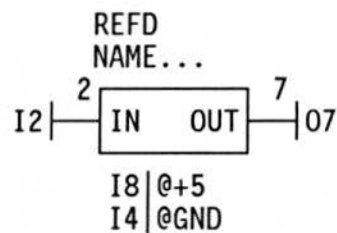
9. Add power and ground to the chip.
 - a. C GRID should **still be on**.
 - b. Select BODY-DF PI again.
 - c. Make sure that ■- is selected.
 - d. Type in (**uppercase I**) Ix where x is the pin number of a power input, (for example I8).
 - e. Use the mouse to move the pin definition (I8|) below the chip.
 - f. Reselect BODY-DF PI to select -■.
 - g. Type in @+x where x is your power supply value (for example @+5 for a 5 volt supply or GND for ground).
 - h. Use the mouse to **overlap the vertical bar** with the first vertical bar containing the Ix information.
 - i. Repeat steps b through h until finished with all power supply pins.

example:



10. Add the device name and reference designator (U1, U2, etc). Once again, refer to Figure 35 for optimum placement of BODY-PARM N and BODY-PARM R.
 - a. Now make sure that C GRID is off.
 - b. Select BODY-PARM N (not BODY-PARM R). (Reselecting BODY-PARM cycles through BODY-PARM N, BODY-PARM R, BODY-PARM 1, BODY-PARM 2, BODY PARM 3, BODY-PARM 4 and back to BODY-PARM N again.)
 - c. Type in NAME... (**exactly as shown "NAME..."**)
 - d. Use the mouse to move "NAME..." just above the device (lined up on the left).
 - e. Select BODY-PARM R (not BODY-PARM N).
 - f. Type in REFD (**exactly as shown "REFD"**)
 - g. Use the mouse to move "REFD" just above the "NAME..." text.

example:



11. Save the design (called a BODY OBJECT design).
 - a. Select LIBR-SAVE
 - b. If you want the part in a particular library, then type N= where N is the library number you chose prior to typing the device name (see next step).
 - c. Type in the device name with an orientation and (optional) component name (-1A indicates the first orientation of the first gate or component, the next orientation of the same part would be -2A). For example, if library 8 is the library I want the device saved in, I enter 8=74XXX-1A for our 74XXX design).
 - d. To determine which library number is associated with which library select CONFIG-LIBR 1. This will show you which library is library 1. Reselecting CONFIG-LIBR will show you the next library until you reach library 9.
12. Clear OBJECT EDITOR memory.
 - a. Be sure to save (step 11) the device before entering the next command.
 - b. Select LIBR-NEW. This clears the screen and erases the schematic memory.
13. You have created an BODY OBJECT, now create a NAME OBJECT.
 - a. Make sure that C GRID is on.
 - b. Select NAME-DESC 1
 - c. Enter a part number (e.g. 436-3848-602)
 - d. Use the mouse to place the part number on the screen.
 - e. Select NAME-DESC 2
 - f. Enter a description for the part number.
 - g. Use the mouse to place the description below the first description.

example:

436-3848-602
74LSXXX - QUAD NOTHING GATE

14. Now, associate the BODY OBJECT with the NAME OBJECT.
 - a. Select NAME-BODY
 - b. Enter the name you used in step 11 without the orientation and library number (e.g. 74XXX).
 - g. Use the mouse to place the text below the description.

example:

436-3848-602
74LSXXX - QUAD NOTHING GATE
XXXX

15. Save the device.
 - a. Select LIBR-SAVE
 - b. Enter the actual part name without the orientation but with the library number (e.g. 8=74XXX).
16. Now you're finished. You can call up the device in SCHDRAW by entering DRAW-OBJECT and the name of the device.

**DESIGNING TESTABILITY INTO
AN EXISTING MICROPROCESSOR BOARD**

by

MICHAEL CRAIG PAULSON

B.S., Electrical Engineering, Kansas State University, 1988

B.S., Computer Engineering, Kansas State University, 1989

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Electrical Engineering

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1992

Abstract

This thesis presents the hardware implementation of a microprocessor system built according to the microprocessor-controlled Built-In Self-Test (BIST) techniques as presented by Gordon [Gordon 1991a]. It covers issues relating to the isolation of the edge connector and secondary board areas, to the use and function of a field-programmable gate array (Logic Cell Array)TM, to the implementation of electronic wrap-arounds, and to the general design of the software to support the testability features. In addition, it covers the general use of the ANSI/IEEE Std. 1149.1 test bus for diagnosis, isolation, and board-level partitioning.