

F8680 PC/CHIP™

System Design Guide

PC/CHIP™

CHIPS®



F8680 PC/CHIP™

System Design Guide

Copyright Notice

Software Copyright © 1991, Chips and Technologies, Inc.

Manual Copyright © 1991, Chips and Technologies, Inc.

All Rights Reserved.

Printed in U.S.A.

Trademark Acknowledgement

CHIPS®, and the CHIPS logotype are registered trademarks of Chips and Technologies, Inc.
PC/CHIP™, SMARTMAP™, SuperState™, Virtual I/O™, Virtual Interrupts™, and Visual Map™
are trademarks of Chips and Technologies, Inc.

AT&T® is a registered trademark of AT&T Bell Laboratories.

Centronics® is a registered trademark of Centronics Corp.

IBM®, PC/AT®, and PS/2® are registered trademarks of International Business Machines, Inc.

Maxim® is a registered trademark of Maxim Integrated Products, Inc.

MicroTouch® is a registered trademark of MicroTouch Systems, Inc.

National® is a registered trademark of National Semiconductor Corp.

Disclaimer

This manual is copyrighted by Chips and Technologies, Inc. You may not reproduce, transmit, transcribe, store in a retrieval system, or translate into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, any part of this publication without express written permission of Chips and Technologies, Inc.

The information contained in this document is being issued in advance of the production cycle for the device(s). The parameters for the device(s) may change before final production.

Chips and Technologies, Inc. makes no representations or warranties regarding the contents of this manual. We reserve the right to revise the manual or make changes in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or changes.

The information contained in this manual is provided for general use by our customers. Our customers should be aware that the personal computer field is the subject of many patents. Our customers should ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Chips and Technologies, Inc. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

Restricted Rights and Limitations

Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.277-7013

Chips and Technologies, Inc.
3050 Zanker Road
San Jose, California 95134
Phone: 408-434-0600

Preface

The *F8680 PC/CHIP™ System Design Guide* provides useful information for incorporating the F8680 microchip in the hardware design of a computer system. This manual is intended for use by engineers who need to understand the features and signal interface of the F8680 chip.

Design Guide Organization

This design guide is organized as 11 chapters and 3 appendixes. The first chapter introduces the chip architecture, the second chapter provides sample configurations using the F8680 chip, and the remaining chapters deal with various aspects of incorporating the chip into a design.

Chapter 1 provides an overview of the F8680 chip, highlighting those features of the chip which are new to PC-on-a-chip technology. It also lists available chip resources, and indicates general design considerations to be observed.

Chapter 2 gives examples of how to use the F8680 chip in various practical designs, and indicates where the specific details for each example will be found in the design guide.

Chapter 3 describes the memory interface of the F8680 chip.

Chapter 4 describes the display interface of the F8680 chip and provides alternative connections for external display subsystems.

Chapter 5 describes the power management circuitry of the F8680 chip.

Chapter 6 describes the keyboard interface options and mouse connection options available on the F8680 chip.

Chapter 7 describes the connection of a floppy disk controller circuit to the F8680 chip.

Chapter 8 describes the connection of a hard disk interface circuit to the F8680 chip.

Chapter 9 describes the use of the internal serial port of the F8680 chip and discusses the options for incorporating a parallel port into system design.

Chapter 10 describes the facilities available for connection to external peripherals through the XT bus.

Chapter 11 describes special applications, such as connecting a display overlay or a digital-to-analog converter.

Appendix A summarizes the processor architecture of the F8680 chip.

The manual also provides an index of major topics.

Conventions

This manual uses certain conventions in describing signals and programming concepts, such as the following:

- Reference is made to CREG and SDATA. CREG indicates an index to the configuration register set of the F8680 chip. SDATA indicates an index to the status data register set of the F8680 chip. The CREGs and SDATAs are summarized in the *F8680 PC/CHIP™ Data Sheet*.
- A group of signals that act as a bus is specified with a colon, e.g., address lines ADR0 to ADR25 are referred to as ADR25:0 when they act as a single bus. A group of signals that does not act as a bus is specified with a hyphen, e.g., interrupt request lines IRQ0 to IRQ7 are referred to as IRQ0-7. This same convention applies to programming bits that refer to signal groups.
- A signal that is considered active when at logic 0 is specified with a trailing asterisk, e.g., the write enable line is specified as WE0*. The asterisk is not used to refer to a footnote in this manual.
- Numeric values must be assumed to be in hexadecimal. The letter h is often used to specify a hex base when the reference is ambiguous.
- I/O addresses are always specified as three hexadecimal digits, from 000 to 3FF. Memory addresses are absolute values (not offsets) in hex, unless otherwise specified.

References

This manual should be used in conjunction with the following related documents:

- *F8680 PC/CHIP™ Programmer's Reference Manual*
- *F8680 PC/CHIP™ Data Sheet*

The following companies supply products that are referenced in this publication.

Maxim Integrated Products, Inc.
120 San Gabriel Drive
Sunnyvale, California 94086
1-408-737-7600

Micrel Semiconductor
560 Oakmead Parkway
Sunnyvale, California 94086
1-408-245-2500

MicroTouch Systems, Inc.
55 Jonspin Road
Wilmington, Massachusetts 01887
1-800-866-6873

Contents

Preface	iii
Design Guide Organization	iii
Conventions	iv
References	v

CHAPTER 1

System Overview	1-1
Hardware Design	1-3
Architectural Enhancements	1-3
SuperState R Mode	1-4
Virtual I/O Feature	1-5
Virtual Interrupts Feature	1-5
Hardware Description	1-6
CPU Architecture	1-6
Memory Subsystem	1-8
CGA-Compatible Graphics Controller	1-9
XT Subsystem	1-9
UART	1-9
System Design Resources	1-10
Hardware Triggers to Switch to SuperState R Mode	1-10
Waveform Generation	1-10
Chip Select	1-11
Processor Status	1-11
Graphics Control	1-11
PCMCIA Interface	1-11
General Design Considerations	1-12
Power Considerations	1-12
Signal Characteristics	1-13
EMI Considerations	1-13

CHAPTER 2

Design Applications 2-1
Palmtop Application 2-2
Laptop Application 2-4
Portable Application 2-6
Desktop Application 2-8
Pen-Based Application 2-10
Embedded Controller Application 2-12

CHAPTER 3

Memory 3-1
Memory Operational Theory 3-1
 Bank Switching Logic 3-2
 Memory Mapping Logic 3-3
How to Approach Memory Design 3-5
DRAM 3-6
 Signals 3-6
 Memory Address Multiplexing 3-8
 Page Mode Operation 3-9
 Refresh 3-9
 Interface to Common Types 3-9
SRAM/PSRAM 3-13
 Signals 3-13
 Refresh for PSRAM 3-14
 Interface to Common Types 3-14
ROM 3-16
 BIOS ROM 3-16
 Applications in ROM 3-16
 Signals 3-16
 Interface to 8-bit ROM Banks 3-18
PCMCIA-Standard Interface 3-18
 Signals 3-19
 Chip Support for PCMCIA 3-22
 PCMCIA Memory Performance 3-26
 Medium Types 3-26
 Circuit for Programmable Cards 3-27
 Using the Interface as a General Purpose I/O Port 3-27

CHAPTER 4

Display 4-1

SRAM Interface 4-1

LCD Panel Interface 4-2

 Alternative Dot Clock Input on PS4 4-3

 ACDCLK Output Option on PS2 4-4

 Register Programming 4-5

 Controlling the Backlight 4-6

CRT Interface 4-7

Connecting an External Display Controller 4-8

 Disabling the Internal Controller 4-8

 Connecting the 82C455 Low-power VGA Controller 4-8

 Connecting the 82C450 Super VGA Controller 4-10

CHAPTER 5

Power Management 5-1

Power Control Issues 5-1

 Software Power-Off Control 5-2

 Power-On Clock Comparator 5-2

 Power Control State Machine 5-3

Connections to the PWRUP Input 5-3

Using OSCPW to Drive System Oscillators 5-5

Signals Available for Control/Status 5-6

 Programmable Pins 5-6

Controlling Power to Peripheral Devices 5-8

Monitoring Battery Voltage 5-9

Power-Saver Mode 5-11

CHAPTER 6

Keyboard and Mouse 6-1

Connections to the XT Keyboard 6-1

Direct Scanning of an Integral Keyboard 6-2

 Keyboard Scanning Principles 6-4

 Other Keyboard Emulation Issues 6-5

AT Keyboard Emulation Considerations 6-6

Providing for a Mouse 6-7

CHAPTER 7

Floppy Disk Drives 7-1
Connecting the 82C765 Floppy Disk Controller 7-1
Floppy Disk Controller Registers 7-3
Connecting the 82C710 Universal Peripheral Controller 7-3

CHAPTER 8

Hard Disk Drives 8-1
XT-Type Hard Disk Drive Signals 8-1
XT-Type Hard Disk Drive Registers 8-3
AT-Type Hard Disk Drive Signals 8-4
AT-Type Hard Disk Drive Registers 8-6

CHAPTER 9

Serial and Parallel Ports 9-1
RS232-C Signals 9-1
 Connecting a Modem Module 9-3
 Connecting Line Drivers 9-4
Parallel Port Signals 9-5
Parallel Interface 9-6
 SuperState R Support for Parallel Port 9-8
 Connecting the CHIPS 82C605 Multifunction Controller 9-10
 Connecting the CHIPS 82C710 Universal Peripheral Controller 9-11

CHAPTER 10

XT Subsystem 10-1
Generating RESETDRV for the XT Bus 10-3
Connecting the Interrupt Request (IRQ) Pins 10-4
Emulating Extra IRQ Lines Through the Programmable Pins 10-4
Connecting the DREQ/DACK Pins 10-5
Driving Peripheral Devices on the XT Bus 10-6
Connecting a Speaker 10-7

CHAPTER 11

Special Applications 11-1
 Incorporating a Display Overlay 11-1
 Connecting the Overlay 11-2
 Monitoring for Stylus Contact 11-3
 Monitoring a Control Panel 11-3
 Connecting a Digital-to-Analog Converter 11-4
 Connecting an Analog-to-Digital Converter 11-5
 Emulating the Real-Time Clock 11-6

APPENDIX A

Processor Architecture A-1
 Processor Pipeline A-1
 Performance Reduction A-3

INDEX

Representatives

List of Figures

Figure 1-1. F8680 Chip Architecture—Block Diagram 1-2

Figure 1-2. F8680 Processing Registers 1-7

Figure 1-3. Supply Voltage vs. Operational Speed 1-12

Figure 2-1. Palmtop Application 2-3

Figure 2-2. Laptop Application 2-5

Figure 2-3. Portable Application 2-7

Figure 2-4. Desktop Application 2-9

Figure 2-5. Pen-based Application 2-11

Figure 2-6. Embedded Controller Application 2-13

Figure 3-1. Bank Switching Logic 3-2

Figure 3-2. Memory Mapping Logic 3-4

Figure 3-3. Interface to 4-bit DRAM 3-10

Figure 3-4. Interface to 8-bit DRAM 3-11

Figure 3-5. Interface to 1-bit DRAM 3-12

Figure 3-6. Interface to SRAM 3-15

Figure 3-7. Interface to 8-bit ROM 3-18

Figure 3-8. Single Card Slot Interface 3-23

Figure 3-9. Dual Card Slot Interface 3-25

Figure 3-10. Controlling Program Voltage V_{pp} to the Card Slot 3-27

Figure 4-1. Connection of Display SRAM 4-1

Figure 4-2. Display Controller Signals to the LCD Panel 4-2

Figure 4-3. Alternative Dot Clock, AC Drive Clock Connections 4-3

Figure 4-4. Power Control Using a Programmable Pin 4-6

Figure 4-5. Display Controller Signals to the CRT 4-7

Figure 4-6. Connecting the 82C455 VGA Controller 4-9

Figure 4-7. Connecting the 82C450 VGA Controller 4-10

Figure 5-1. Connecting PWRUP to a Toggle Switch 5-4

Figure 5-2. Connecting PWRUP to a Momentary Switch 5-4

Figure 5-3. Using OSCPW to Control Power to the Oscillators 5-5

Figure 5-4. Connecting a Power Control Device 5-8

Figure 5-5. Simple “Battery Low” Indicator Circuit 5-9

Figure 5-6. Two-Level “Battery Low” Indicator Circuit 5-10

Figure 5-7. Battery Voltage Monitor Circuit With A/D Converter 5-11

Figure 6-1. Connections to the XT Keyboard 6-2

Figure 6-2. Connections to Keyboard Matrix for Direct Scanning 6-3

Figure 7-1. Floppy Disk Drive Interface Signals 7-2

Figure 8-1. Connections to an XT-Type IDE Hard Disk Drive 8-2

Figure 8-2. Connections to an AT-Type IDE Hard Disk Drive 8-5

Figure 9-1. RS232-C Interface Signals 9-2

Figure 9-2. Connecting the Modem Module 9-3

Figure 9-3. Line Drivers and Receivers for External RS232-C Interface .. 9-4

Figure 9-4. Centronics Parallel Port Signals 9-5

Figure 9-5. Data Register—I/O Address 378/278 (W/R) 9-6

Figure 9-6. Printer Status Register—I/O Address 379/279 (R) 9-7

Figure 9-7. Printer Control Register—I/O Address 37A/27A (W/R) 9-8

Figure 9-8. Parallel Port Implementation With Latches 9-9

Figure 9-9. Connections to the 82C605 Multifunction Controller 9-10

Figure 9-10. Connections to the 82C710 Universal Peripheral Controller .. 9-11

Figure 10-1. XT Bus Card-Edge Connector 10-3

Figure 10-2. Buffers for the XT Bus 10-6

Figure 10-3. Driver Circuits for Sounding Devices 10-7

Figure 11-1. Connecting an Overlay 11-2

Figure 11-2. Incorporating the D/A Converter 11-4

Figure 11-3. Incorporating the A/D Converter 11-5

Figure A-1. Processor Pipeline A-2

List of Tables

Table 1-1.	System Memory Usage	1-8
Table 1-2.	Output Pin Drive Capability	1-13
Table 3-1.	Bank Switch Register Ranges	3-3
Table 3-2.	Pin Assignments—DRAM Interface Signals	3-7
Table 3-3.	Address Multiplexing	3-8
Table 3-4.	Pin Assignments—SRAM Interface Signals	3-13
Table 3-5.	Address Bus Connections for Various SRAM Device Capacities	3-15
Table 3-6.	Pin Assignments—ROM/PROM Interface Signals	3-17
Table 3-7.	PCMCIA Signal Name Equivalents	3-19
Table 3-8.	Pin Assignments—PCMCIA Interface Signals	3-20
Table 5-1.	Programmable Pin Functions	5-6
Table 6-1.	Possible Address Ranges for Keyboard Scan Decoding	6-5
Table 6-2.	Address Sequence to Pulse Each ROW Line Low	6-6
Table 7-1.	Register Names and I/O Addresses	7-3
Table 8-1.	XT-Type IDE Hard Disk Drive Interface Signals	8-1
Table 8-2.	XT-Type Hard Disk Drive Registers	8-3
Table 8-3.	AT-Type IDE Hard Disk Drive Interface Signals	8-4
Table 8-4.	AT-Type Hard Disk Drive Registers	8-6
Table 10-1.	XT Bus Signals	10-1
Table 10-2.	XT-Compatible Interrupt Assignments	10-4
Table 10-3.	AT-Compatible Assignment for Second Interrupt Controller	10-4
Table 10-4.	XT-compatible DMA Request Line Assignment	10-5

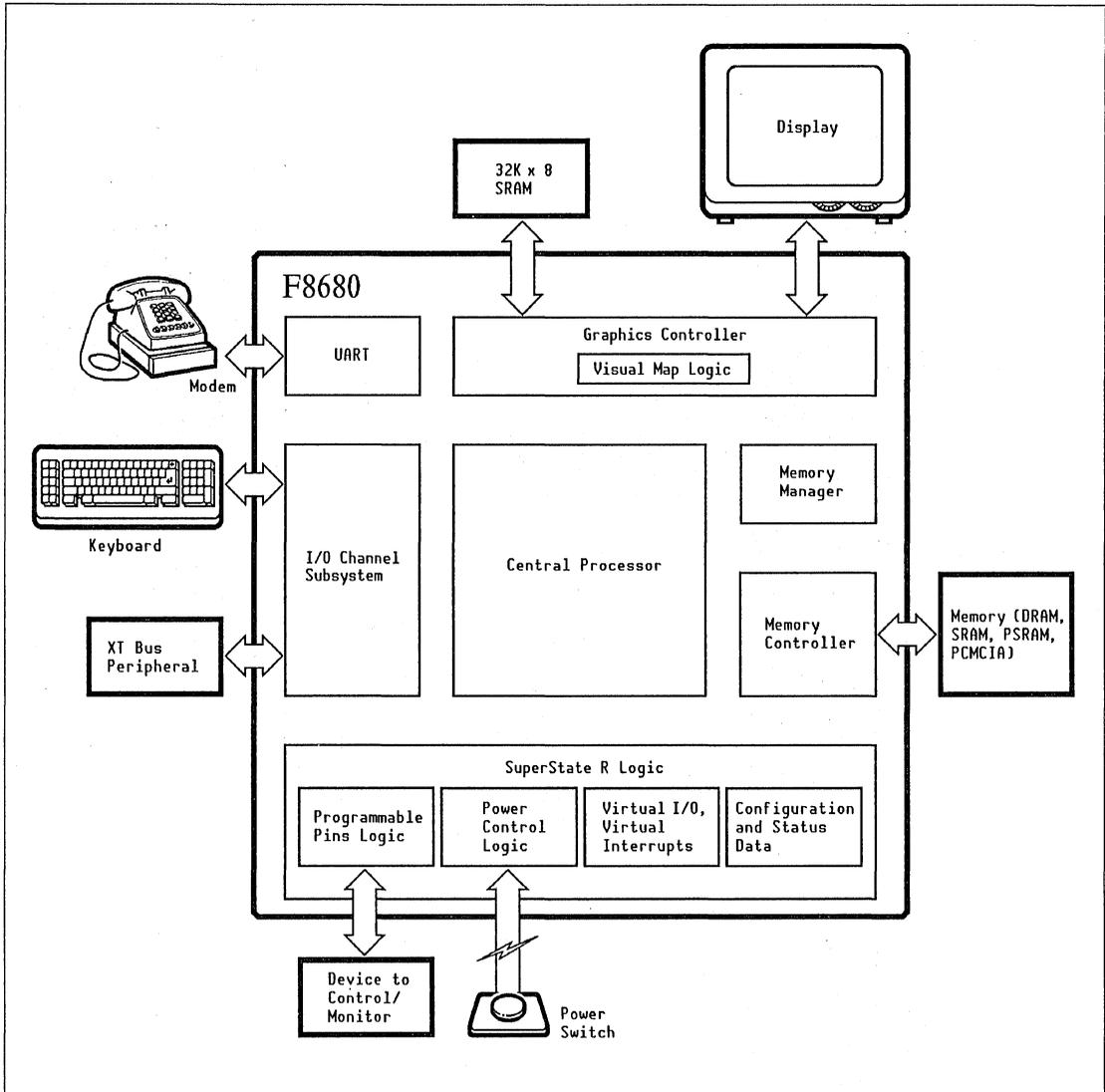
System Overview

The F8680 PC/CHIP™ microchip implements the following subsystems of an XT-compatible computer:

- 8086-compatible central processor with enhancements to provide a protected management mode
- 8254-compatible timer
- 8259-compatible interrupt controller
- Memory controller with bank-switching logic
- DMA emulation mechanism that is 8237-compatible
- Keyboard interface circuit (XT protocol)
- CGA-compatible graphics controller
- 16C450-compatible UART.

Figure 1-1 illustrates the relationship of these subsystems in the F8680 chip design. The F8680 chip architecture and features give you, the design engineer, freedom from dealing with the routine details of XT-compatible architecture. You can spend your design time where it counts: developing a powerful system by selecting the memory, storage devices, graphics display, and I/O peripherals that will be most useful for your application.

Figure 1-1. F8680 Chip Architecture—Block Diagram



Hardware Design

The design of the F8680 chip is tailored for use in low-cost DOS-compatible systems and expansion cards, handheld and portable computers, and embedded controller applications. From the perspective of hardware design, this results in a chip with the following features:

- Provides fully static operation, allowing a near-complete power-down with no loss of data.
- Integrates popular peripheral devices such as a graphics controller, UART, and continuous time-of-day clock.
- Activates oscillator power and enables internal clocks in the proper sequence at power-up/power-down times to eliminate the need for external logic.
- Provides a practical mechanism for bank-switching memory, allowing EMS memory addressing and memory card addressing with no external mapping logic.
- Eliminates the need for external decode logic by providing up to four programmable chip select decodes.
- Can trap all accesses to nonexistent hardware and simulate the hardware in software.
- Provides an extremely flexible memory mapping mechanism to make the most efficient use of virtually any memory configuration.
- Allows hardware configuration to be performed under software control, maximizing hardware design flexibility.

These features relieve you of the burden of adapting your hardware design to the chip. Instead, you program the F8680 chip so that it adapts to your hardware design.

Architectural Enhancements

The F8680 chip provides functionality comparable to an XT-compatible system with an 8086 processor but with better performance. Above this, the F8680 chip provides independent management features through hardware and microcode enhancements. The enhancement schemes involve the following new concepts:

- SuperState™ R supervisory mode is used to execute system management code in an environment that is protected from applications.
- Virtual I/O™ mechanism intercepts I/O operations for monitoring and servicing by SuperState R code.

- Virtual Interrupts™ mechanism intercepts interrupts for monitoring and servicing by SuperState R code.
- Visual Map™ contrast control facility guarantees that applications written for a color display will be viewable on a monochrome LCD panel.

These features are discussed briefly in the following sections of this chapter. Refer to the *F8680 PC/CHIP Programmer's Reference Manual* for more complete details of these features.

SuperState R Mode

The SuperState R operating environment is separate from the normal operating environment of DOS and the system BIOS. If your hardware design includes any extensions to the XT architecture (power control circuitry, PCMCIA memory card slots, etc.), these extensions can be managed in SuperState R mode and thus are transparent to normal DOS and BIOS operations.

Entering SuperState R Mode

The F8680 chip is always in SuperState R mode after a reset or initial power-up. At that time, your initialization routines can program the F8680 chip to enter SuperState R mode on any of the following events:

- A hardware or software interrupt must be intercepted (Virtual Interrupts feature).
- An IN or OUT instruction has executed and a device on the I/O bus must be emulated or monitored (Virtual I/O feature).
- A specified period of time elapses.
- A DMA channel must be set up or auto-initialized.
- An external request made on one or more of the four programmable I/O pins on the F8680 chip must be serviced.
- A power-down request, generated by a status change on the power-up input pin, must be serviced.

Your routines can also disable or enable these events to cause a switch to SuperState R mode as needed during normal system operation.

Configuration Space

SuperState R mode provides a means of setting system hardware configuration through a new CPU instruction. Application programs cannot modify this information with any I/O or memory instruction. Moreover, the configuration registers do not occupy any of the system I/O address space.

SuperState R code can intercept and interpret other setup programming formats. This means that the standard setup and BIOS code you have used in previous designs can be used with virtually no modifications.

Virtual I/O Feature

The Virtual I/O feature, which is programmed and maintained while in SuperState R mode, allows I/O operations to each and every port to be monitored, redirected, emulated, or suppressed as needed.

The Virtual I/O feature can be used to emulate the operation of a device that is not actually present. When adequately comprehensive emulation code is used, applications cannot distinguish Virtual I/O operations from true I/O with a peripheral device.

Virtual Interrupts Feature

Any system interrupt, whether caused by a hardware IRQ or by a software INT instruction, can be trapped and examined by SuperState R code. Trapping occurs before any TSR programs or interrupt handlers see the interrupt.

Once trapped, the SuperState R code can substitute register values and pass the call back for normal interrupt handling, or it can emulate the interrupt handler itself and bypass the normal mechanism. Emulating the interrupt handler is especially useful when handling file access interrupts, for example, to emulate a hard disk in RAM.

Hardware Description

The F8680 hardware is intended to be XT-compatible on the interface level of operation. Integrating all subsystems into a single physical package also results in a substantial overall performance improvement over the standard XT. Therefore, in your hardware design you have only to take advantage of the simplicity of the F8680 PC/CHIP architecture. The following paragraphs describe the major hardware subsystems of the F8680 chip and illustrate this simplicity.

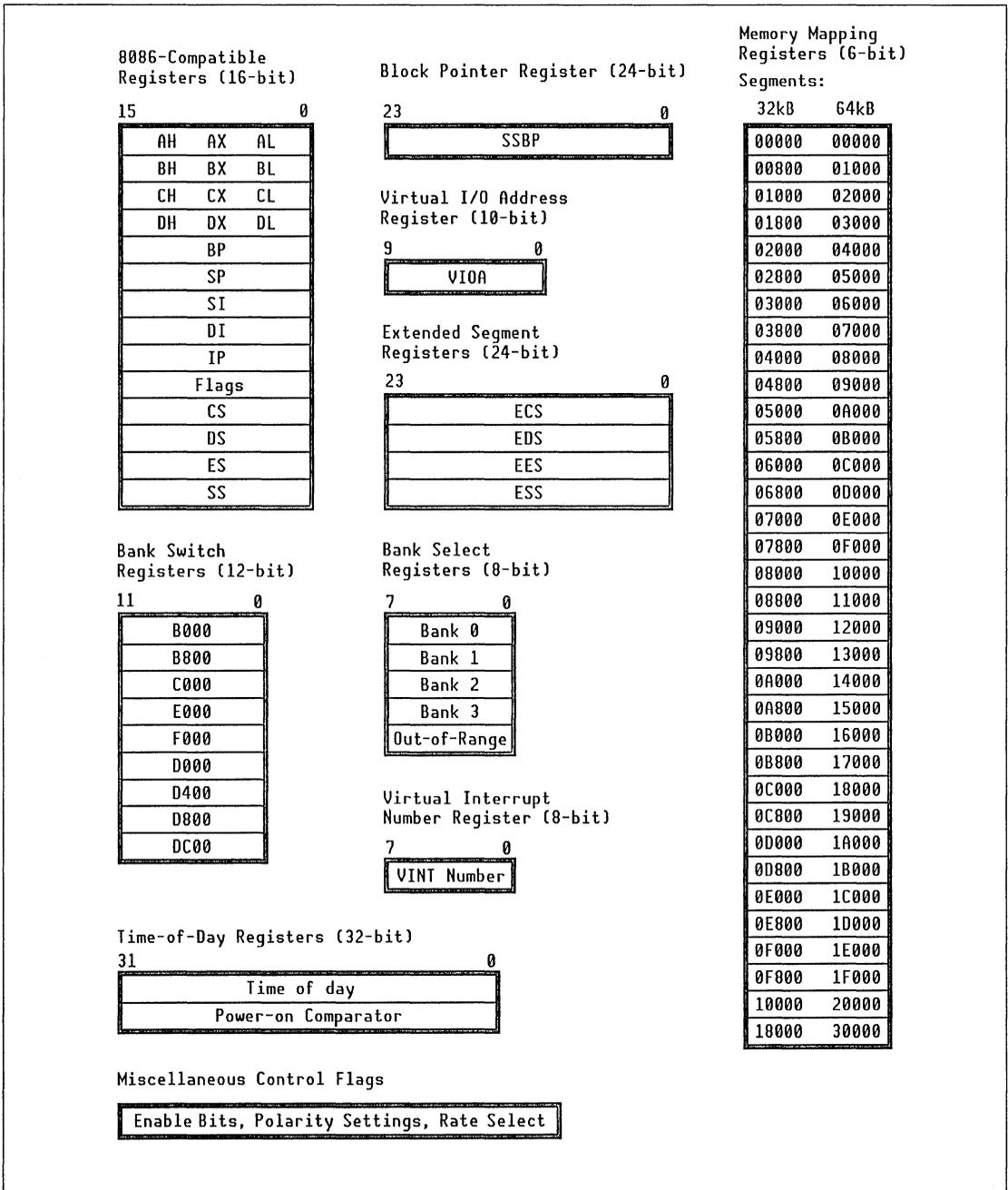
CPU Architecture

The central processing logic of the F8680 chip is compatible with the 8086 processor, including its true 16-bit data path. However, the F8680 chip has non multiplexed buses. Moreover, the F8680 chip generates addresses in a 64MB range, as opposed to the 1MB range of the 8086. Instruction fetch, decode, and execution are processed through a four-stage pipeline. Appendix A provides details of the pipeline operation.

The CPU executes the entire instruction set of the 8086 processor as well as the new SuperState R instruction set. Moreover, it incorporates 24-bit segment registers and several new registers to support SuperState R operation. Figure 1-2 shows the processing registers available in the F8680 chip. The peripheral device (interrupt controller, UART, etc.) registers are not shown.

Unlike the CPU of the XT, the F8680 central processor is given the added responsibility of performing DMA transfers. The logic provides the necessary DMA controller registers in hardware. DMA transfer instructions are executed at the microcode level. Refer to the *F8680 PC/CHIP Programmer's Reference Manual* for complete details on DMA operation. Note, however, that the DMA emulation mechanism will have no impact on your hardware design.

Figure 1-2. F8680 Processing Registers



Memory Subsystem

The F8680 chip provides an extremely versatile memory management and control system. The system can take advantage of every block of memory in your design, to a resolution of 32kB.

Memory Management

Mapping of CPU addresses to physical RAM is done in 32kB or 64kB segments, allowing great flexibility in utilizing the memory type you choose for your application. Once mapped, an independent mechanism allows segments to be bank-switched to provide EMS memory and a PCMCIA-standard memory card interface. The system directly manages a maximum of 4MB of RAM; it can address additional memory up to 32MB. The system memory usage is shown in Table 1-1.

Table 1-1. System Memory Usage

Memory Range	Size (kB)	Usage
0F0000-0FFFFFFF	64	BIOS ROM
0E0000-0EFFFFFF	64	Often used for PCMCIA memory card access or ROM applications
0D0000-0DFFFFFF	64	Often used for EMS memory page frames
0C0000-0CFFFFFF	64	Often used for ROM applications or PCMCIA memory card access
0B8000-0BFFFF	32	CGA graphics memory
0A0000-0A7FFF	96	Often used for alternate video controller
000000-09FFFF	640	MS-DOS and applications

Memory Controller

The memory controller generates cycles for dynamic RAM (DRAM), static RAM (SRAM), pseudo-SRAM (PSRAM), PCMCIA-standard memory, ROM, and the XT bus. It provides support for 32k x 8, 128k x 8, and 512k x 8 SRAM and PSRAM, as well as support for 256k x 1, 256k x 4, 512k x 8, 1M x 1, 1M x 4, and 4M x 1 DRAM. Page mode is supported with both byte and word requests.

The controller provides all timing and control signals to allow direct support for up to three banks of 8-bit or 16-bit memory. You can use both 8-bit and 16-bit banks in the same system.

CGA-Compatible Graphics Controller

The graphics controller supports both CRT and LCD panel displays with a fully CGA-compatible register set. It supports 80 x 25 and 40 x 25 text modes, as well as 640 pixel 2-color and 320 pixel 4-color graphics modes, all at 200 or 400 lines of resolution.

When driving a CRT, the graphics controller displays colors the same way as the IBM® Color Graphics Adapter (CGA). For monochrome panels the processing of the attributes is identical, but the resulting colors are translated to gray levels. Translation of up to 16 levels of gray is possible.

The Visual Map feature overcomes a problem that arises with monochrome LCD panels used with text mode applications written for a color display. Many controllers map colors to shades of gray, but colors that are close in intensity are barely distinguishable on a monochrome display. The problem is pronounced when text mode foreground and background colors are mapped to the same shade of gray: the text disappears.

The Visual Map feature programs each possible foreground and background color combination into a table as a specific combination of shades of gray. Each combination provides contrast that is closer to that of a color display than simple mapping techniques can achieve. See the *F8680 PC/CHIP Programmer's Reference Manual* for details of the Visual Map mechanism.

XT Subsystem

The standard XT subsystem is for the most part implemented in hardware. The interrupt controller is a hardware equivalent of the 8259A component and the timer is a hardware equivalent of the 8254 component for those functions available in an XT-compatible environment. The DMA controller is not actually present in hardware but is emulated by a combination of hardware, microcode, and software. From a hardware aspect, the XT subsystem is completely compatible with that of the XT computer.

UART

The F8680 chip provides a Universal Asynchronous Receiver/Transmitter (UART), an asynchronous serial communications element compatible with the National Semiconductor NS16C450 chip. You can assign the UART to respond as either COM1 or COM2, or you can disable the UART.

System Design Resources

The F8680 chip provides a wealth of resources to make your design job easier. But you may find that an even more important design aspect is the flexibility it gives you in assigning F8680 chip resources. The F8680 PC/CHIP architecture allows you to program the logic to provide only those features you need, so you can specifically tailor the F8680 chip to your application. Moreover, you need not end up with wasted logic that robs power and uses up valuable board space.

The following paragraphs provide an overview of the resources at your disposal for facilitating system design.

Hardware Triggers to Switch to SuperState R Mode

The SuperState R logic accepts switch requests on any of lines PS1, PS2, PS3, PS4, and PWRUP. The pins can be programmed to request a switch on transition to either a high or a low state, and can be dynamically reprogrammed to switch on the opposite sense. Thus, you can use either a momentary pulse or a steady-state signal to request a switch to SuperState R mode. Once in SuperState R mode, the code you provide can determine the pin(s) that caused the switch and monitor their instantaneous state(s). A status data (SDATA) register provides this information.

Note that the IOCHCK line on the XT bus is available to generate a non-maskable interrupt (NMI) as in the XT computer. Generally speaking, you should avoid using the NMI. Using a switch to SuperState R mode is preferable to using the NMI mechanism because you do not have to worry about operating system conflicts when SuperState R code services a switch request.

Waveform Generation

The four programmable pins PS1-PS4 plus the CARDB pin can be programmed for periodic waveform generation as follows:

- Square Wave Generators—256Hz, 512Hz, 16384Hz, 32768Hz
- Clock Pulse Generators—one half-clock-period pulse 32768 times a second, 2048 times a second, or 1024 times a second
- Frame pulse—the vertical frame interval divided by 2.

Chip Select

The four programmable pins PS1-PS4 can all be programmed to decode I/O reads, I/O writes, or both, to any I/O address or address range. This feature provides true chip select lines that you can use directly, with no additional decode logic, to activate peripheral I/O devices.

Processor Status

Your external hardware can monitor certain status conditions that previously required polling an internal CPU status register, as well as some that could not be monitored at all in an XT-compatible system. These status conditions include the following:

- CPU Clocks Enabled signal—to determine whether the system is fully awake and active (not in suspend mode)
- Interrupts Enabled signal—to determine whether the interrupt flag in the 8086 flags register is set or cleared
- Instruction Complete signal—to determine whether an instruction has executed, providing a true indication of system execution speed (MIPS)
- Instruction Fetch signal—to determine whether the current memory cycle is an instruction fetch.

The PS1-4 pins provide these signals when programmed for the appropriate operation.

Graphics Control

The F8680 chip provides an output function on programmable pin PS2 and an input function on programmable pin PS4 for use with the LCD panel. PS2 can output the AC Drive Clock signal needed for many LCD panels. PS4 can input any dot clock frequency for use with nonstandard LCD panels.

PCMCIA Interface

The F8680 chip provides a complete set of PCMCIA interface signals. When the interface is not used, some of these signals can be used as general purpose status monitoring inputs. The interface can also be used as a general purpose 16-bit I/O port.

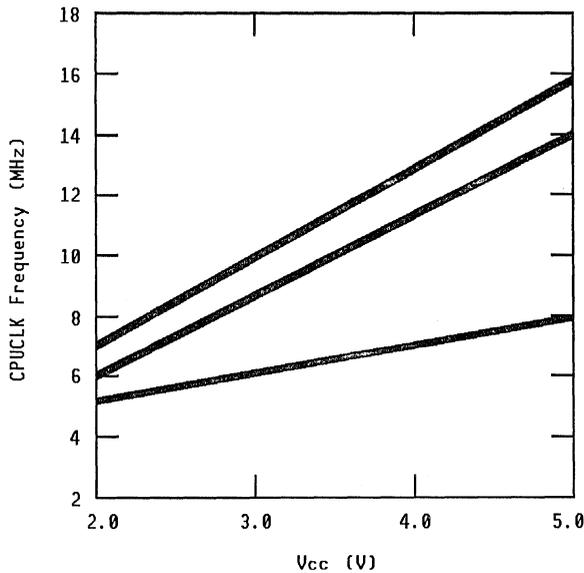
General Design Considerations

You should consider the following power and signal characteristics when designing your system. You might also want to read the design applications discussed in Chapter 2. These examples may give you some ideas on how to best take advantage of the F8680 features. Chapter 2 also directs you to the appropriate chapters for implementing each aspect of the design.

Power Considerations

The F8680 chip operates at a nominal voltage of +5V. Its full operating range is +3V - 10% to +5V + 10%. Note that the F8680 chip operational speed is reduced at lower voltages. Figure 1-3 shows the relationship of operating speed to supply voltage for some typical operating frequencies.

Figure 1-3. *Supply Voltage vs. Operational Speed*



The F8680 chip provides eight connections to Vcc. All of these pins must be connected to a supply voltage. The core processor can be operated at a lower voltage than the rest of the F8680 chip. You can connect pins 19 and 115 to supply the core at a nominal voltage of +3V and the remaining Vcc pins (VCCPAD) to supply the rest of the F8680 chip at a nominal voltage of +5V.

The F8680 chip provides 12 connections to ground. The ground pins must all be connected together. Be sure to make appropriate use of bypass capacitors near the F8680 chip.

Signal Characteristics

All F8680 chip outputs source/sink current as shown in Table 1-2. All outputs are tri-stated when the FLOAT pin is high. All F8680 chip inputs are Schmitt-triggered CMOS.

TABLE 1-2

Table 1-2. *Output Pin Drive Capability*

	Drive Current at Vcc = 3V	Drive Current at Vcc = 5V
All outputs except RD7:0	4mA	8mA
RD7:0	8mA	16mA

EMI Considerations

In your design you will want to keep electromagnetic interference (EMI) to a minimum. To this end:

- Strive to keep the OE0-1*, WE0-1*, CS10-11*, and CS20-22 lines as short as possible.
- Keep address and data bus lines short.
- If you must have long signal line runs, separate adjacent signal lines with a ground line that is grounded at both ends.

Note that all F8680 chip outputs are slew-rate-controlled to further reduce EMI tendencies.

Design Applications

The following pages provide examples of how you might imaginatively use the F8680 features in various design applications. This chapter presents the following applications:

- Palmtop
- Laptop
- Portable
- Desktop
- Pen-based
- Embedded controller.

The remaining chapters of this guide are devoted to the specifics of implementing such designs.

Palmtop Application

For a palmtop-sized system you need the best possible performance in the smallest of spaces. What you do *not* need is a system that requires several chips and provides features you cannot use in a small system. The F8680 chip excels in such an application.

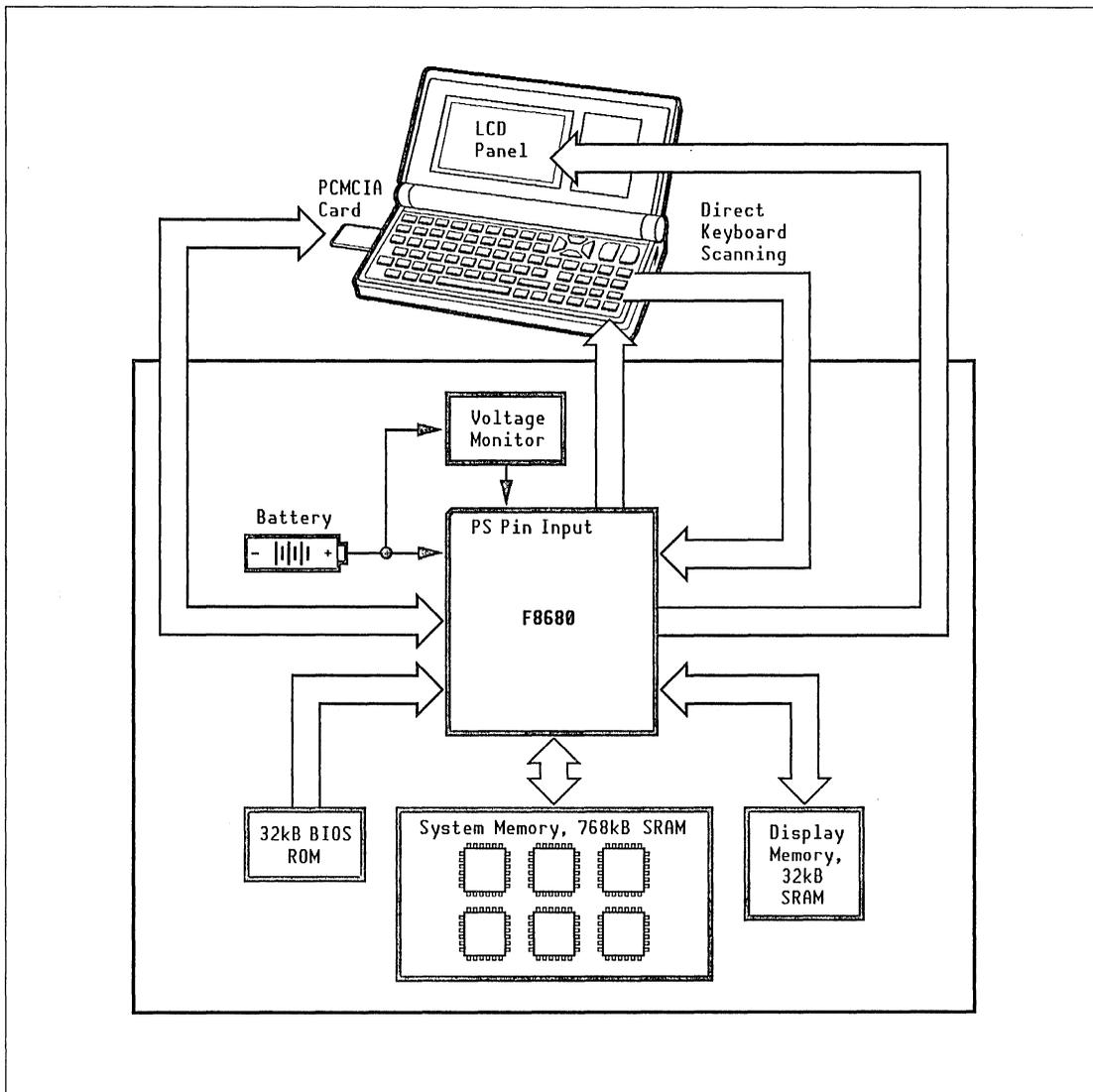
The palmtop application shown in Figure 2-1 provides a highly versatile and full-featured computer with a minimal component count. The design takes advantage of the ability of the F8680 chip to manage memory in small blocks so that the system fully utilizes all available RAM.

The 768kB memory is made up of three banks of SRAM, each organized as 128k x 8 by two bytes wide. The small BIOS requires only a 32kB ROM, which can be shadowed in RAM. The memory layout provides 608kB conventional memory, 128kB EMS memory, and 32kB for shadowing ROM. The single PCMCIA card slot allows addressing of up to 32MB on a memory card. The Execute-In-Place (XIP) feature of the memory card can be used to execute programs directly from the card. Memory subsystem design is described in Chapter 3. The memory programming for this design is found in Chapter 5 of the *F8680 PC/CHIP Programmer's Reference Manual*.

The keyboard matrix is scanned directly by the F8680 chip—no keyboard processor is needed. This scanning application is described in Chapter 6.

The battery voltage monitor circuit provides two signal inputs on the programmable pins. One signal indicates that battery voltage is low but still adequate for operation. This signal causes a switch to SuperState R mode so that the power management code can take appropriate action (sound a beep once a minute, for example). The second low voltage signal also causes a switch to SuperState R mode. The power management code recognizes this power level as too low for safe operation (the batteries might be damaged) and shuts the system down. The battery voltage monitor circuit is described in Chapter 5.

Figure 2-1. Palmtop Application



Laptop Application

Your laptop design has to perform like a desktop system, but performance means nothing if it cannot be balanced against power consumption. The variable performance features of the F8680 chip allow you to design a system that balances these two opposing forces in real time by means of your SuperState R power management code.

The laptop application shown in Figure 2-2 illustrates a low-cost laptop that features 1MB of system DRAM and a hard disk drive. The F8680 programmable pins feature is used to minimize the external logic required.

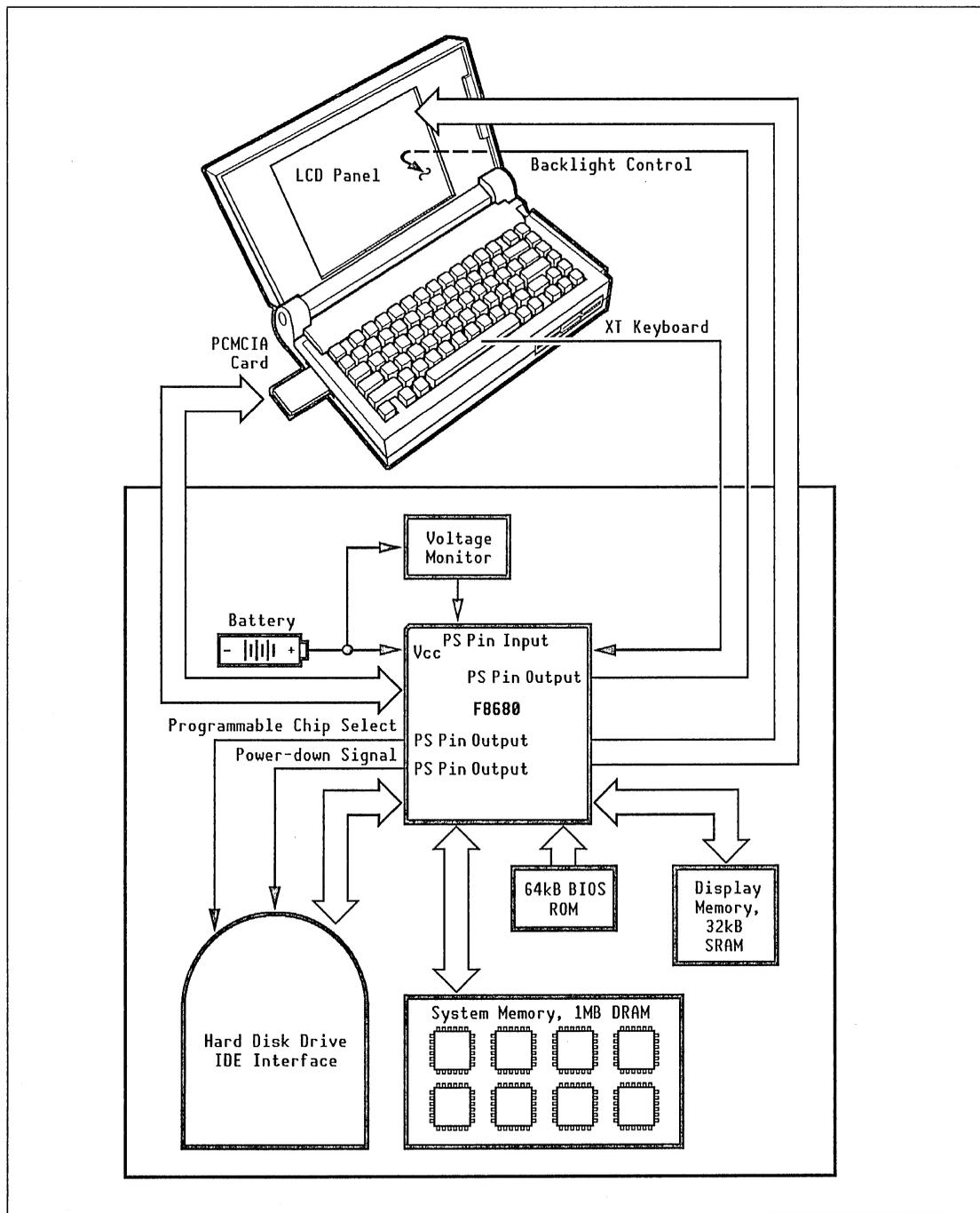
A full 640kB of RAM is available for conventional memory. The 64kB ROM is shadowed in RAM. This leaves 320kB for EMS memory. Memory subsystem design is described in Chapter 3.

In addition to the battery voltage monitor circuit described in “Palmtop Application,” the power management code also controls the LCD panel backlight and the hard disk spin-down timeouts. These controls are discussed in Chapter 5.

The hard disk drive controller is part of the hard disk electronics (IDE hard disk). A programmable pin decodes I/O accesses in the XT hard disk I/O range. Incorporation of an IDE hard disk is discussed in Chapter 8.



Figure 2-2. Laptop Application



Portable Application

A portable computer may no longer be just “XT-compatible.” It must offer higher performance, better graphics, more functionality, and greater ease-of-use than the large, clumsy systems previously available. The F8680 chip lets you design this full-featured portable system at very low cost.

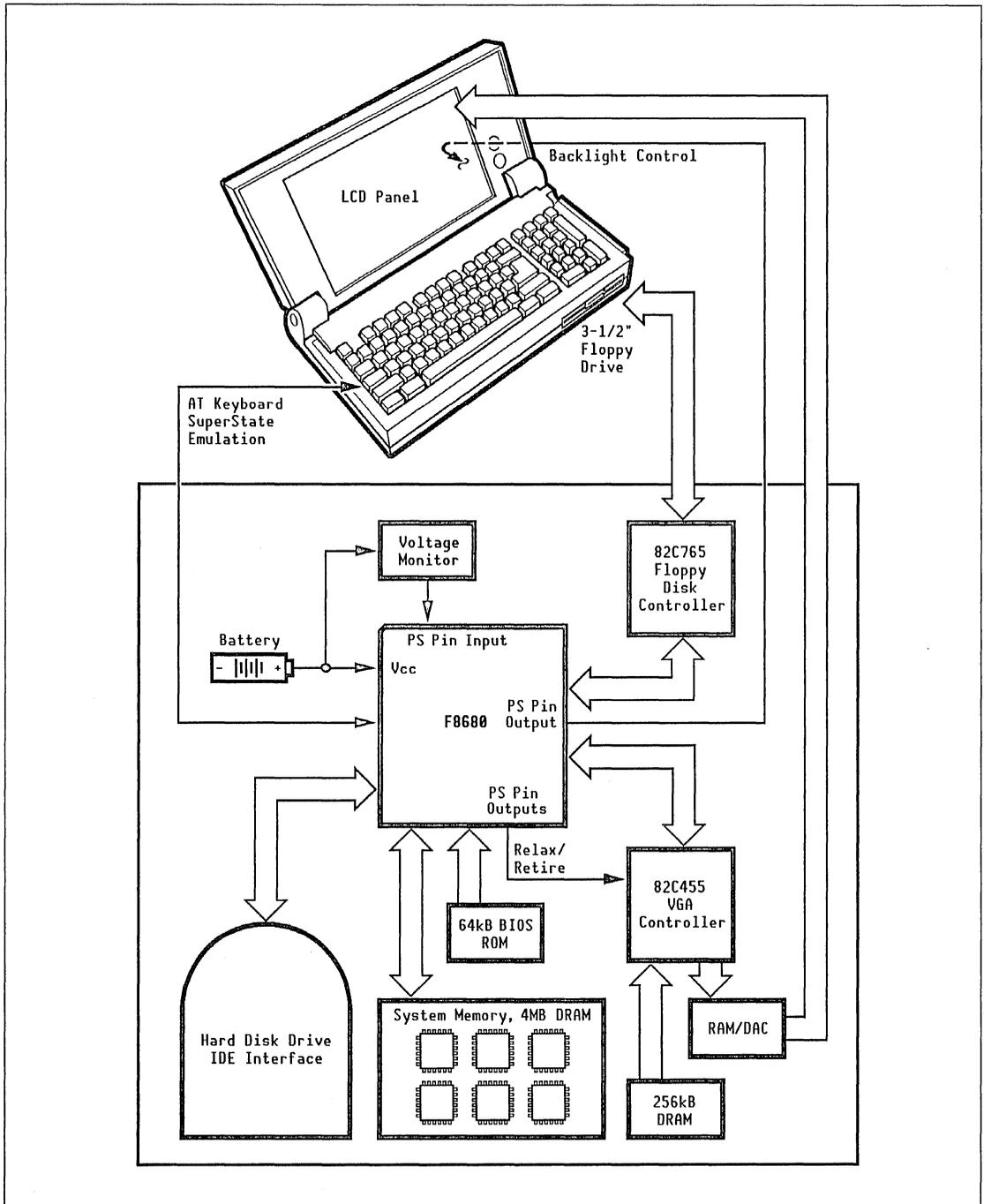
The portable application shown in Figure 2-3 provides desktop features in a portable computer, including low-power VGA, an AT-style keyboard, floppy and hard disk drives, and a large amount of system memory. Thanks to the F8680 logic, very few components are needed.

The floppy disk drive uses the CHIPS® 82C765 Floppy Disk Controller, which requires only frequency generation circuitry to implement a floppy disk subsystem. The F8680 programmable chip-select feature eliminates the need for external decoding. The floppy drive interface is described in Chapter 7.

The CHIPS 82C455 Flat Panel/CRT VGA Controller is selected to provide full 640 x 480 VGA display resolution on the LCD panel. The relax/retire modes of the 82C455 VGA controller can be controlled by the F8680 programmable pins to reduce power consumption, allowing the system to run on battery power if needed. Interfacing the 82C455 VGA controller is described in Chapter 4.

All the scan codes of the AT-style keyboard can be sent to the system through SuperState R emulation. This procedure is described in Chapter 6.

Figure 2-3. Portable Application



Desktop Application

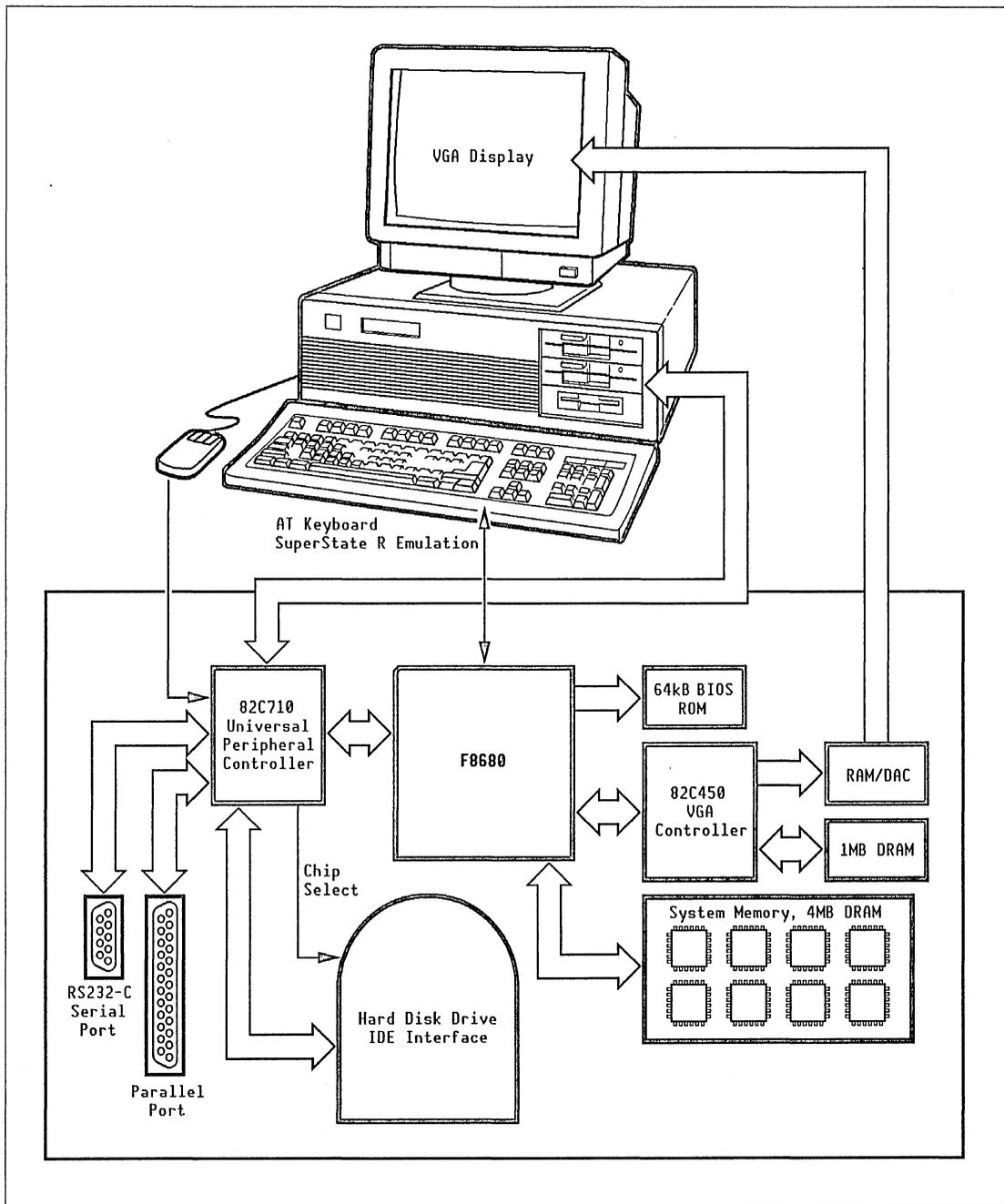
The standard desktop system of today is not considered useful unless it can handle large amounts of memory and provide a high-resolution display. These features are often associated only with 386-based systems, but the vast majority of programs never actually use any of the 386 instructions or features. The F8680 chip provides performance approaching that of many 386 systems as well as EMS memory addressing, but at a fraction of the cost.

The desktop application shown in Figure 2-4 provides a powerful super-XT desktop computer with up to 12MB on-board DRAM, super VGA display capability, two serial ports, a parallel port, a floppy drive, and a hard disk drive.

The CHIPS 82C710 Universal Peripheral Controller provides the floppy disk controller, hard disk drive interface, and parallel port, and adds a second serial port, all in a single component. Interfacing the 82C710 controller is described in Chapter 9.

The 82C450 One Megabit DRAM VGA Graphics Controller provides super VGA display resolution (1024 x 768) when used with 1MB of display RAM. The controller uses either two or four 256k x 4 DRAM chips so that a basic 512kB system can be upgraded to 1MB high-resolution operation at any time. A total of only seven chips provide the entire VGA display subsystem. Interfacing the 82C450 VGA controller is described in Chapter 4.

Figure 2-4. Desktop Application



Pen-Based Application

The pen-based computer design depends on ease of portability and low power consumption. It also demands a protected management mode so that reading the pen data will not interfere with normal DOS operations. The F8680 chip is perfectly suited to the task of managing pen data while simultaneously running MS-DOS applications.

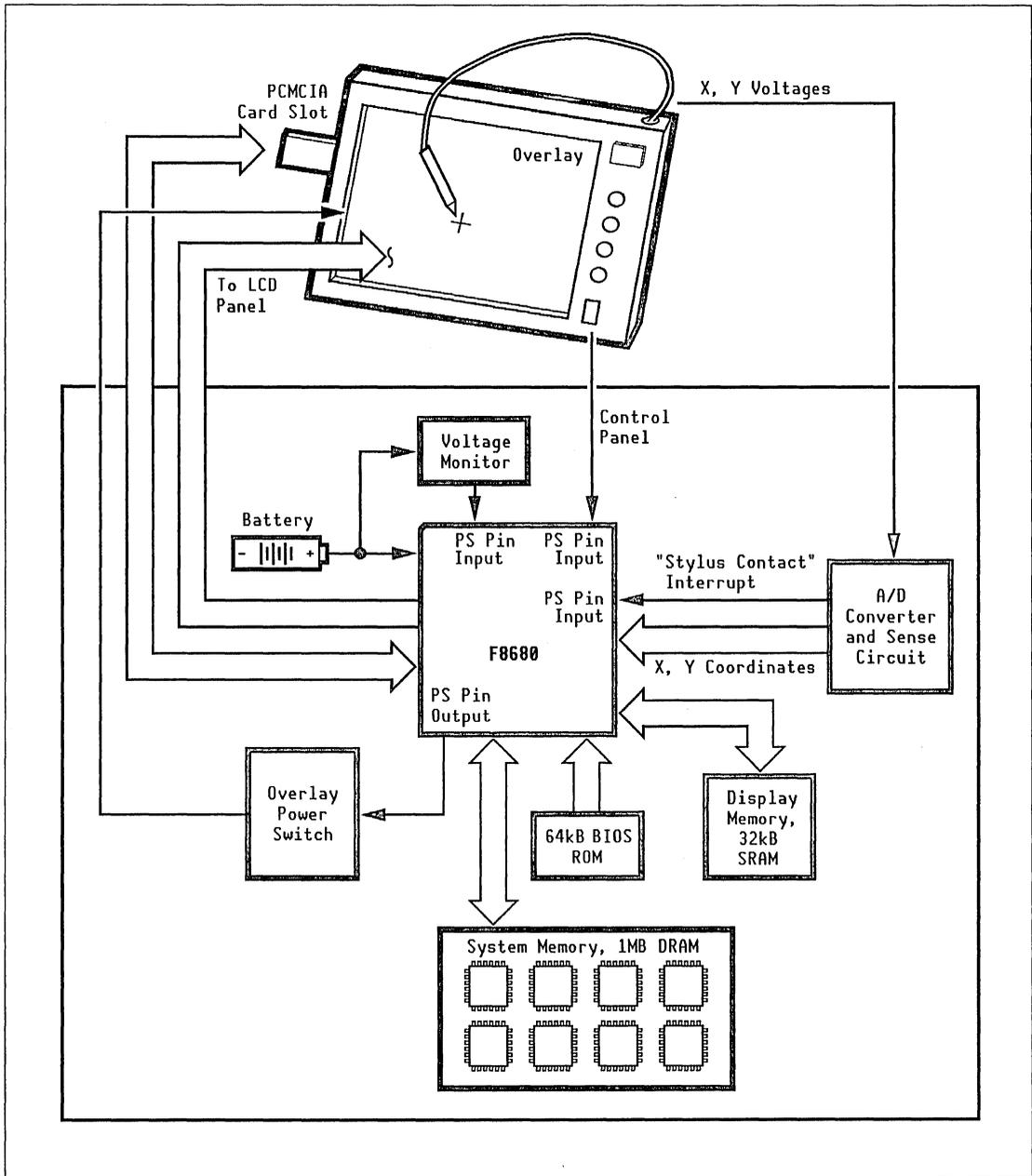
The pen-based application shown in Figure 2-5 uses the external SuperState R switch feature of the F8680 chip to respond to the “stylus contact” interrupt generated when the pen touches the conductive overlay surface of a pen-based computer.

Only an analog-to-digital (A/D) converter and a “stylus contact” sensing circuit are required to interface the F8680 component to the overlay. The programming is straightforward: When the pen contacts the overlay, the F8680 chip switches to SuperState R mode to read the voltage sensed through the A/D converter. The F8680 chip can then return digitized X-Y coordinates to the application program.

The SuperState R external switch request feature simplifies the interrupt process by eliminating the need for the “stylus contact” interrupt to use any of the system IRQ lines. SuperState R code reads the X-Y position information without interruptions from applications. The F8680 programmable pins also make monitoring control buttons a simple matter. Each button can generate its own switch request to SuperState R mode if desired.

Chapter 11 describes the interfacing of an overlay.

Figure 2-5. Pen-based Application



Embedded Controller Application

The F8680 chip is ideal for use in an embedded controller application if the application meets any of the following criteria:

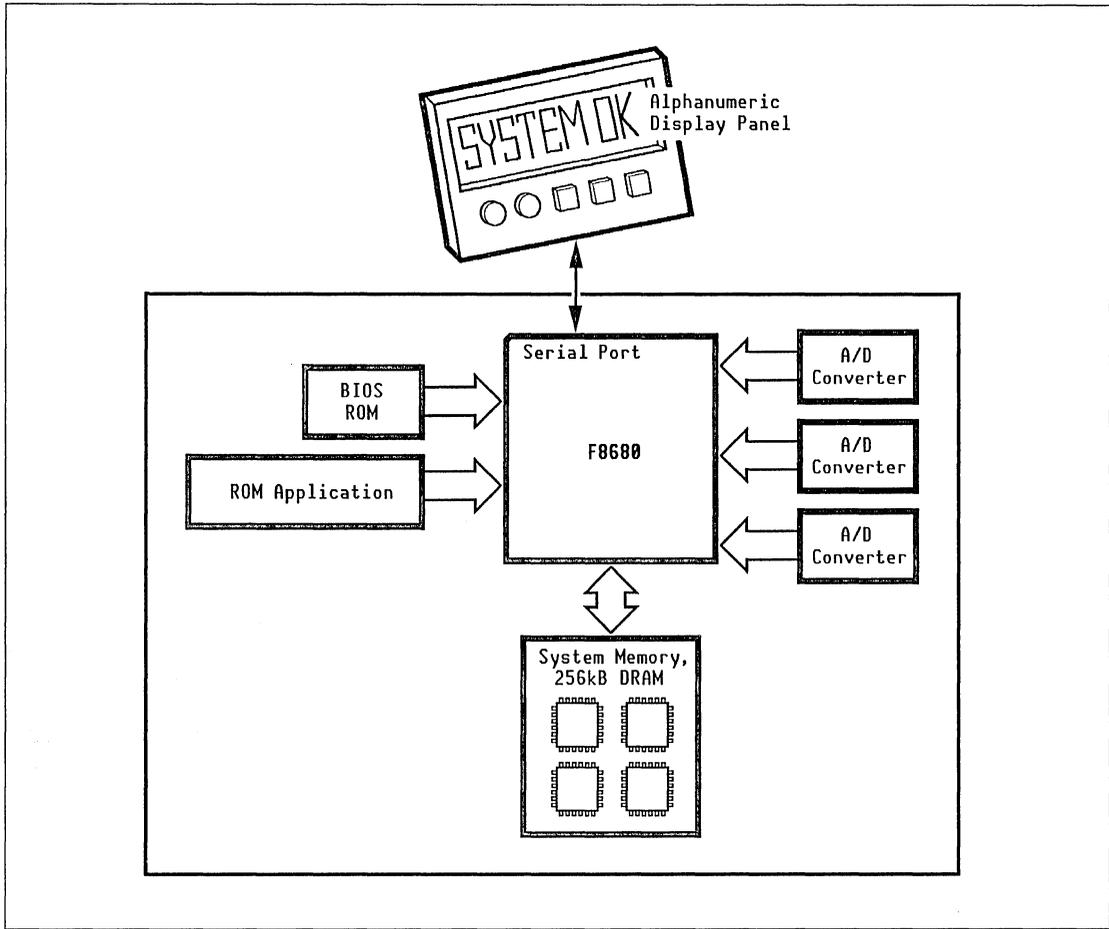
- You prefer DOS compatibility so that you can use off-the-shelf software development tools.
- You want to avoid dependence on programmers versed in the arcane assembly language specific to certain embedded controllers.
- You need a PC-compatible bus interface, but you do not want to waste money on an expandable system that will never be expanded beyond your initial design.
- You want to improve performance and add new capabilities to your current 8086-based embedded controller design, but you do not want to risk modifying reliable code in order to support a new hardware platform.

The F8680 chip allows you to attain your design goals by providing a well-understood, industry-standard processing platform that has the flexibility to adapt to your unique needs. Moreover, you can use SuperState R code to add new performance features to your system without altering 8086 code and possibly introducing problems.

The embedded controller application shown in Figure 2-6 illustrates a small Supervisory Control And Data Acquisition (SCADA) system. The system monitors analog voltages and displays them on a distant alphanumeric display console, through the serial port. A D/A converter could also be added to remotely control an output voltage.

The connection of A/D and D/A converters is described in Chapter 11. The serial port connections are described in Chapter 9.

Figure 2-6. Embedded Controller Application



Memory

The memory controller of the F8680 chip supports most of the commonly used memory types. The controller provides chip-select decoding for up to three banks of memory. You have to provide external address decoding for any additional banks you want to connect.

Each bank can be any one of the following types:

- 256k x 1, 256k x 4, 512k x 8, 1M x 1, 1M x 4, and 4M x 1 DRAM
- Any type of SRAM or PSRAM (such as 32k x 8, 128k x 8, and 512k x 8)
- PCMCIA memory card
- ROM
- Memory on the XT bus.

You can mix these types any way you choose. The best choices are DRAM, SRAM/PSRAM, and PCMCIA memory, all of which provide good performance when configured for word (16-bit) access.

Connecting the memory is generally very routine and predictable. The only decisions to make on the hardware side are the memory type and capacity. You make all your mapping and configuration choices through software. Use the memory configuration worksheet provided in the “Memory” chapter of the *F8680 PC/CHIP Programmer’s Reference Manual* when you are ready to decide how to map your memory into the system address space.

Memory Operational Theory

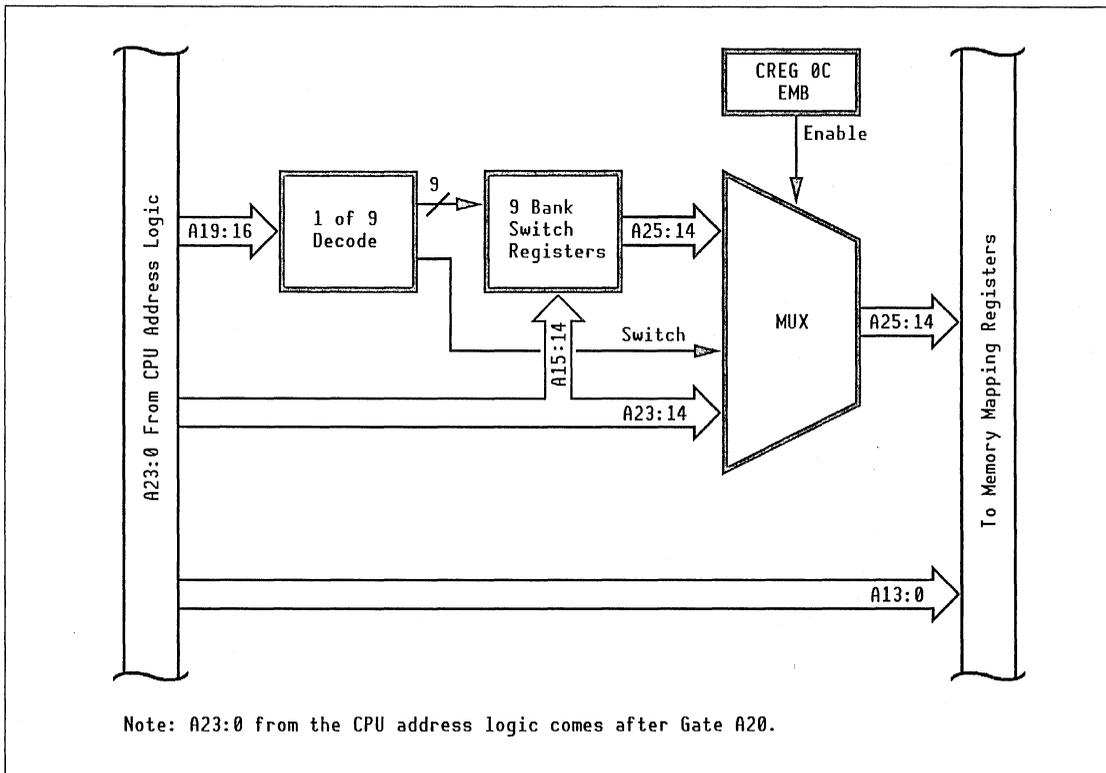
This section provides only a brief summary of the concepts involved in system memory mapping. For a more detailed explanation, refer to the *F8680 PC/CHIP Programmer’s Reference Manual*.

The memory manager provides independent *bank switching* and *memory mapping* mechanisms to deal with the translation from logical CPU addresses to physical addresses in RAM. The bank switching mechanism takes linear CPU addresses that fall in the B0000 to FFFFF range and moves them so that the effective logical address can fall anywhere in the 64MB address space of the F8680 chip. The memory mapping mechanism takes the logical address, after any possible shifting by the bank switching mechanism, and maps it on a block-by-block basis to available space in physical RAM. Either 32kB or 64kB blocks can be selected.

Bank Switching Logic

Figure 3-1 illustrates the bank switching logic used in the system. The logic first decodes address bits A19:16 from the CPU. If the address is in the B0000 to FFFFF range, the decoder signals the multiplexer to use the contents of one of the nine Bank Switch Registers instead of the upper bits of the address from the CPU.

Figure 3-1. Bank Switching Logic



The decoder selects the Bank Switch Register whose contents will substitute the upper address bits from the CPU according to the nine subranges listed in Table 3-1. Address bits A15:14 may or may not be substituted by the Bank Switch Register address bits, depending on the size of the subrange decoded.

Table 3-1. *Bank Switch Register Ranges*

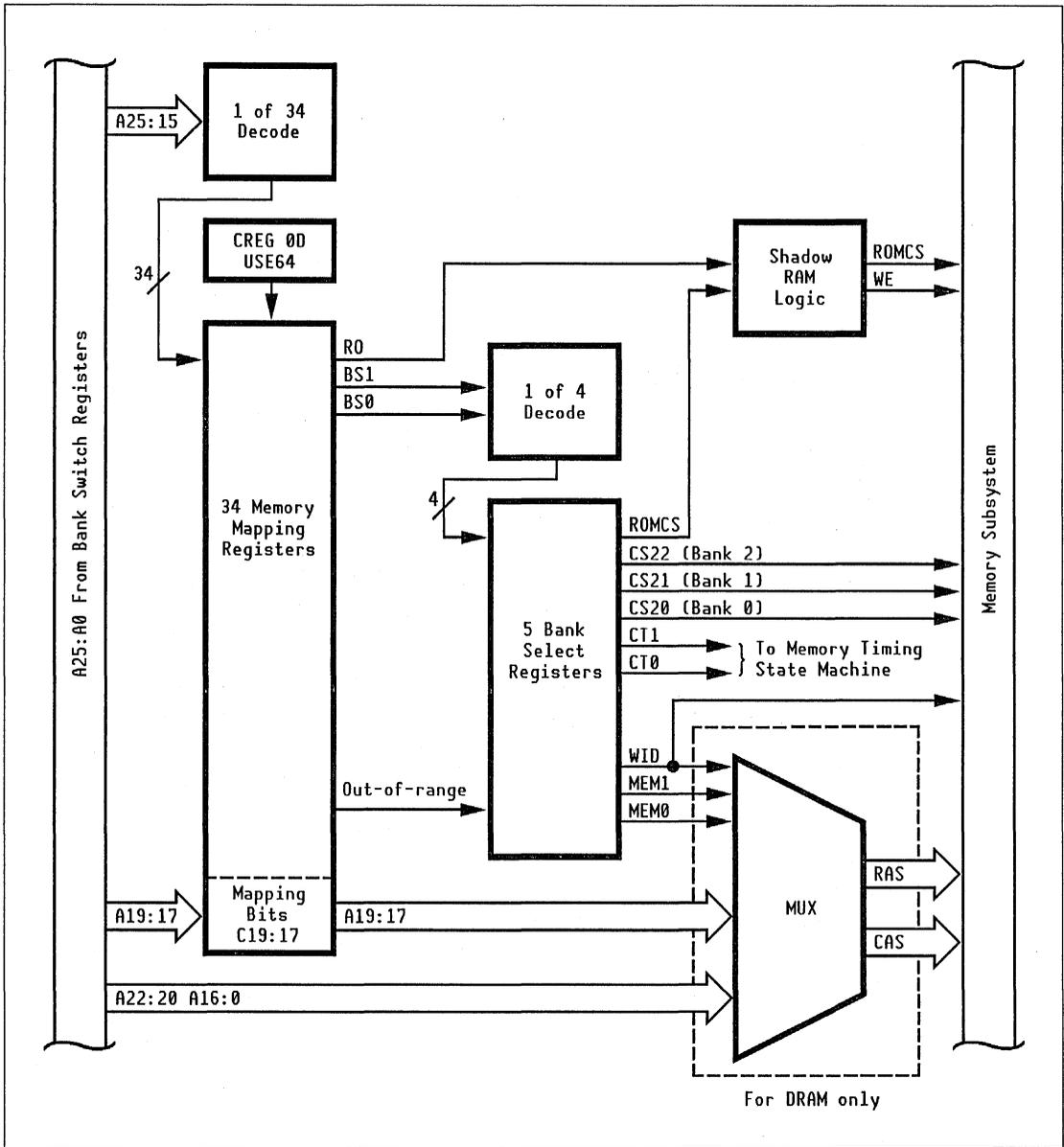
Bank Switch Register	Corresponding Subrange
0	B0000-B7FFF
1	B8000-BFFFF
2	C0000-CFFFF
3	D0000-D3FFF
4	D4000-D7FFF
5	D8000-DBFFF
6	DC000-DFFFF
7	E0000-EFFFF
8	F0000-FFFFFF

The multiplexer passes on the substituted address bits only when enabled by the EMB bit in CREG 0C. The resulting 26-bit address A25:0 is passed on to the memory mapping logic. Note that bank switching is disabled on power-up.

Memory Mapping Logic

Figure 3-2 illustrates the logic used to map the logical addresses from the bank switch mechanism to locations in physical RAM. The upper address bits A25:15 are decoded to select one of 34 address ranges. The USE64 bit in CREG 0D selects the decoding method. If USE64 = 0, each of the first thirty-two 32kB blocks of logical addresses corresponds to a separate mapping register, with the next two 512kB blocks each corresponding to a mapping register. If USE64 = 1, each of the first thirty-two 64kB blocks of logical addresses corresponds to a separate mapping register, with the next two 1MB blocks each corresponding to a mapping register.

Figure 3-2. Memory Mapping Logic



Unlike the Bank Switch Registers, the Memory Mapping Registers do not contain replacements for the logical address bits from the CPU. The mapping registers simply select the physical bank of RAM to which the logical address should be directed. However, the registers do provide mapping bits. These bits serve to shift the address up or down in multiples of 128kB within the physical RAM bank, allowing “lost” RAM (such as RAM whose address would overlap the display SRAM address range) to be utilized elsewhere.

Each Memory Mapping Register selects one of four Bank Select Registers, which in turn activates the control signals for the selected bank of RAM. If the CPU generates an address outside the range of the 34 Memory Mapping Registers, the logic automatically selects a fifth Bank Select Register reserved for this purpose. No mapping bits are provided for out-of-range accesses.

How to Approach Memory Design

The background provided in the previous section and the following information should enable you to begin a memory subsystem design.

The memory controller handles three banks of RAM, providing the bank select signals CS20, CS21, and CS22 to activate banks 0, 1, and 2, respectively (see Figure 3-2). You can also use additional banks, but the F8680 chip provides no additional bank select signals. You would have to decode the memory addresses with external logic.

You program the memory controller through CREG locations for each bank according to the following parameters: memory cycle type (DRAM, SRAM, XT bus, or PCMCIA); bank width (one or two bytes); address multiplexing (for DRAM only); and required ROMCS line status (active/inactive) for that bank.

You can use DRAM, SRAM, or PSRAM in your design, and you can mix all three. You cannot mix types within the same bank, but you can mix different size devices of the same type. Chapter 10 of the *F8680 PC/CHIP Programmer's Reference Manual* describes an interesting application (Example 2) where RAMs of different sizes and widths are used to optimize performance in a certain addressing range.

Refer to the following sections for basic examples of the connections required to interface each type of memory to a bank. Then refer to the *F8680 PC/CHIP Programmer's Reference Manual* to determine the most effective way to utilize the RAM in your hardware design.

Note: When the graphics subsystem is enabled, all accesses at segment 0B800 are considered graphics cycles and are automatically forwarded to the graphics controller.

DRAM

You use DRAM in your hardware design for efficient, high-performance, low-cost memory space. There are drawbacks, however. For example, DRAM consumes a lot of power when compared to SRAM. Special features of the F8680 chip, such as page mode operation, slow refresh, and performance control, help to lessen the impact of this power consumption and allow you more flexibility in your design. However, you may still want to consider a mix of DRAM, SRAM, and PCMCIA memory in your system, as mixing memory types is effortless with the F8680 chip.

Signals

Table 3-2 lists the pin assignments and signals provided for the DRAM interface.


Table 3-2. Pin Assignments—DRAM Interface Signals

Address Bus		Data Bus		Control Bus	
Pin	Signal	Pin	Signal	Pin	Signal
52	ADR0	60	RD0	69	CS10*
50	ADR1	61	RD1	86	CS11*
47	ADR2	62	RD2	73	CS20*
45	ADR3	63	RD3	74	CS21*
43	ADR4	64	RD4	75	CS22*
40	ADR5	66	RD5	70	OE0*
38	ADR6	67	RD6	87	OE1*
36	ADR7	68	RD7	71	WE0*
34	ADR8	77	RD8	88	WE1*
32	ADR9	78	RD9		
29	ADR10	79	RD10		
27	ADR11	80	RD11		
		81	RD12		
		83	RD13		
		84	RD14		
		85	RD15		

where: ADR11:0

Address Bus (O)—provides memory addresses for linear addressing on any byte boundary. When DRAM is set up for word access, only ADR11:1 are used; the CS0-1, OE0-1* and WE0-1* lines select between the high byte (1) and the low byte (0).

CPUCLK

CPU Clock (I)—processor clock input that also sets memory timings.

CS10-11*

CAS Lines (O)—CS10* indicates CAS selection for the low byte; CS11* indicates CAS selection for the high byte.

CS20-22*

RAS Lines (O)—CS20* indicates RAS selection for bank 0; CS21* indicates RAS selection for bank 1; CS22* indicates RAS selection for bank 2. The active low sense can be changed to active high through CREG 0D.

OE0-1*

Output Enables (O)—indicate to the low (OE0*) and high (OE1*) bytes of the currently selected RAM bank whether they should enable data from memory onto the RD bus. The OE0-1* signals are used only with 4-bit-wide DRAM.

- RD15:0 Data (I/O)—connected to system memory as well as to the XT data bus. RD15:8 form the high byte, and RD7:0 form the low byte.
- WE0-1* Write Enables (O)—indicate to the low (WE0) and high (WE1) bytes of the currently selected RAM bank whether they should write the data on the RD bus to memory.

Memory Address Multiplexing

The MEM bits and the WID bit of the Bank Select Registers select the type of address multiplexing that will be performed during DRAM cycles. Refer to the *F8680 PC/CHIP Programmer's Reference Manual* for Bank Select Register programming details.

Connect one-byte-wide RAM to the address bus starting with bit 0, and two-bytes-wide RAM starting with bit 1. The address bits are multiplexed according to Table 3-3. Address bits above bit 11 are not changed during a DRAM cycle. You can decode these bits with external chip select decode logic if you need to support more than three banks of memory and have no more CS2x lines available.

Table 3-3. Address Multiplexing

MEM Bits	WID Bit	ADDR	11	10	9	8	7	6	5	4	3	2	1	0
		CAS	11	10	9	8	7	6	5	4	3	2	1	0
00 256k x 1	0	RAS	20	19	18	17	16	15	14	13	12	11	10	9
01 1M x 1	0	RAS	x	x	18	17	16	15	14	13	12	11	10	19
10 4M x 1	0	RAS	x	19	18	17	16	15	14	13	12	11	20	21
00 256k x 2	1	RAS	x	x	18	17	16	15	14	13	12	11	10	x
01 1M x 2	1	RAS	x	19	18	17	16	15	14	13	12	11	20	x
10 4M x 2	1	RAS	20	19	18	17	16	15	14	13	12	21	22	x

Page Mode Operation

The memory controller can provide page mode operation, in which the controller leaves the CS2x (RAS) line low between memory accesses. Then, if the next memory request does not change the row addressing, only the CS1x (CAS) part of the cycle will be performed. Refer to “DRAM Interface” in Chapter 5 of the *F8680 PC/CHIP Programmer’s Reference Manual* for details on page mode operation.

Refresh

The F8680 chip generates the periodic refresh necessary to keep DRAM contents alive. The refresh timing is set by programming timer channel 1 just as it would be for the XT. A slow refresh feature is also available through programmable configuration by setting CREG 16h. Refer to “Refresh” in Chapter 5 of the *F8680 PC/CHIP Programmer’s Reference Manual* for details on refresh.

The memory controller provides a CAS before RAS type of refresh: the controller activates the CS1x line (CAS) first, and follows this by activating the CS2x (RAS) line. This method reduces the power required to perform refresh when compared to a RAS-only refresh.

When DRAM is present on more than one bank, the banks are refreshed in sequence, not simultaneously, to avoid the current surge that could occur if all banks were refreshed at once.

Interface to Common Types

Figures 3-3 through 3-5 illustrate the connections necessary to interface each type of DRAM to the F8680 chip.

Figure 3-3 shows the connections to a bank of DRAM made up of 4-bit devices (256k x 4 or 1M x 4 DRAM).

Figure 3-3. Interface to 4-bit DRAM

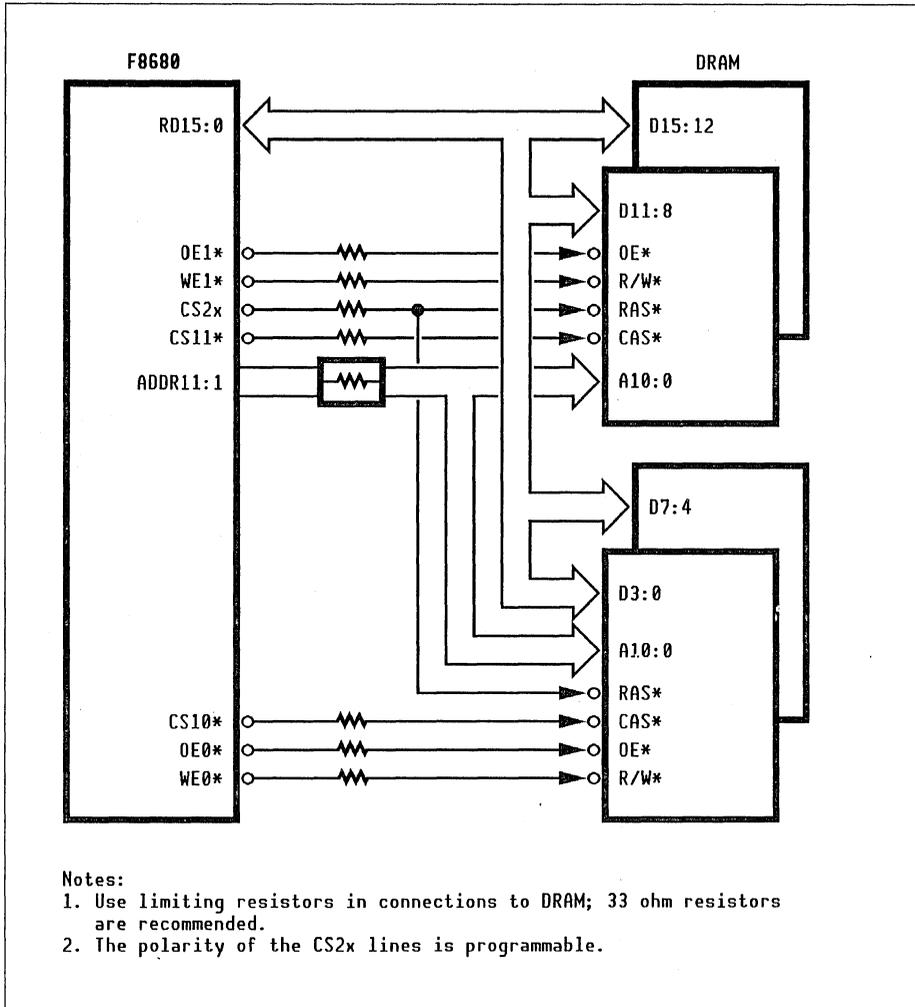


Figure 3-4 shows the connections to a bank of DRAM made up of 8-bit devices (512k x 8 DRAM).

Figure 3-4. Interface to 8-bit DRAM

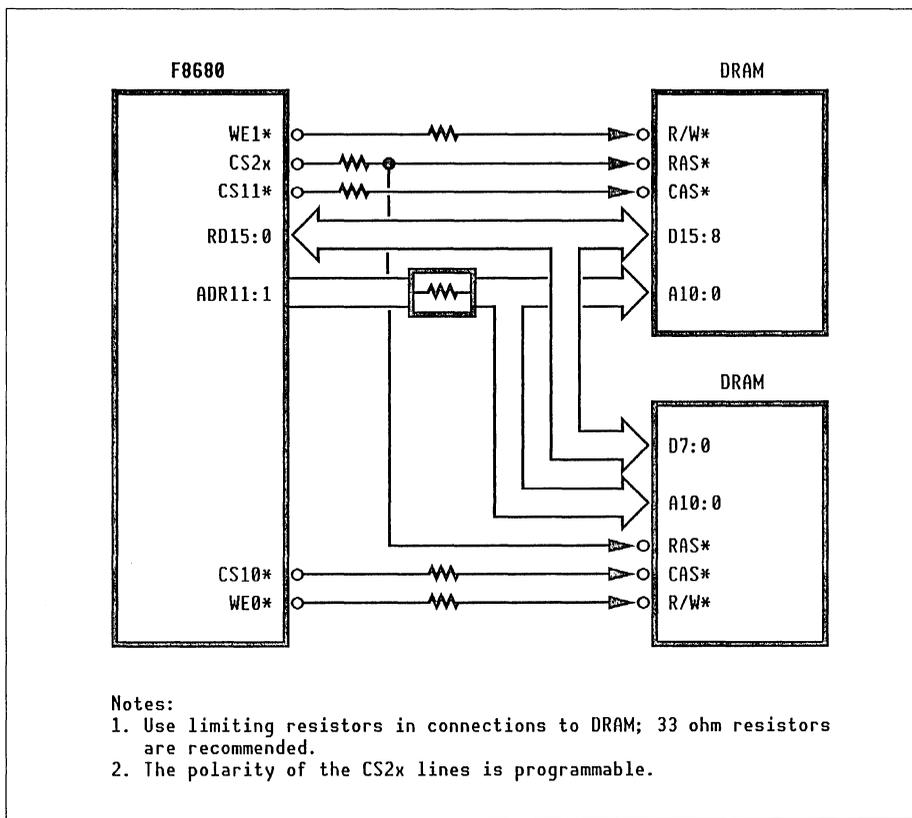
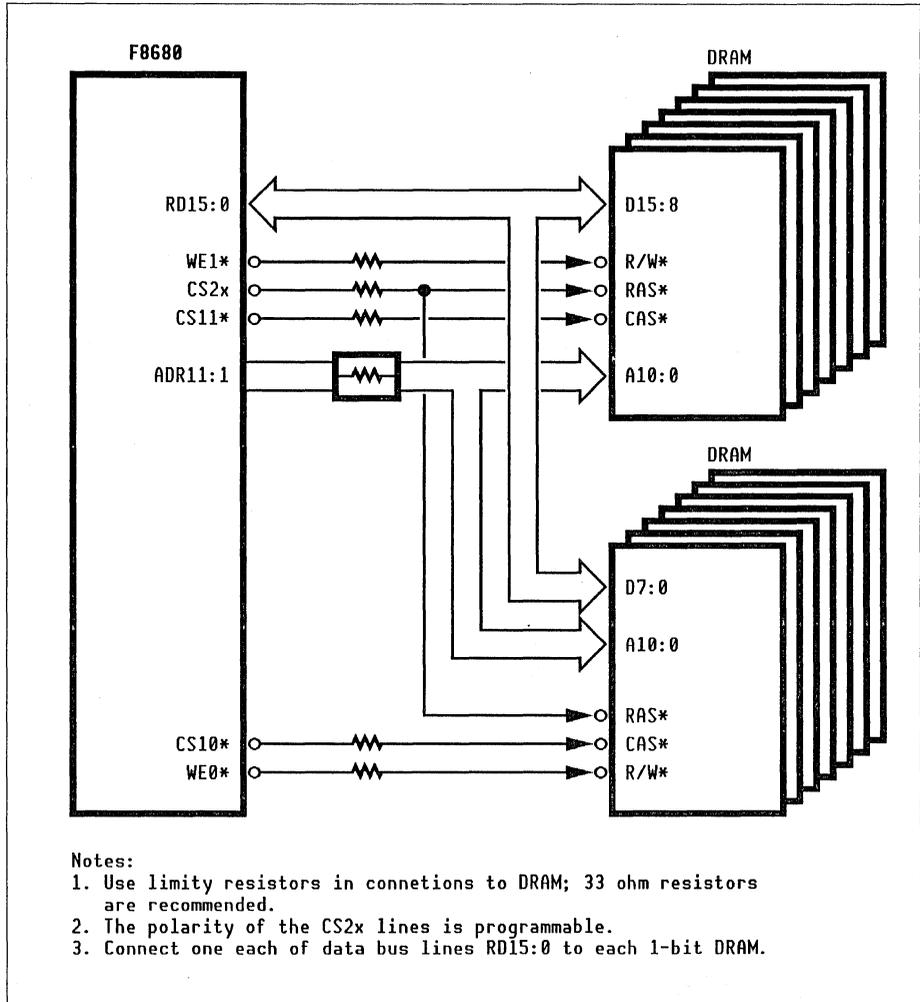


Figure 3-5 shows the connections to a bank of DRAM made up of 1-bit devices (256k x 1, 1M x 1, or 4M x 1 DRAM).

Figure 3-5. Interface to 1-bit DRAM



SRAM/PSRAM

SRAM is an ideal memory type for high performance and low power draw. It has the added advantage that it retains its contents without the periodic refresh required by DRAM. However, it is expensive when compared to DRAM and is not the best choice in low-cost applications.

Signals

Table 3-4 lists the pin assignments and signals provided for the SRAM interface.

Table 3-4. Pin Assignments—SRAM Interface Signals

Address Bus		Data Bus		Control Bus	
Pin	Signal	Pin	Signal	Pin	Signal
52	ADR0	60	RD0	73	CS20*
50	ADR1	61	RD1	74	CS21*
47	ADR2	62	RD2	75	CS22*
45	ADR3	63	RD3	70	OE0*
43	ADR4	64	RD4	87	OE1*
40	ADR5	66	RD5	76	REFRESH*
38	ADR6	67	RD6	71	WE0*
36	ADR7	68	RD7	88	WE1*
34	ADR8	77	RD8		
32	ADR9	78	RD9		
29	ADR10	79	RD10		
27	ADR11	80	RD11		
25	ADR12	81	RD12		
23	ADR13	83	RD13		
21	ADR14	84	RD14		
18	ADR15	85	RD15		
16	ADR16				
14	ADR17				
11	ADR18				

where: ADR19:0

Address Bus (O)—provides memory addresses for linear addressing on any byte boundary. When SRAM is configured for word access, only ADR18:1 are used; the 0 and 1 CS, OE, and WE lines select between the high byte (1) and the low byte (0).

CS20-22*	Chip Select Lines (O)—CS20* selects bank 0; CS21* selects bank 1; CS22* selects bank 2. The active low sense can be changed to active high through CREG 0D.
OE0-1*	Output Enables (O)—indicate to the low (OE0*) and high (OE1*) bytes of the currently selected RAM bank whether they should enable data from memory onto the RD bus.
REFRESH*	Refresh (O)—indicates when PSRAM should perform a refresh operation.
RD15:0	Data (I/O)—connected to system memory as well as to the XT data bus. RD15:8 form the high byte, and RD7:0 form the low byte.
WE0-1*	Write Enables (O)—indicate to the low (WE0*) and high (WE1*) bytes of the currently selected RAM bank whether they should write the data on the RD bus to memory.

Refresh for PSRAM

PSRAM provides its own internal refresh logic, allowing the PSRAM to maintain its contents when the system stops sending REFRESH* pulses. The memory controller provides a refresh signal REFRESH* that is compatible with the internal refresh logic of the PSRAM.

Just before the F8680 chip goes into suspend mode, the chip emits a final REFRESH* pulse of minimum 8 μ s duration. This pulse enables the internal refresh logic of the PSRAM. The PSRAM contents cannot be accessed, but they are maintained. As soon as the F8680 chip leaves suspend mode and begins to send regular REFRESH* pulses, the PSRAM goes active and disables its internal refresh logic.

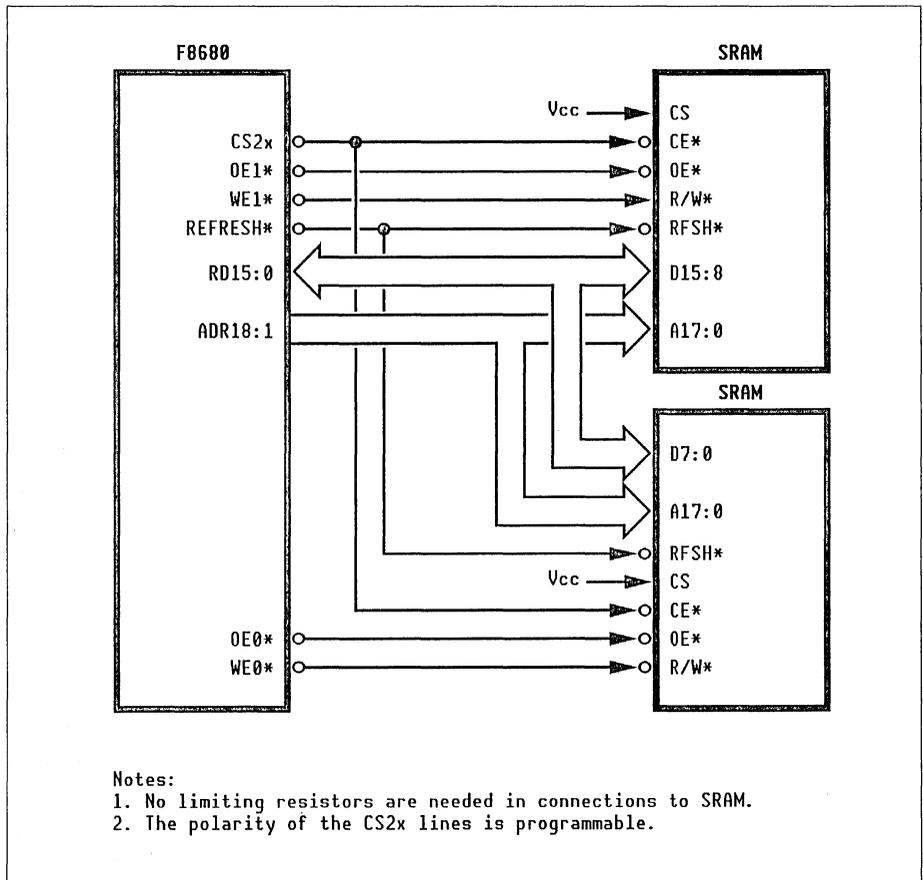
Interface to Common Types

Figure 3-6 illustrates the connections necessary to interface a word-wide bank of SRAM to the F8680 chip. The interface accommodates 32k x 8, 128k x 8, and 512k x 8 devices with the same connections. Only the number of address lines connected varies, as shown in Table 3-5. To interface a byte-wide bank of SRAM, connect the system address lines ADR18:0 directly to the SRAM address lines A18:0.

Table 3-5. Address Bus Connections for Various SRAM Device Capacities

System Address	SRAM Address		
	32k x 8	128k x 8	512k x 8
ADR15:1	A14:0	A14:0	A14:0
ADR17:16	n/c	A16:15	A16:15
ADR19:18	n/c	n/c	A18:17

Figure 3-6. Interface to SRAM



ROM

The memory controller supports ROM as a device on the XT bus. For this reason, only 8-bit accesses can be made to ROM. If the ROM device meets the timing requirements of SRAM or PCMCIA memory, you can use ROM banks that are one word wide. You would configure the banks as if they were SRAM or PCMCIA cards.

BIOS ROM

The most common use of ROM is to provide the system BIOS. After reset the F8680 chip sets its ROMCS* line active and begins executing from bank 0 (BS0), which points to the XT bus by default. Therefore, the BIOS ROM bank will be 8-bits wide.

You will probably want to consider copying the BIOS from ROM to system RAM at boot time. This is known as *shadowing* ROM in RAM. Running the BIOS from RAM is much faster, not only because ROM devices are usually slow but also because RAM is often configured in a 16-bit bank.

Shadowing ROM in RAM is a decision you make in software; it does not impact your hardware design. Refer to Chapter 5 of the *F8680 PC/CHIP Programmer's Reference Manual* for more information on shadowing the BIOS in RAM.

Applications in ROM

Your system design might also provide application programs in ROM. Generally these programs will be copied as needed from ROM to RAM for execution, then deleted from RAM when their execution is complete. Running applications from ROM is not recommended, as performance is poor.

Signals

Table 3-6 lists the F8680 chip signals provided by the ROM/PROM interface to 8-bit devices. If you intend to connect ROM using the SRAM or PCMCIA interface, refer to the related sections in this chapter on the appropriate interface for more signal information.

Table 3-6. Pin Assignments—ROM/PROM Interface Signals

Address Bus		Data Bus		Control Bus	
Pin	Signal	Pin	Signal	Pin	Signal
52	ADR0	60	RD0	48	ROMCS*
50	ADR1	61	RD1	8	MEMW*
47	ADR2	62	RD2	10	MEMR*
45	ADR3	63	RD3		
43	ADR4	64	RD4		
40	ADR5	66	RD5		
38	ADR6	67	RD6		
36	ADR7	68	RD7		
34	ADR8				
32	ADR9				
29	ADR10				
27	ADR11				
25	ADR12				
23	ADR13				
21	ADR14				
18	ADR15				
16	ADR16				
14	ADR17				
11	ADR18				
9	ADR19				

where: ADR19:0

Address Bus (O)—provides memory addresses for linear addressing on any byte boundary.

MEMR*

Memory Read (O)—indicates that the ROM is being accessed for a memory read.

MEMW*

Memory Write (O)—indicates that the PROM is being accessed for a memory write. Usually a special programming voltage is also required for writing to programmable ROM.

ROMCS*

ROM Chip Select (O)—activates the ROM, usually for accesses to the BIOS before the BIOS has been shadowed in RAM.

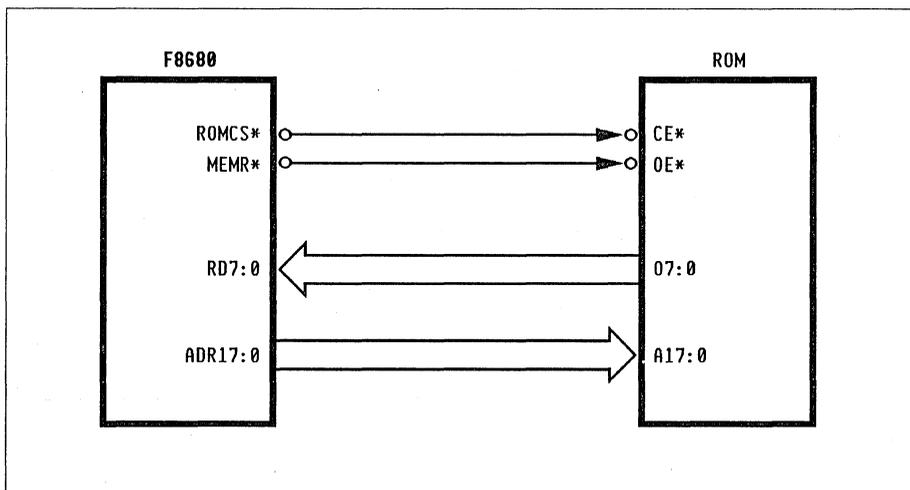
RD7:0

Data (I/O)—connected to system memory, as well as to the XT data bus. RD15:8 form the high byte, and RD7:0 form the low byte.

Interface to 8-bit ROM Banks

Figure 3-7 illustrates the connections necessary to interface a byte-wide bank of ROM.

Figure 3-7. *Interface to 8-bit ROM*



PCMCIA-Standard Interface

The PC Memory Card Interface Association (PCMCIA) standard defines a mass-storage memory card in terms of its physical, electrical, and programming interfaces. All RAM cards, ROM cards, flash-type EPROM cards, and silicon disks that adhere to this standard can use the same physical and electrical interfaces. Separate software drivers are usually required for each storage card type.

A description of the full PCMCIA standard is provided in the *PC Card Standard* document available from PCMCIA. Refer to this document for more complete and useful information about the PCMCIA standard.

Signals

The F8680 chip provides control and status lines that conform to PCMCIA guidelines. The PCMCIA signals and their equivalent signal names on the F8680 chip are listed in Table 3-7.

Table 3-7. *PCMCIA Signal Name Equivalents*

Signal Description	PCMCIA Name	F8680 Name
Address Bus	A25:0	ADR24:0 ^{1,2}
Data Bus	D15:0	RD15:0
Card Enable (even bytes)	CE1	MCCE1
Card Enable (odd bytes)	CE2	MCCE2
Attribute Memory Select	REG	ADR23 or PS pin ¹
Output Enable	OE	OEO
Write Enable/Program	WE/PGM	WE0
Ready/Busy	RDY/BSY	MCRDY
Write Protect	WP	PS pin (input)
Program Voltage	Vpp	PS pin (output) ³
Refresh	RFSH	REFRESH
Card Detect	CD1-2	MCCD1-2
Battery Voltage Detect	BVD1-2	MCBAT1-2
Card B Select	—	CARDB

¹ See “Chip Support for PCMCIA” section for suggested use of signals.

² PCMCIA address line A25 is not used.

³ The PS pin would gate a power control device to provide programming voltage.

Table 3-8 lists the pin assignments of the PCMCIA interface signals.

Table 3-8. Pin Assignments—PCMCIA Interface Signals

Address Bus		Data Bus		Control Bus	
Pin	Signal	Pin	Signal	Pin	Signal
52	ADR0	60	RD0	70	OE0
50	ADR1	61	RD1	71	WE0*
47	ADR2	62	RD2	76	REFRESH*
45	ADR3	63	RD3	90	MCCE2*
43	ADR4	64	RD4	91	MCCE1*
40	ADR5	66	RD5	92	MCRDY
38	ADR6	67	RD6	93	MCCD1*
36	ADR7	68	RD7	94	MCCD2*
34	ADR8	77	RD8	95	MCBAT1
32	ADR9	78	RD9	96	MCBAT2
29	ADR10	79	RD10	97	CARDB
27	ADR11	80	RD11		
25	ADR12	81	RD12		
23	ADR13	83	RD13		
21	ADR14	84	RD14		
18	ADR15	85	RD15		
16	ADR16				
14	ADR17				
11	ADR18				
9	ADR19				
58	ADR20				
57	ADR21				
56	ADR22				
55	ADR23				
54	ADR24				
53	ADR25				

where: ADR25:0

Address Bus (O)—provides system addresses for linear addressing on any byte boundary.

CARDB

Card B (O)—programmed through CREG 90h. Usually used to indicate that the current PCMCIA address refers to the second of two memory card slots.

MCBAT1-2

Memory Card Battery (I)—indicate the status of the battery on PCMCIA memory cards. Can be read at SDATA 0A. PCMCIA name is BVD1-2.

MCCD1-2*	Memory Card Detect (I)—both pulled low by the PCMCIA memory card to indicate that the card is properly inserted. PCMCIA name is CD1-2.
MCCE2*	Memory Card Select High (O)—enables the high (odd) bytes for I/O on the data bus. PCMCIA name is CE2.
MCCE1*	Memory Card Select Low (O)—enables the low (even) bytes for I/O on the data bus. PCMCIA name is CE1.
MCRDY	Memory Card Ready (I)—indicates whether the memory card circuits are busy. PCMCIA name is RDY/BSY.
OE0*	Output Enable (O)—indicates to the currently selected PCMCIA card whether it should enable data from memory onto the PCMCIA data bus. PCMCIA name is OE*.
RD15:0	Data (I/O)—connected through buffers to provide a local data bus for the PCMCIA interface.
REFRESH*	Refresh (O)—indicates when the memory card should perform a refresh operation. PCMCIA name is RFSH.
WE0*	Write Enable (O)—indicates to the currently selected PCMCIA card whether it should write data on the RD bus to PCMCIA memory. Also controls buffer direction between the local PCMCIA bus and the RD bus. PCMCIA name is WE/PGM*.

Chip Support for PCMCIA

The F8680 chip manages a 64MB address space, of which the upper 32MB is reserved for memory card access. Memory cycles in this range are always PCMCIA cycles. You can also use low memory for memory card access. The Bank Select Registers allow you to select PCMCIA cycles in any bank of low memory. You might use this access method to implement a system with no on-board RAM. In this case, the user would have to insert a PCMCIA card containing RAM before using the system. In any case, the Bank Select Registers determine whether access will be 8-bit or 16-bit.

The F8680 chip accommodates its 64MB address range through a 26-bit address bus. A25 = 1 indicates that the address is in the upper 32MB of system memory, the area reserved for memory cards. Bit A24 is often used for controlling the memory card interface instead of for addressing. The third implementation described below assumes you will follow this convention, but there is no requirement that you do so. Use the approach that best suits your application.

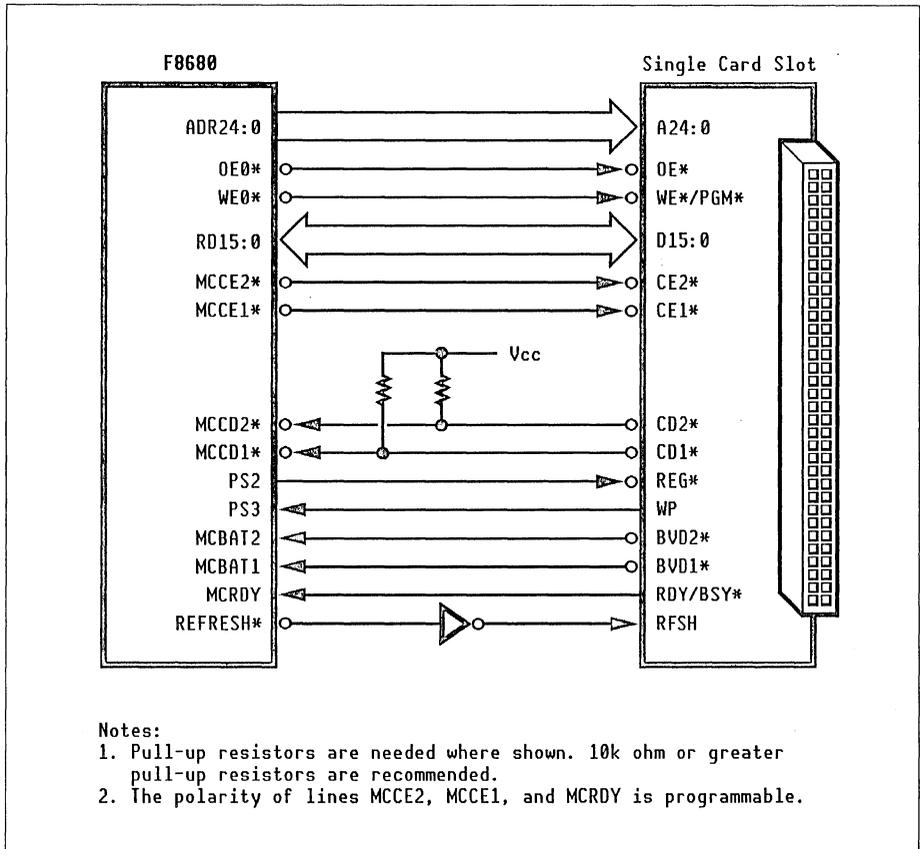
One Card, Bank-Switched from High Memory

When a single card slot is to be bank-switched into low memory as a mass storage device, the following system logic is often used to support PCMCIA interface signals.

- Bit A25 = 1 indicates that the address on the bus is in the upper 32MB of the 64MB address space. This is the area reserved for memory cards. Bit A25 can optionally be used by external logic to enable the card signal buffer.
- Programmable pin PS2 is used to select between the attribute register space and the data space of the card (as selected by the REG signal of the PCMCIA interface).
- The 25 address lines A24:0 are available to access a 32MB data space (as well as a 32MB attribute register space) on the storage card.

Figure 3-8 illustrates the typical connections for a single card slot configuration.

Figure 3-8. Single Card Slot Interface



One Card in Low Memory

For a system where PCMCIA RAM will act as system RAM instead of emulating a mass storage device, the following logic can be used:

- The CS2x line corresponding to the memory bank assigned as PCMCIA memory can optionally be used to enable the memory card enable lines.
- One of the programmable pins is used to select between the attribute register space and the data space of the card (as selected by the REG signal of the PCMCIA interface).
- Any number of address lines A24:0 can be used to address the card.

The connections shown in Figure 3-8 are also valid for implementing a single card slot to be accessed in low memory.

Note: If you were to implement card slots in both low and high memory, you would have to qualify the MCCE1* and MCCE2* signals with the appropriate CS2x line for the card in low memory and with A25 for the card in high memory.

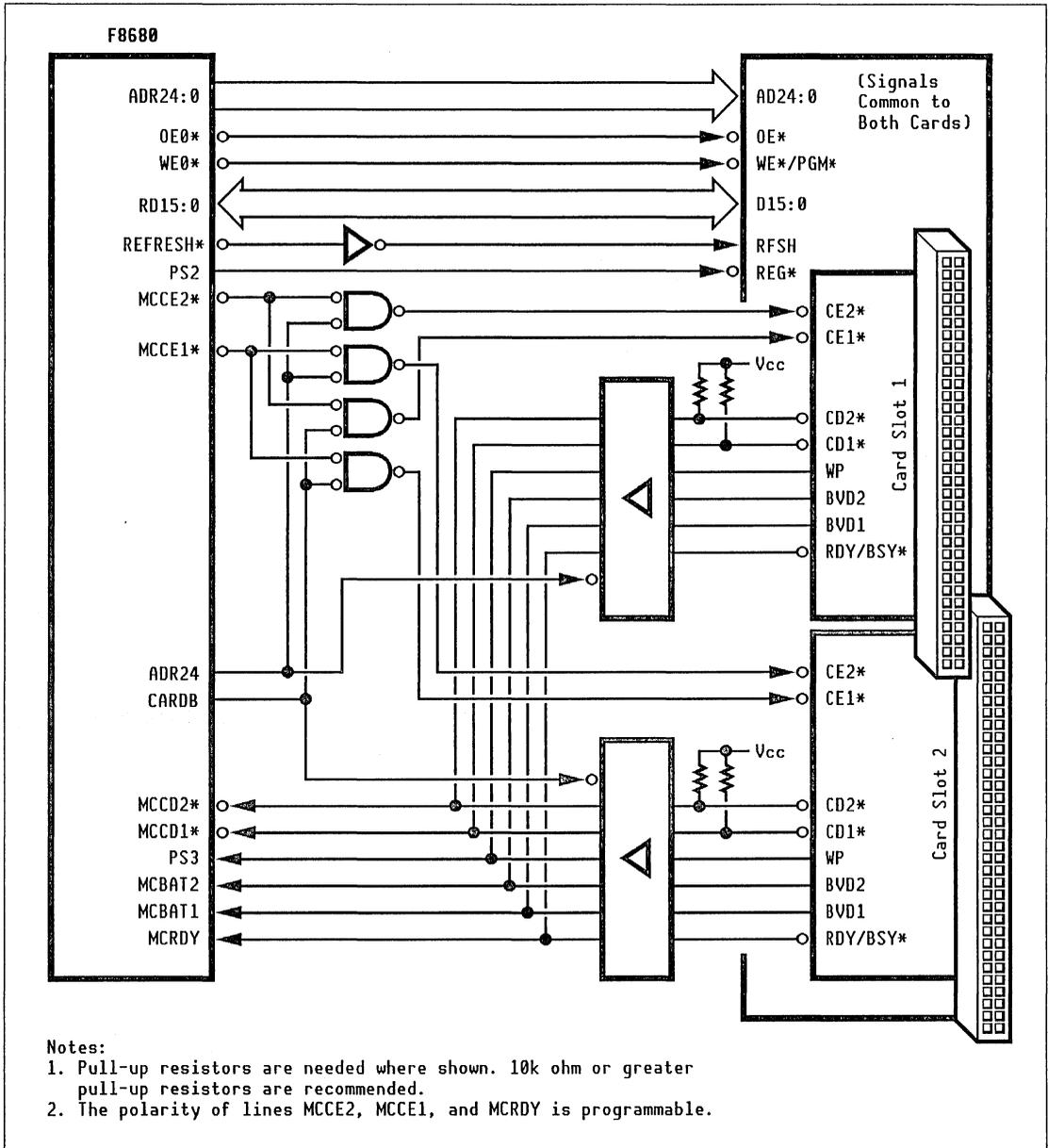
Two Cards, Bank-Switched from High Memory

When support for two card slots is required, a different organization is often adopted.

- Bit A25 = 1 indicates that the address on the bus is in the upper 32MB of the 64MB address space. This is the area reserved for memory cards. Bit A25 can optionally be used by external logic to enable the signal buffer selected by A24.
- Bit A24 is used to select the memory card being accessed. An inverted version of A24 can be programmed to appear on the CARDB signal line. These signals should be used to enable the card 1 and card 2 signal line buffers.
- Programmable pin PS2 selects between the attribute register space and the data space of the card (as selected by the REG signal of the PCMCIA interface).
- The 24 address lines A23:0 provide for a maximum of 16MB data space (as well as 16MB attribute register space) on any one storage card.

Figure 3-9 illustrates the typical connections for a dual card slot configuration.

Figure 3-9. Dual Card Slot Interface



PCMCIA Memory Performance

PCMCIA cards offer good performance. Code can be executed directly from the card, either by means of the bank-switching features of the F8680 chip or by direct addressing when the card is configured as a bank of system RAM in low memory.

Note: If execution directly from a memory card will occur, your system design should include some form of mechanical interlock to prevent accidental removal of the card during execution.

Medium Types

The PCMCIA standard provides for MaskROM, OTPROM, EPROM, EEPROM, Flash-EPROM, and SRAM as supported memory media. These are all defined in versions with 250ns, 200ns, and 150ns access times. 100ns support is also provided, but for SRAM only.

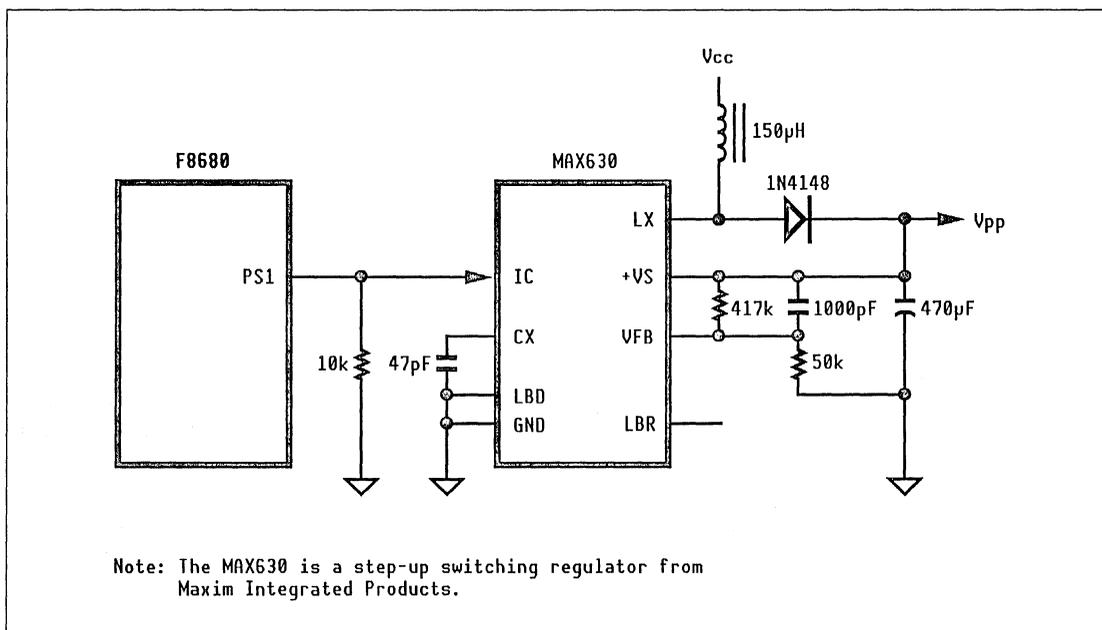
The PCMCIA software driver must initially assume that the slowest card is being used and program the F8680 chip for the longest access time (through CREG13h). The card attribute register space provides information on the true access speed of the card. Only after reading this information can the access time be shortened. If two card slots are provided, the access speed of the slow card is used. CREG 13h is used to make these PCMCIA control signal line settings.

The PCMCIA status lines should be checked before commencing any operation with the PCMCIA memory cards. SDATA 0A is used to return the status of various PCMCIA interface signals.

Circuit for Programmable Cards

If your design will allow for an electrically erasable PROM (EEPROM), you must provide a programming voltage on the V_{pp} pins of the PCMCIA interface connector. You can use the circuit shown in Figure 3-10 with any available programmable pin to control this voltage. The voltage control device shown is capable of generating a 12V programming voltage from a 5V source.

Figure 3-10. Controlling Program Voltage V_{pp} to the Card Slot



Using the Interface as a General Purpose I/O Port

If your design does not include memory card slots, you can use the PCMCIA interface signals to implement a general purpose I/O port. All of the PCMCIA control signals are available to implement a 16-bit port.

See Chapter 8 for an example of the use of the PCMCIA interface to connect an AT-type IDE hard disk drive.

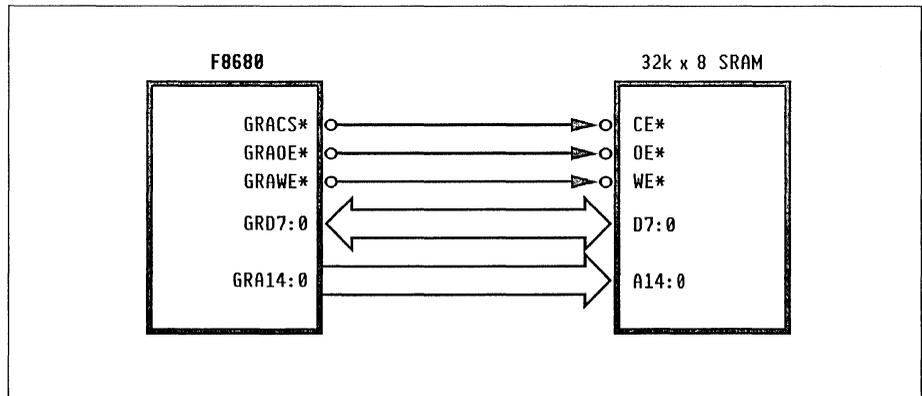
Display

The F8680 display controller options allow you to interface directly to an LCD panel or CRT. Alternatively you can disable the internal display controller and connect an external high-resolution display subsystem.

SRAM Interface

To use the internal display controller you must connect a 32k x 8 SRAM rated for 120ns or better access time. The connection is straightforward and is illustrated in Figure 4-1.

Figure 4-1. Connection of Display SRAM



where: GRA14:0

Graphics Address (O)—address bus to graphics SRAM.

GRACS*

Graphics Chip Select (O)—chip select to graphics SRAM.

GRAOE*

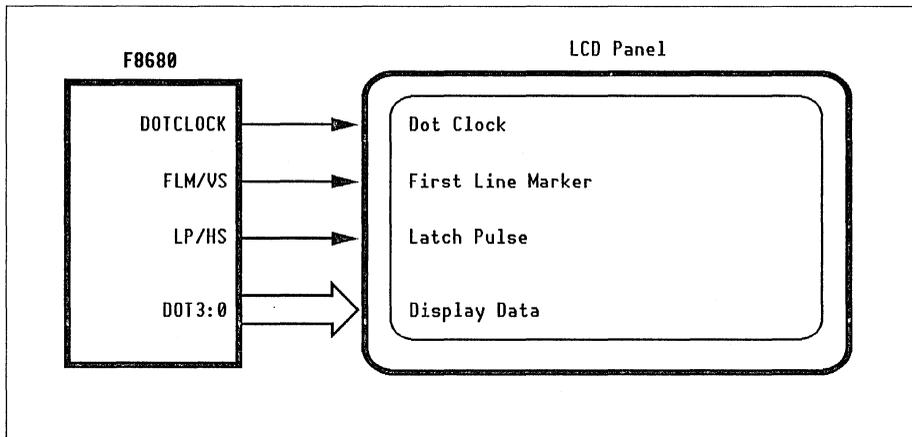
Graphics Output Enable (O)—output enable to graphics SRAM.

GRAWE* Graphics Write Enable (O)—write enable to graphics SRAM.
 GRD7:0 Graphics Data (O)—data bus to graphics SRAM.

LCD Panel Interface

The display controller circuit of the F8680 chip provides the signals shown in Figure 4-2 for connection to a 640 pixel x 200 line LCD panel. These signals operate as indicated only when the F8680 chip is programmed for LCD mode (through CREG 11h).

Figure 4-2. *Display Controller Signals to the LCD Panel*



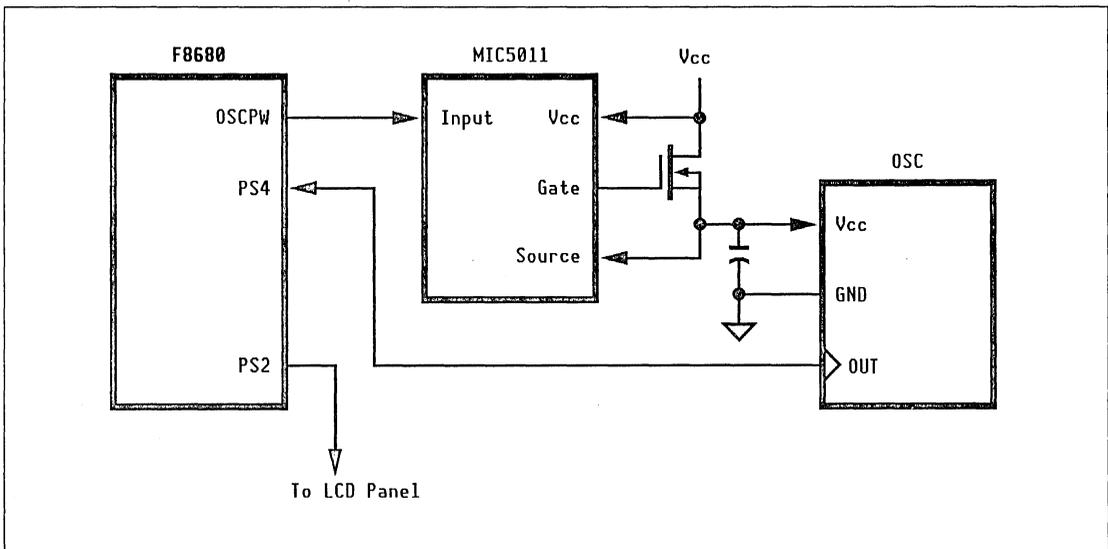
where: DOT3:0 Display Data (O)—pixel output to LCD panels.
 DOTCLOCK Dot Clock (O)—output to LCD panels.
 HS (LP) Horizontal Sync/Latch Pulse (O)—latch pulse signal when in LCD mode.
 VS (FLM) Vertical Sync/First Line Marker (O)—first line indicator signal when in LCD mode.

For panels that require special consideration, the F8680 programmable pins provide two display-related functions: alternative dot clock input and AC drive clock output.

Alternative Dot Clock Input on PS4

You can use an LCD panel that requires a non-standard dot clock rate by providing the appropriate clock input to pin PS4. Figure 4-3 illustrates the necessary connection.

Figure 4-3. *Alternative Dot Clock, AC Drive Clock Connections*



The alternative dot clock function is available only on PS4, and is enabled through code similar to the following:

```
LFEAT 8C,0          ; disable PS4 for output through CREG 8C
LFEAT 11,11001000b ; enable dot clock as PS4, set LCD mode in CREG 11
```

Note that you will need to adjust the byte you write to CREG 11h according to the other features that are also set through that CREG.

ACDCLK Output Option on PS2

LCD panels use an AC drive clock (“M” clock) signal to control the bias polarity of cells in the panel such that none of the pixel cells is subjected to a non-zero average DC bias. Such a bias would cause vertical lines to appear on the display, and could possibly damage the panel. While many LCD panels provide their own on-board circuitry for generating the panel AC drive clock signal, you can accommodate panels without this circuitry through pin PS2. Figure 4-3 illustrates the necessary connection.

The AC drive clock function ACDCLK is available only on PS2 and is programmed through CREG 84h and CRT Controller register index 05 (only when in LCD mode). You might want to use code similar to the following to set up ACDCLK operation:

```
LFEAT 11,10001000b ; set LCD mode in CREG 11
LFEAT 84,01011110b ; enable ACDCLK output on PS2 through CREG 84

MOV DX,03D4 ; write CRT controller index
MOV AL,05 ; index 05 counts LPs per ACDCLK
OUT AL,DX

MOV DX,03D5 ; write CRT controller data
MOV AL,value ; how many LPs per ACDCLK?
OUT AL,DX
```

The *value* that you write depends on the frequency you want at PS2 for use as ACDCLK. The *value* indicates the number of latch pulses (LP) you want to count before toggling PS2, and should be an odd value because the panel uses an even number of lines. You should determine an appropriate *value* by experimenting for the specific panel in use. The AC drive clock signal produced will always have a 50 percent duty cycle.

As an alternative to using the ACDCLK function, you can program a 256Hz frequency output on PS2 and externally gate the signal with the First Line Marker signal VS(FLM) to provide an AC drive clock signal on alternate frames.

Register Programming

Setting up LCD panel operation requires that you first program CREG 0E and CREG 11h. CREG 0E handles the hardware configuration as follows:

- EDC, DCPH, and DCP enable the dot clock signal out of the F8680 chip on the DOTCLOCK pin, and select signal phase and polarity.
- CP selects the signal polarity of the DOT3:0 pixel lines (to determine whether you get a positive or a negative image).
- CHS and CVS select the signal polarity of Latch Pulse signal LP (pin HS/LP) and First Line Marker signal FLM (pin VS/FLM).
- EGOUT enables all display controller output pins, or forces them to ground.
- GENB enables the graphics subsystem, or disables it so that an external display controller can respond instead.

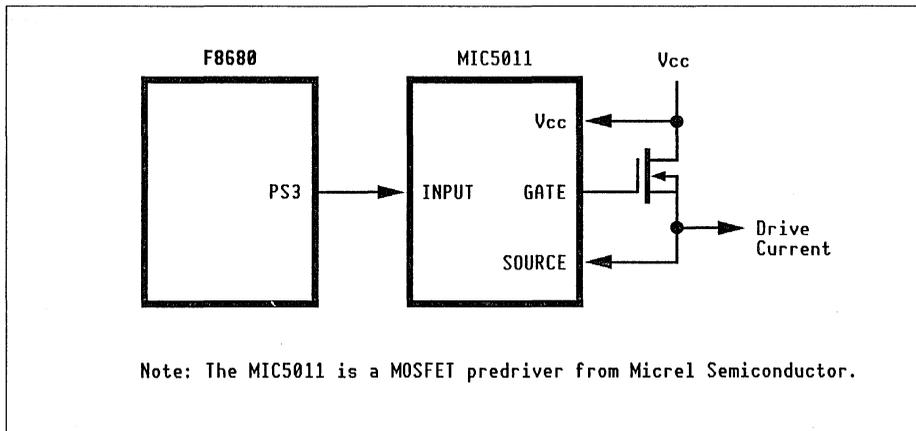
CREG 11h handles system software operational details, so you should not need to consider them in your hardware design. However, note that CREG 11h holds the DCSEL bit that chooses the source of the dot clock, either the internally generated clock or the optional external dot clock input on PS4. CREG 11h also holds the LCD bit, which you must set before LCD panel operation is possible.

The remaining registers, the CRT Controller registers, are all CGA-compatible in their operation.

Controlling the Backlight

If you are incorporating a backlight for the LCD panel in your system design, you can use one of the programmable pins to control it. The PS pins cannot supply the current needed to drive a backlight. However, you can gate an FET or other power-control semiconductor to turn the light on and off. Figure 4-4 illustrates a typical power-control circuit for this purpose.

Figure 4-4. *Power Control Using a Programmable Pin*



Once your hardware design is complete, you can use simple code such as the following to activate the device.

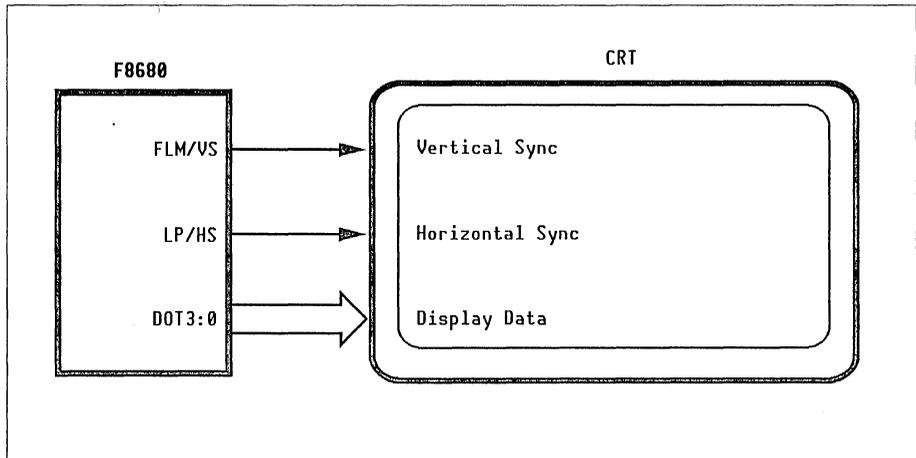
```

LFEAT 80,40      ; set PS1 pin to Vcc
.
.
LFEAT 80,60      ; set PS1 pin to ground
    
```

CRT Interface

The display controller defaults to operation in CRT mode. Figure 4-5 shows the signals available for driving a CRT.

Figure 4-5. *Display Controller Signals to the CRT*



where: DOT3(I)
DOT2(R)
DOT1(G)
DOT0(B)

HS

VS

Display Data (O)—used as output to CRTs.

Horizontal Sync (O)—horizontal synchronization signal to CRT when programmed for CRT mode.

Vertical Sync (O)—vertical synchronization signal to CRT when programmed for CRT mode.

Connecting an External Display Controller

When your hardware design requires a high-resolution display, you can disable the display subsystem of the F8680 chip and provide external display logic.

Disabling the Internal Controller

The graphics controller is enabled through the GENB bit of CREG 0E. Since it defaults to disabled on reset, you can use an external display subsystem without explicitly disabling the internal CGA controller.

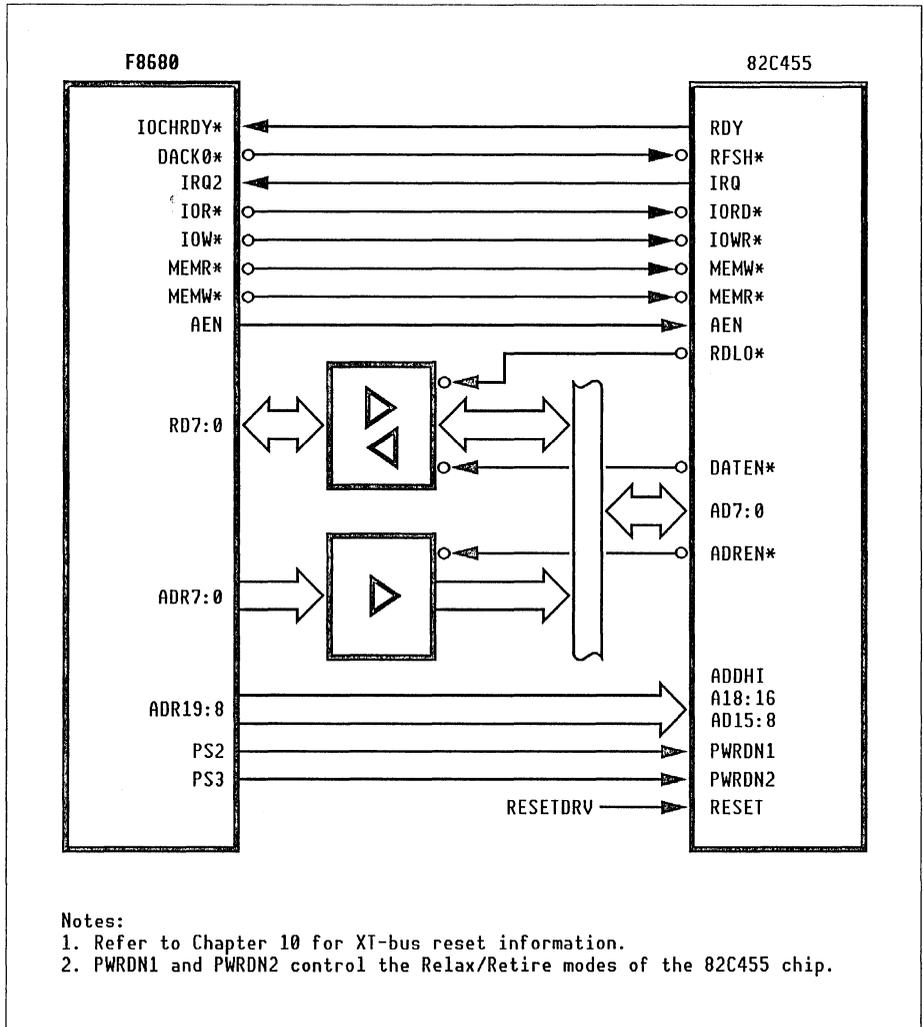
Connecting the 82C455 Low-power VGA Controller

The CHIPS 82C455 Flat Panel/CRT VGA Controller provides full 640 x 480 VGA display resolution on an LCD panel. The controller offers the following features:

- SMARTMAP™ intelligently maps colors to shades of gray in text modes.
- Relax and Retire modes allow software to power-down the LCD panel in stages. In its Retire mode, the current flow through the 82C455 chip is only 10 mA. The Relax/Retire modes of the 82C455 VGA controller can be controlled by the F8680 programmable pins.
- Vertical Compensation fills the entire LCD panel screen when lower-resolution display modes would normally result in reduced display height.

The 82C455 VGA controller is available in a 144-pin plastic flat-pack package. Figure 4-6 illustrates the typical interconnections necessary for incorporating the controller into your hardware design.

Figure 4-6. Connecting the 82C455 VGA Controller

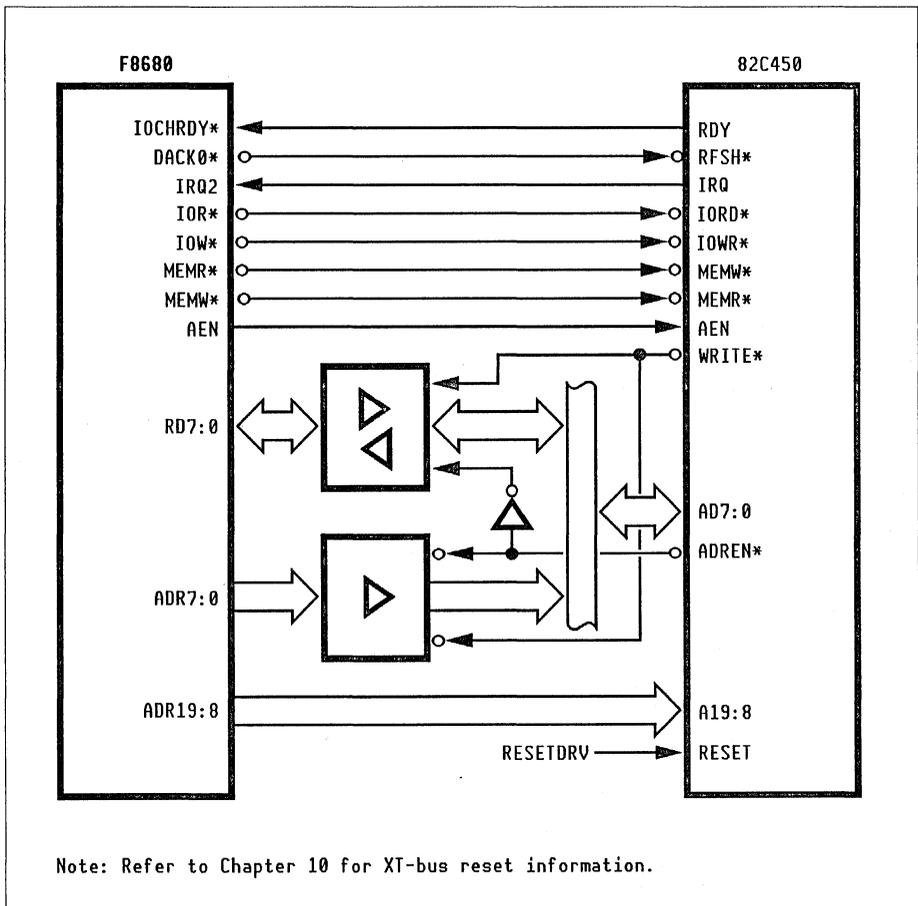


Connecting the 82C450 Super VGA Controller

The CHIPS 82C450 One Megabit DRAM VGA Graphics Controller provides super VGA display resolution (1024 x 768) on a high-resolution CRT when used with 1MB of display RAM. The controller allows you to implement the entire display subsystem in only seven chips, including memory.

The 82C450 VGA controller is available in a 100-pin plastic flat-pack package. Figure 4-7 illustrates the typical interconnections necessary for incorporating the controller into your hardware design.

Figure 4-7. *Connecting the 82C450 VGA Controller*



Power Management

The F8680 chip provides many hardware and software features that allow you to design an effective yet unobtrusive power management mechanism into your system. The most basic level of management involves only two chip modes:

- The F8680 chip is in *active mode* when it is powered and its state machines for timing are running.
- The F8680 chip is in *suspend mode* when only the 32kHz clock is running. The chip continues to keep time and maintain configuration parameters while in suspend mode.

Power must be maintained to the F8680 chip at all times. This should not be a problem given that the F8680 chip consumes negligible power when in suspend mode.

When terms such as “power on,” “power up,” and “power down” are used in the following sections, they do not refer to the actual application of power to and removal of power from the F8680 chip. Rather, these terms refer to the transition between active mode and suspend mode.

Power Control Issues

You must decide the aspects of power consumption you need to control and monitor. Consider the following issues.

- Your design should leave Vcc connected to the F8680 chip at all times. Likewise, the 32kHz clock should run at all times. This arrangement is sufficient to keep all configuration information active in the F8680 chip and maintain the time-of-day clock.
- The PWRUP input commands the power-up of the F8680 chip into active mode and can initiate power-down of the F8680 chip into suspend mode. You can command power-up or power-down through either a pulse or a steady signal sense.

- The F8680 chip provides the OSCPW output to control power to system oscillators. The power control state machine of the F8680 chip sets OSCPW active a half-second in advance of enabling the timing state machines.
- You may want to control the UART oscillator separately, through a programmable pin. This allows you to turn off the oscillator when the UART is not in use.
- If your design includes an LCD panel with a backlit screen, you may want to use a programmable pin to control the backlight.
- Your design might include a battery voltage monitor, which could be anything from a simple low-voltage comparator that provides a battery “OK/LOW” signal to an analog-to-digital (A/D) converter that returns the battery voltage with an accuracy of a fraction of a volt.

The SuperState R mode of the F8680 chip makes it easy to perform all of these power control and monitoring functions, as described in the following sections.

Software Power-Off Control

The power control state machine can be instructed to remove power from the oscillators through the POFF bit at CREG 1C. When POFF is set to 1, the power control state machine will begin to sequence the F8680 chip into suspend mode. Refer to the *F8680 PC/CHIP Programmer's Reference Manual* for details of this operation.

Power-On Clock Comparator

You should be aware that power-up can also be commanded through the clock comparator register of the F8680 chip. When in suspend mode, power will be restored when the time in the power control comparator matches the time-of-day count value. Operation will resume regardless of the PWRUP input signal level. CREGs 18h through 1B comprise the 32-bit comparator register.

If the F8680 chip is already active when the time-of-day reaches the comparator value, the comparator will have no effect. Refer to the *F8680 PC/CHIP Programmer's Reference Manual* for details of this comparator.

Power Control State Machine

The logic provides a state machine that sequentially powers up and powers down the F8680 chip. The power control sequencing operates as follows.

- Upon receiving a power-up request (initiated by either the PWRUP input or the power control comparator), the power control state machine sets the OSCPW output active within 0.5s. It enables the various timing state machines in the F8680 chip exactly 0.5s after setting OSCPW active.
- Upon receiving a software-commanded power-down request (the only kind possible), the power control state machine disables the various timing state machines within 0.5s. It sets the OSCPW output inactive exactly 0.5s after disabling the timing state machines.

Therefore, power-up and power-down sequences always require at least 0.5s, but always less than 1.0s, to execute.

Connections to the PWRUP Input

The power-up pin PWRUP indicates when the F8680 chip should exit suspend mode and begin normal operation. Once the F8680 chip is operational, the PWRUP signal can be used to initiate a power-down into suspend mode.

Note: PWRUP by itself cannot cause a power-down; it can only *request* a power-down. Software must command a power-down. However, PWRUP can be programmed to cause a switch to SuperState R mode, which in turn can power down the F8680 chip.

You generally have two choices when incorporating a power switch in your system design: a toggle switch or a momentary switch. The SuperState R configuration register at CREG 1C provides a bit to select whether a high or a low signal on PWRUP will cause a switch to SuperState R mode. The programming of this register will depend on your choice of power-up switch.

Figure 5-1 illustrates how an SPDT toggle switch is connected so that the PWRUP input is pulled to logic ground to turn the system off, and is pulled up to Vcc to activate the F8680 chip. You must provide software that recognizes a low signal at PWRUP as a power-down request.

Figure 5-1. *Connecting PWRUP to a Toggle Switch*

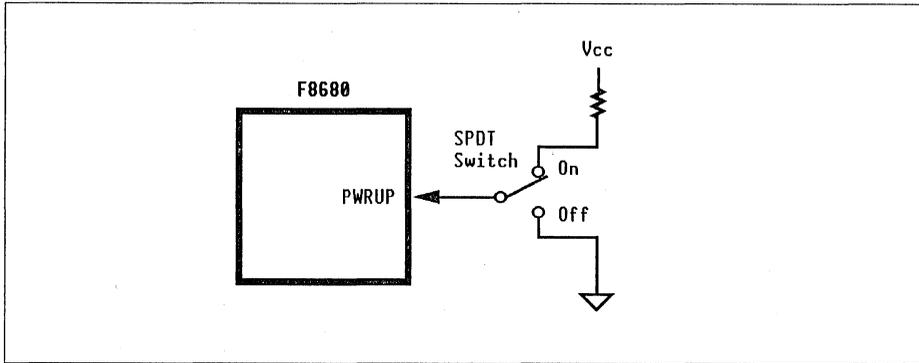
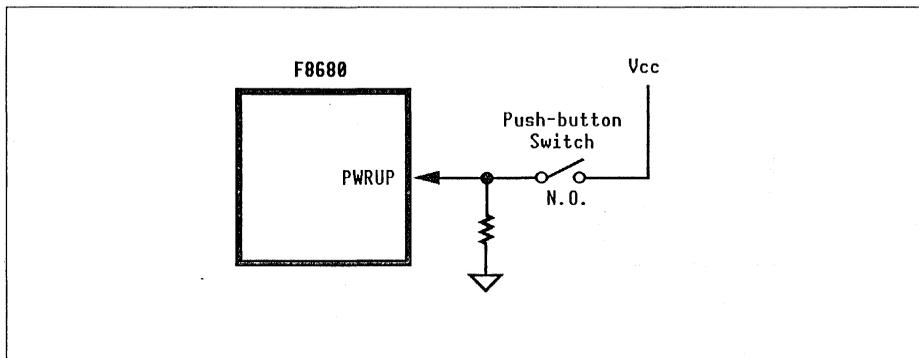


Figure 5-2 illustrates how a momentary switch pulls PWRUP to Vcc to activate the F8680 chip, allowing PWRUP to go back to logic ground as soon as the momentary switch is released. You must provide software that recognizes a high pulse at PWRUP as a power-down request.

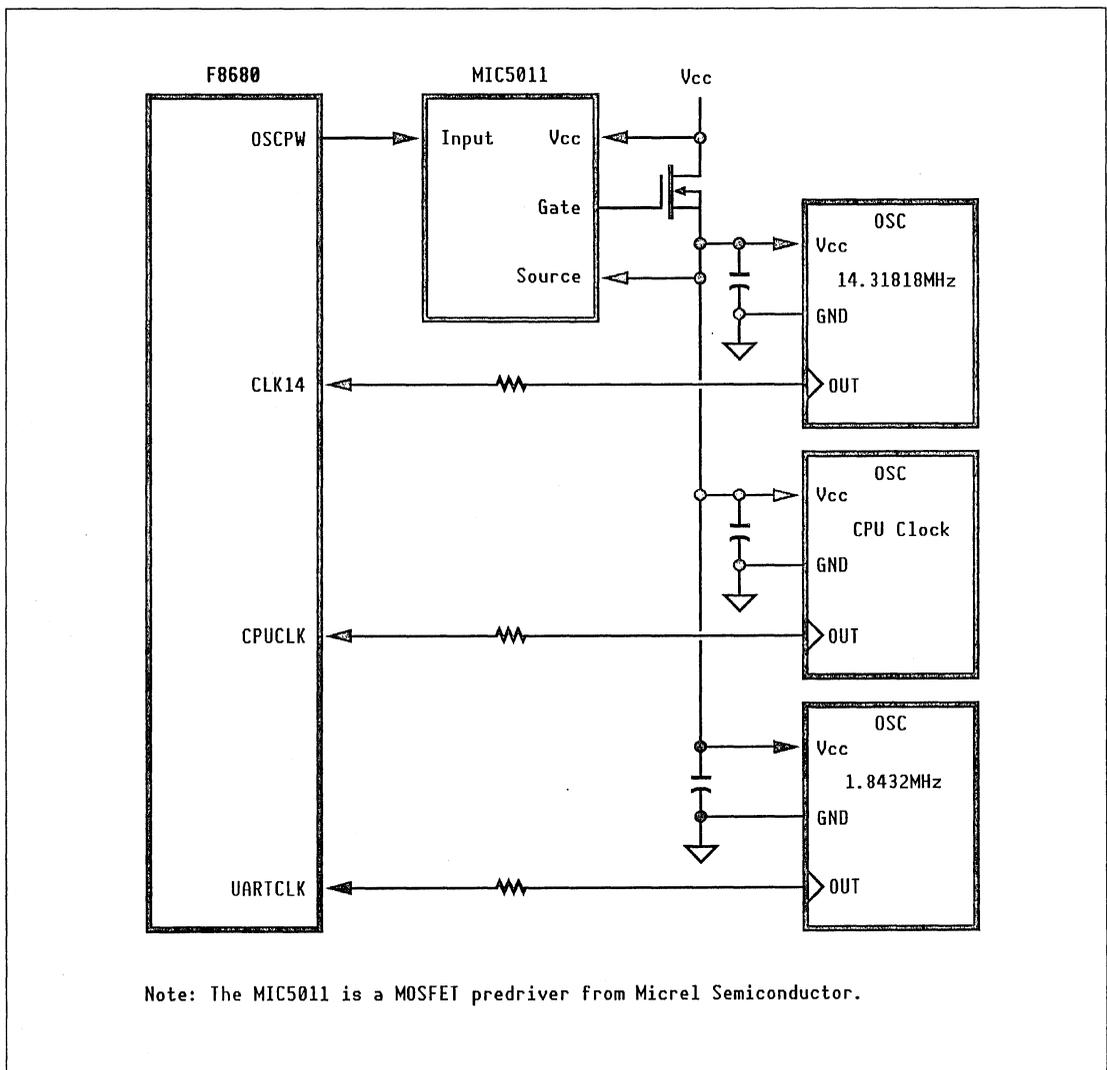
Figure 5-2. *Connecting PWRUP to a Momentary Switch*



Using OSCPW to Drive System Oscillators

The power control state machine activates the signal OSCPW as part of the start-up sequence. You should use OSCPW to control power to all system oscillators except the 32kHz clock input. However, you may want to provide separate control for oscillators that need not run constantly (such as the UART clock). Figure 5-3 illustrates how you might control oscillator power using OSCPW.

Figure 5-3. Using OSCPW to Control Power to the Oscillators



Signals Available for Control/Status

You have a large number of lines available for peripheral device control and status monitoring. In addition to the obvious choices (such as the programmable pins) there are some less evident possibilities that could save you external logic.

Programmable Pins

You can use the programmable pins in a variety of ways. Software control makes these pins extremely flexible and easy to incorporate in your design. You need to know that the pins can operate as either inputs or outputs and can source/sink 4mA minimum (see the *F8680 PC/CHIP Data Sheet* for specific values) when used as outputs. Table 5-1 lists the specific functions available through the programmable pins.

Table 5-1. Programmable Pin Functions

Function	Type	Available on:				
		PS1	PS2	PS3	PS4	CARDB
CLK32	O	x	x	x	x	x
CLK1/16	O	x	x	x	x	x
CLK1/32	O	x	x	x	x	x
PCS	O	x	x	x	x	
FR0	O	x	x			
512Hz	O	x				
256Hz	O		x			
32768Hz	O			x		
16384Hz	O				x	x
IFACTIVE	O		x	x	x	
CLKCANRUN	O			x		
FLAG9	O			x	x	
ACDCLK	O		x			
IMPLT	O			x		
CLKCANRUN	O				x	
MAD24	O					x
EXTSS	I	x	x	x	x	
DOTCLK	I				x	

<i>where:</i> CLK32	32kHz Clock—PS1-PS4 and CARDB can provide a 32kHz clock pulse.
CLK1/16	1/16 32kHz Clock—PS1-PS4 and CARDB can provide a clock with a positive pulse that occurs once every sixteen 32kHz clock cycles. The programmable pin stays high only for the duration of one phase (1/2 clock period) of the 32kHz clock.
CLK1/32	1/32 32kHz Clock—PS1-PS4 and CARDB can provide a clock with a positive pulse that occurs once every thirty-two 32kHz clock cycles. The programmable pin stays high only for the duration of one phase (1/2 clock period) of the 32kHz clock.
PCS	Programmable Chip Select—PS1-PS4 can decode I/O reads and/or writes at any I/O address or range of addresses.
FR0	Frame Rate 0—PS1-PS2 can provide the vertical interval signal divided by 2.
256Hz 512Hz 16384Hz 32768Hz	These frequencies can be output on PS2, PS1, PS4, and PS3 respectively.
IFACTIVE	Instruction Fetch Active—PS2-PS4 can indicate whether the current memory request is for instruction data.
ICMPLT	Instruction Complete—PS3 can output a pulse at the end of each instruction executed. This signal can be counted to indicate true MIPS. PS3 stays high only for the duration of one phase (1/2 clock period) of the CPU clock.
FLAG9	Flag bit 9—PS3-PS4 can indicate whether maskable interrupts are enabled.
ACDCLK	AC Drive Clock—PS2 can output a square wave with a 50 percent duty cycle and a programmable period.
CLKCANRUN	Clock Can Run—PS4 can indicate whether the power control state machine has completed the power-up sequence and instruction execution is allowed.
MAD24	Memory Address bit 24—CARD B can output an inverted version of address bit ADR24 so that external logic can select the memory card being addressed.
EXTSS	External SuperState R Switch—PS1-PS4 can be used as inputs to trigger a switch to SuperState R mode.
DOTCLK	Dot Clock—PS4 can be used to input a non-standard dot clock frequency for the graphics controller.

Programming Simultaneous Input and Output

If you program a pin to input an external SuperState R switch request, you can still use that pin for output. For example, you could select a 256Hz output on PS2 and program PS2 for SuperState R input at the same time. This arrangement would result in a periodic switch to SuperState R mode (like the timer ticks provide). However, PS2 should not be driven by external circuitry in this configuration.

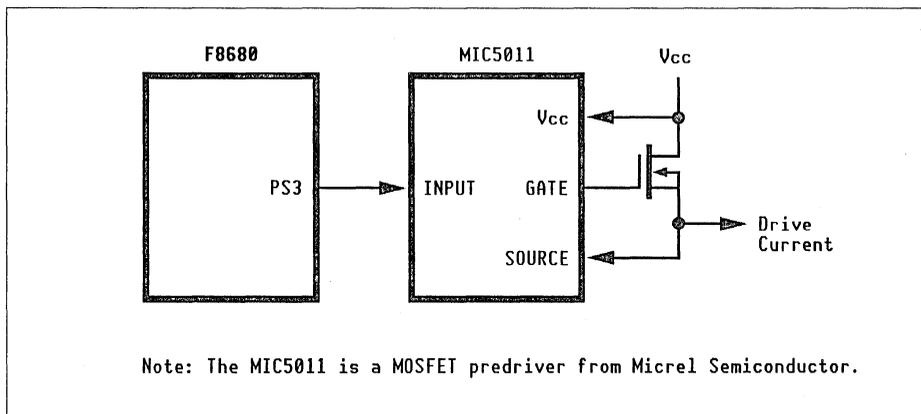
Dynamic Reprogramming

You can connect more than one device to the same programmable pin and reprogram the pin output as often as necessary to accommodate the devices. For example, you could connect two devices: one that generates an interrupt pulse and one that requires a clock input each time an interrupt pulse is generated. Your code would initially set the programmable pin to act as an input. Then when an interrupt is received, your code would switch the programmable pin to a square wave output function. The CLOCK line of an AT-style keyboard uses this type of signalling.

Controlling Power to Peripheral Devices

You can use the various control pins provided by the F8680 chip to control power to peripheral devices. You will need a power control circuit or device such as that illustrated in Figure 5-4.

Figure 5-4. *Connecting a Power Control Device*



Monitoring Battery Voltage

If your system design provides for battery power, SuperState R mode makes it easy to monitor the battery condition. In general, you have several options such as a simple “battery low” indication, a two-level “battery low” indication, or a direct voltage read capability.

Figure 5-5 illustrates a simple circuit to signal a “battery low” condition. The output of the circuit is connected to a programmable pin. You must program the pin so that it causes a switch to SuperState R mode when active.

Figure 5-5. Simple “Battery Low” Indicator Circuit

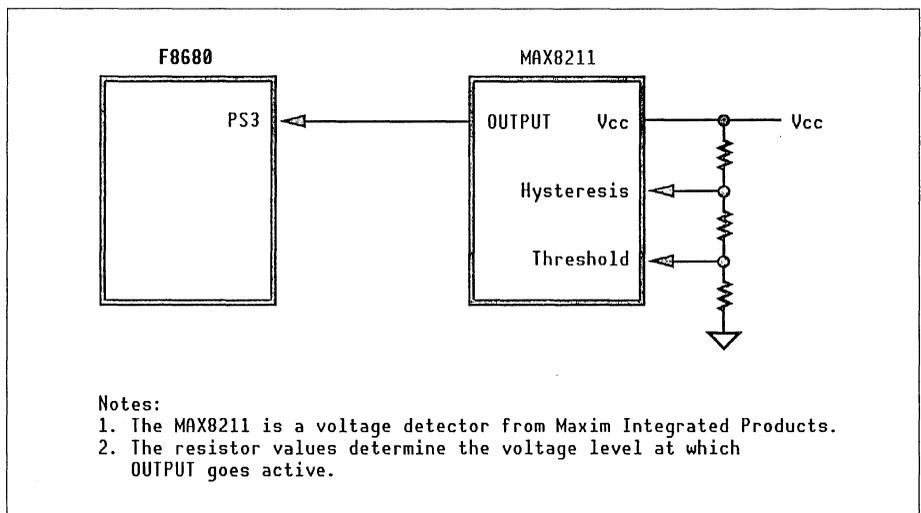


Figure 5-6 illustrates a more useful two-level circuit. At one voltage level the circuit signals “battery low”; at the second level, it signals “battery dangerously low.” Your SuperState R code can respond to this second level of warning according to your design requirements.

Figure 5-6. Two-Level “Battery Low” Indicator Circuit

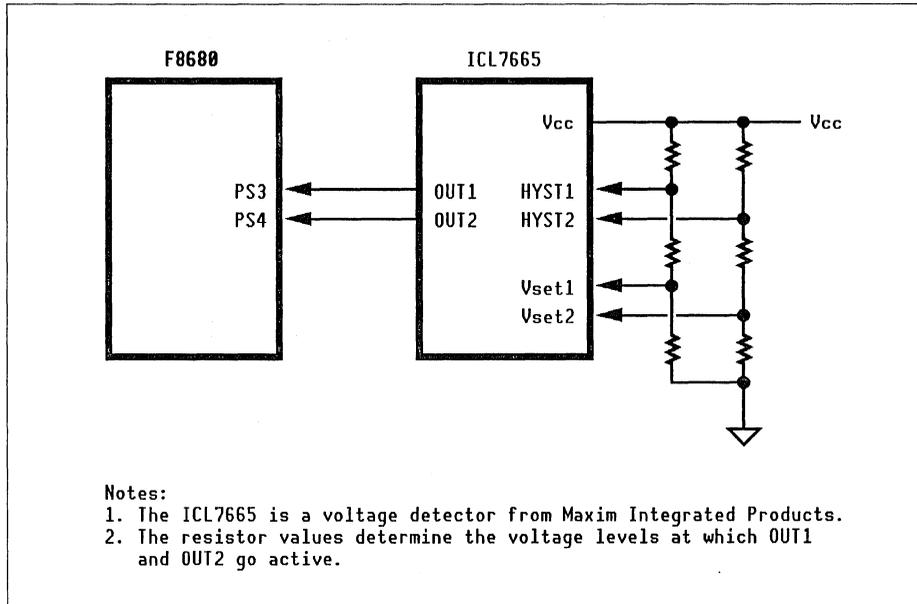
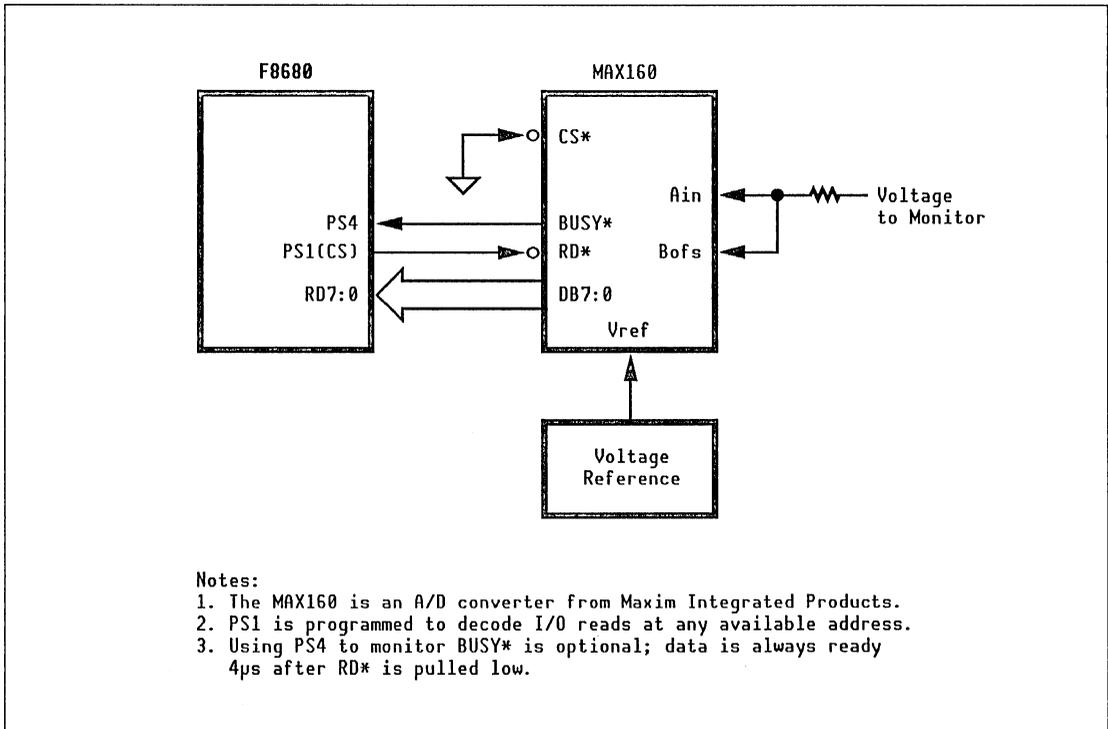


Figure 5-7 shows how you might incorporate an analog-to-digital (A/D) converter into your design. Your SuperState R code checks the voltage level indicated by the A/D converter on a periodic basis.

Figure 5-7. Battery Voltage Monitor Circuit with A/D Converter



Power-Saver Mode

The F8680 chip provides a feature known as performance control to reduce power consumption during off-peak times. Performance control inserts delays between instructions in order to reduce the number of accesses made to DRAM. This is a software feature and does not impact your hardware design.

Refer to Appendix A for a brief discussion of performance control, and to the *F8680 PC/CHIP Programmer's Reference Manual* for a more detailed explanation.

Keyboard and Mouse

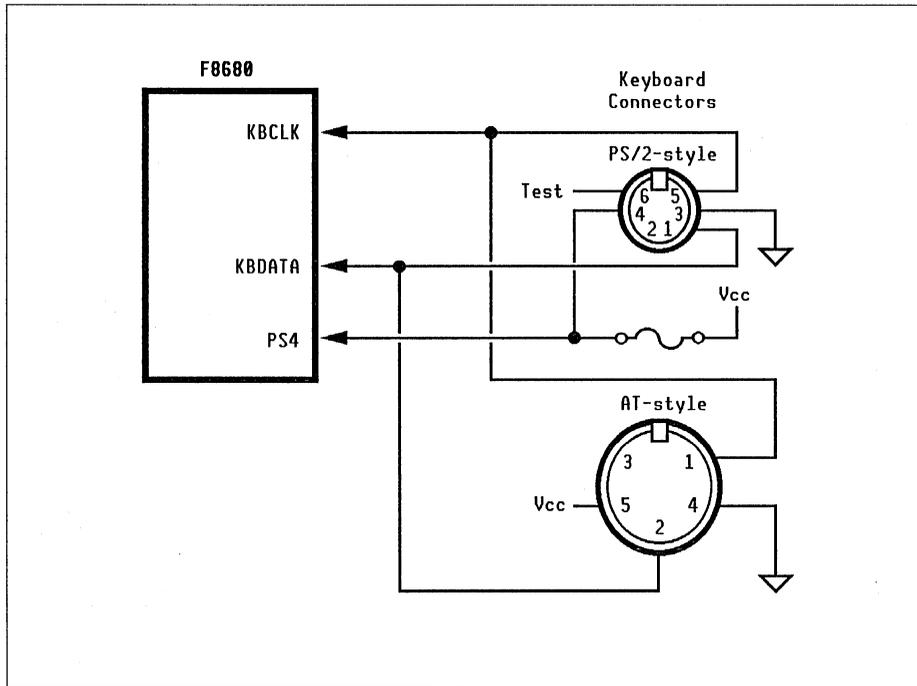
The F8680 chip provides a standard keyboard interface circuit for XT-type keyboards. The XT keyboard communicates over a serial interface using a 9-bit transmission format, composed of a single start bit and eight data bits. Most PC keyboards default to this transmission format at power-on initialization time.

A keyboard generally must provide its own internal processor for scanning the keys and serializing the key scan code for transmission to the system. However, for a portable computer with an integral keyboard, the requirement for a separate key scan processor puts an additional burden on system design. Fortunately, you can take advantage of SuperState R mode to perform keyboard scanning and eliminate the need for a separate key scan processor. When you let SuperState R code do the scanning, the keyboard interface circuit is not used.

Connections to the XT Keyboard

The XT keyboard connects to the system with only a clock and a data line, and additionally requires an operating voltage of +5V. Figure 6-1 illustrates the connections you would typically make.

Figure 6-1. Connections to the XT Keyboard



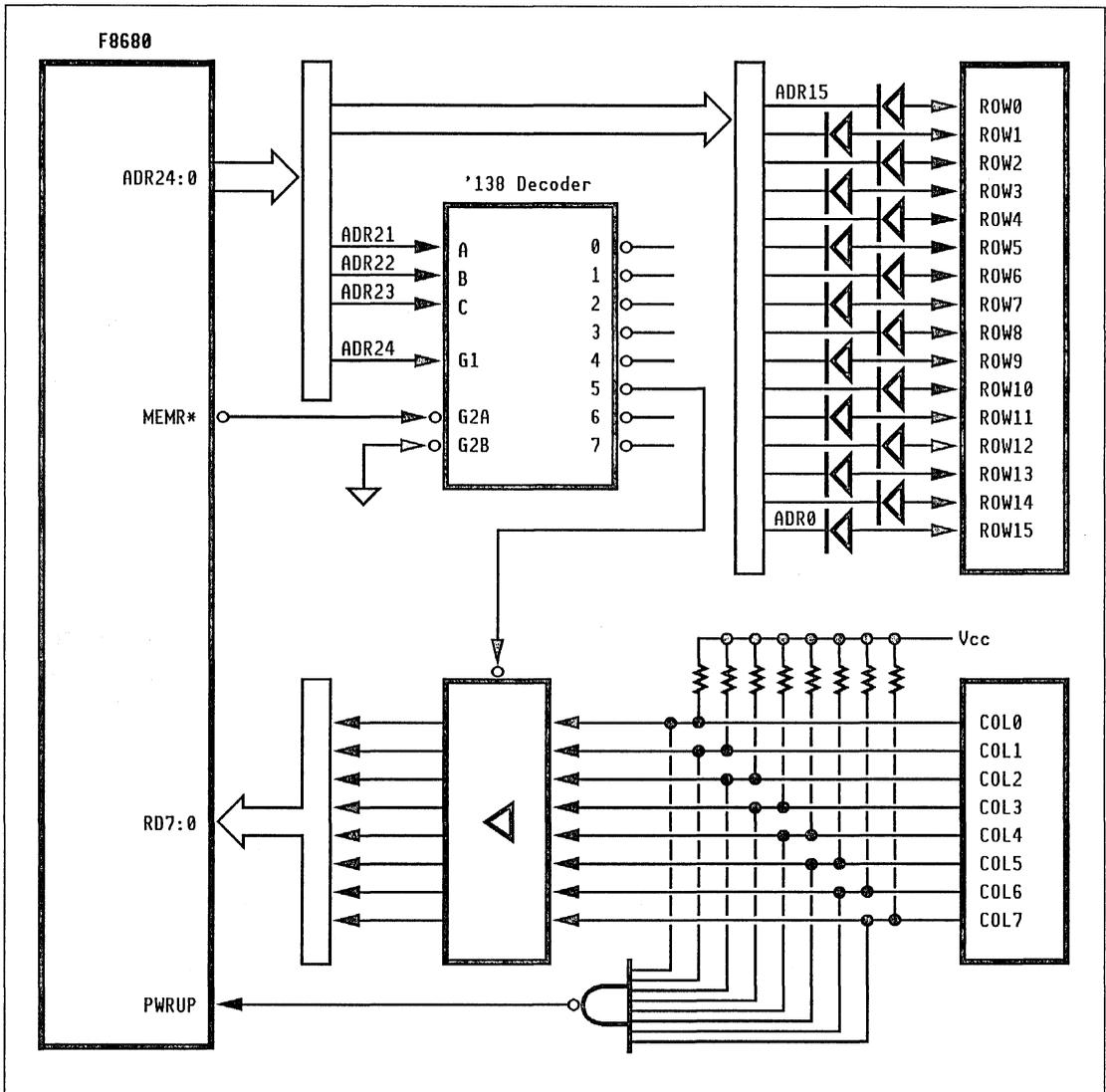
Direct Scanning of an Integral Keyboard

The keys of a standard keyboard are arranged in row and column order as a matrix of switches. The keyboard processor scans the keyboard matrix by pulsing one of the row lines and monitoring all of the column lines to see if a pressed key has completed a connection. If so, the keyboard processor determines the key that is pressed according to the row pulsed and the column on which the pulse appeared. The keyboard processor then monitors this key for a short period of time to ensure that it was pressed. Finally, the keyboard processor sends the scan code of the key to the system.

With the help of SuperState R mode and the system address and data lines, you can design a circuit that performs this same scanning and monitoring function. Your emulation code can then provide the system with XT or AT scan codes, because at this point the keyboard emulation becomes a function of software alone.

Figure 6-2 illustrates the type of connections you could use to implement keyboard scanning through the F8680 chip. Note that the built-in keyboard interface circuit is not involved in this scheme. By scanning for keystrokes directly, you bypass the keyscan/parallel-to-serial/serial-to-parallel process that is needed when an external keyboard is used.

Figure 6-2. Connections to Keyboard Matrix for Direct Scanning



Keyboard Scanning Principles

Keyboard scanning operates according to certain basic principles that mimic the operation of a dedicated keyboard processor.

A key scanning routine must be in place in the SuperState R code and must execute on a periodic switch to SuperState R mode. The periodic switch can be commanded either by the timer tics or by programming a PS pin to generate periodic switches to SuperState R mode (as explained in the “Programmable Pins” section of Chapter 5).

The key scanning routine performs a memory read cycle to an address in high memory. Bits ADR15:0 of the memory read address must be all zeros. If any keyboard key happens to be pressed, making contact between a ROW line and a COL line on the keyboard scanning matrix, one of the COL lines will be pulled to zero as well.

Address bits A24:21 of the memory read address are set so they will activate the decoder according to the decoder output line you select for your hardware design. Bits A23:21 decode to one of eight selections, as shown in Table 6-1. Your hardware design could include jumpers to make this decoded range hardware-selectable. The circuit in Figure 6-2 uses decoder output line 5 as an example, so address 010A0000 must be used to scan the keyboard.

The decoder will activate the data buffer, and any COL line pulled low will appear as a zero on the data bus RD7:0. From the perspective of the key scanning routine at this point, if a memory read at 010A0000 (in the Figure 6-2 example) returns a data byte value other than FF, then a key has been pressed. The routine must now determine which key.

By performing successive memory reads at addresses shown in Table 6-2, the routine can effectively pull each of the 16 ROW lines low one at a time. When any of these reads returns a value other than FF, the key scanning routine has determined both the ROW and COL that are connected and can thus identify the pressed key.

Figure 6-2 also includes a gate for powering the system when a key press is detected. This gate is optional. Once the system is powered, the gate output will continue to toggle as keys are struck. Therefore, you must provide either SuperState R code that handles the PWRUP switch request appropriately or extra logic that disables the gate when the system is active.

Other Keyboard Emulation Issues

There are many other issues involved in properly emulating keyboard input. For example, the key scanning routine must check several times to ensure that a key is pressed (to “debounce” the contacts). The routine must check for other pressed keys and determine which key contact to validate, as well as provide for an “n-key rollover” feature (to ignore a key that has been held pressed if other keys have been subsequently struck). These are all software issues and do not impact your hardware design, other than your prudent assignment of a key order to the ROW/COL scanning layout so that adjacent keys do not share the same ROW or COL line. In such a case, accidentally striking two adjacent keys would leave the key scanning routine unable to determine the second key.

Table 6-1. *Possible Address Ranges for Keyboard Scan Decoding*

A24:21	Address	Decoder Output Line
1000	0100 0000	0
1001	0102 0000	1
1010	0104 0000	2
1011	0106 0000	3
1100	0108 0000	4
1101	010A 0000	5
1110	010C 0000	6
1111	010E 0000	7

Table 6-2. Address Sequence to Pulse Each ROW Line Low

Memory Read Address	Low-Order Bits	Row
010AFFFE	1111 1111 1111 1110	0
010AFFFD	1111 1111 1111 1101	1
010AFFFB	1111 1111 1111 1011	2
010AFF7	1111 1111 1111 0111	3
010AFFEF	1111 1111 1110 1111	4
010AFFDF	1111 1111 1101 1111	5
010AFFBF	1111 1111 1011 1111	6
010AFF7F	1111 1111 0111 1111	7
010AFEFF	1111 1110 1111 1111	8
010AFDFF	1111 1101 1111 1111	9
010AFBFF	1111 1011 1111 1111	10
010AF7FF	1111 0111 1111 1111	11
010AEFFF	1110 1111 1111 1111	12
010ADFFF	1101 1111 1111 1111	13
010ABFFF	1011 1111 1111 1111	14
010A7FFF	0111 1111 1111 1111	15

AT Keyboard Emulation Considerations

The Virtual I/O feature provides a means of simulating the presence of the keyboard controller circuit provided on the PC/AT®. Because Virtual I/O can trap all port accesses, simulating the keyboard controller operation involves only software, not hardware.

This keyboard controller is an intelligent device, as opposed to the simple keyboard interface circuit on the XT. The controller on the AT accepts a series of commands, issued through I/O address 064. A separate status register can be read at this address as well. You can provide SuperState R code to intercept I/O reads and writes in the 060-064 range, which allows you to substitute the appropriate values for an AT-type controller.

Refer to the *IBM AT Technical Reference* for information on the register bits and commands you will have to consider.

Providing for a Mouse

The F8680 chip makes no provisions for a mouse, but there are many options available for connecting one. You must first choose among the mouse types available:

- A serial mouse connects directly to the RS232-C serial port. If you take this route, you need only provide line drivers and receivers as described in Chapter 9. The serial mouse is not an “intelligent” device. The system will spend a moderate amount of available operating overhead dealing with the serial mouse.
- A standard bus mouse requires an interface circuit on the XT bus. Like the serial mouse, the standard bus mouse is not an intelligent device.
- A PS/2[®]-compatible mouse is an intelligent device. It sends information on mouse activity in a compact format and requires little operating overhead. If you use this approach, you should employ the CHIPS 82C710 Universal Peripheral Controller described in Chapter 9. The 82C710 provides a PS/2-compatible mouse interface circuit.

The best mouse choice depends on your specific design application.

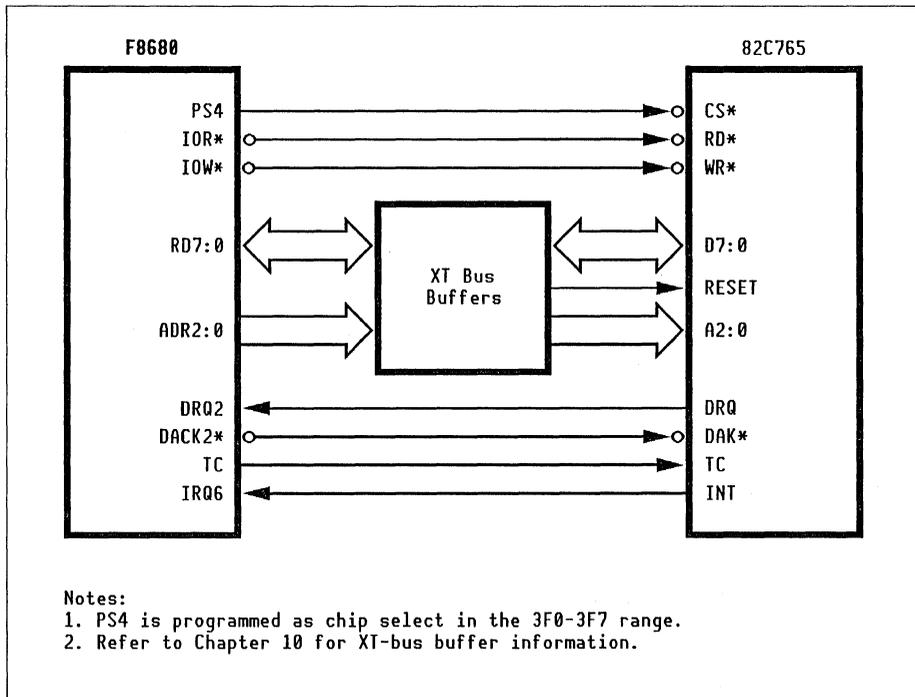
Floppy Disk Drives

The standard XT-compatible floppy disk drive interface always requires a floppy disk controller, which has to be compatible with the NEC 765 Floppy Disk Controller (FDC), and a data separator circuit. The most practical way to incorporate a floppy disk in your system design is to use the CHIPS 82C765 Floppy Disk Controller, which houses both a 765-compatible controller and an analog data separator circuit. If your design also provides for a parallel interface, an Integrated Drive Electronics (IDE) hard disk drive, or a mouse, you may want to use the CHIPS 82C710 Universal Peripheral Controller instead, as it incorporates all of these circuits.

Connecting the 82C765 Floppy Disk Controller

The floppy disk controller circuit is an XT-bus peripheral. Refer to Chapter 10 for information on the XT bus. Figure 7-1 indicates the required connections. Note that these are all standard XT bus signals except for CS*, which you implement by programming one of the PS pins as a programmable chip select in the I/O address range 3F0-3F7h.

Figure 7-1. Floppy Disk Drive Interface Signals



- where:
- A2:0 Address bits 2:0 (I)—used to decode the specific FDC register to be read or written.
 - D7:0 Data bits 7:0 (I/O)—used to transfer commands and data between the system and the FDC, one byte at a time.
 - INT Interrupt Request (O)—interrupt to request system service (on IRQ6 for XT compatibility).
 - RESET Reset (I)— XT bus reset signal.
 - RD* Read (I)—IOR* signal from XT bus.
 - WR* Write (I)—IOW* signal from XT bus.
 - CS* Chip Select (I)—decode of I/O addresses in the 3F0-3F7 range.
 - DRQ DMA Request (O)—signal to the system that a byte must be transferred on D7:0. Connected to DRQ2 for XT compatibility.

DAK*	DMA Acknowledge (I)— signal from the system that the DMA request made on DRQ has been accepted.
TC	Terminal Count (I)— indication from the system that the current DMA transfer is the last one programmed.

The programming required to operate PS4 as a chip select is minimal. You may want to use an instruction sequence similar to the following:

```
LFEAT 8C,01100100b ;output driven and active low, program for chip
                    ;select function
LFEAT 8E,F0         ;lower part F0 of cs address 3F0
LFEAT 8F,7F         ;upper part 3 of cs address 3F0, enable for reads
                    ;and writes, mask lower 3 bits
```

This sequence sets PS4 to act as an active low chip select in the range 3F0-3F7.

Floppy Disk Controller Registers

The I/O address range of 3F0-3F7h must be decoded as the chip select signal to the FDC. The FDC further decodes I/O reads and writes by using XT bus address bits A2:0, as shown in Table 7-1. Refer to the *CHIPS 82C765 Floppy Disk Controller Data Sheet* for programming information on these registers.

Table 7-1. Register Names and I/O Addresses

I/O Address	Type	Register Name
3F2	W	Drive Control Register
3F4	R	Main Status Register
3F5	R/W	Data Register
3F7	W	Data Rate Register
3F7	R	Disk Changed Register (bit 7 only)

Connecting the 82C710 Universal Peripheral Controller

The CHIPS 82C710 Universal Peripheral Controller provides the same core logic as the 82C765 controller for managing a floppy disk drive. Refer to Chapter 9 for interface information on the 82C710 component.

Hard Disk Drives

Incorporating a hard disk drive into system designs is a straightforward process when you select a drive with integrated drive electronics (IDE). The choice of an IDE drive frees you from the details of interfacing to a hard disk controller and then to a drive, as the IDE drive connects directly to the XT bus. Your design must provide only a single chip select signal, which is easily handled through a programmable pin.

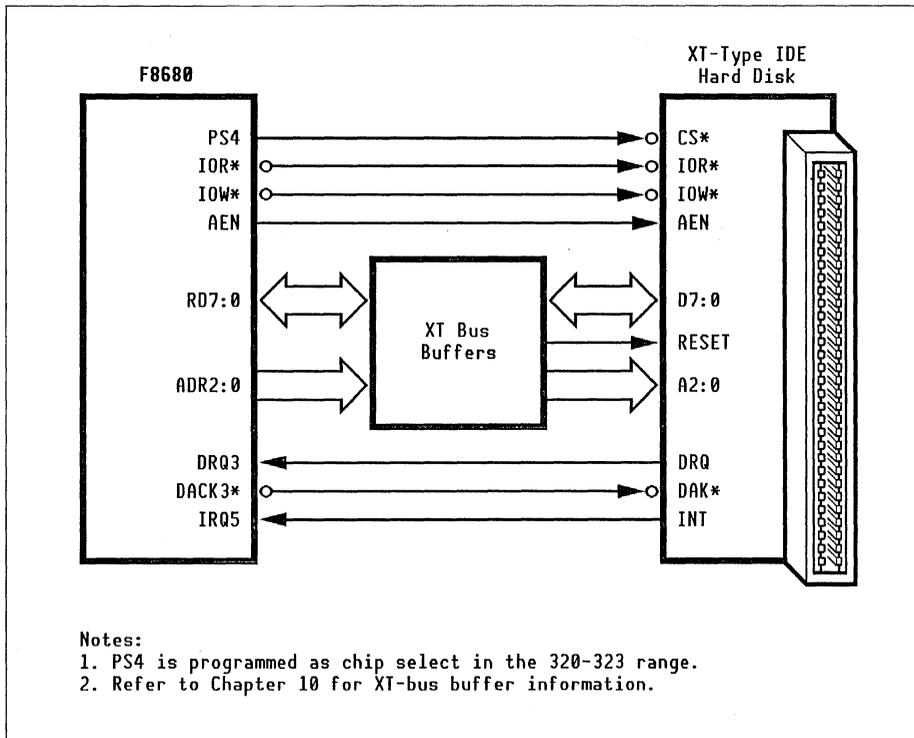
XT-Type Hard Disk Drive Signals

Very few connections are required to interface the typical IDE hard disk drive to the XT bus. Table 8-1 lists the signals commonly used, and Figure 8-1 indicates the required connections. Note that these are all standard XT bus signals except for CS, which you implement by programming one of the PS pins as a programmable chip select in the I/O address range 320-323h.

Table 8-1. *XT-Type IDE Hard Disk Drive Interface Signals*

Signal	Direction Relative to Hard Disk Drive	Description
RESET	I	Reset Strobe
D7:0	I/O	Data Bus
AEN	I	Address Enable
IOW*	I	I/O Write
IOR*	I	I/O Read
DACK3*	I	DMA Acknowledge
DRQ3	O	DMA Request
IRQ5	O	Interrupt Request
A2:0	I	Address Bus
CS*	I	Hard Disk Chip Select

Figure 8-1. Connections to an XT-Type IDE Hard Disk Drive



The programming required to operate PS4 as a chip select is minimal. You may want to use an instruction sequence similar to the following:

```
LFEAT 8C,01100100b ;output driven and active low, program for chip
                    ;select function
LFEAT 8E,20         ;lower part 20 of cs address 320
LFEAT 8F,FF         ;upper part 3 of cs address 320, enable for reads
                    ;and writes, mask lower 4 bits
```

This sequence sets PS4 to act as an active low chip select in the range 320-32F.

XT-Type Hard Disk Drive Registers

The I/O address range of 320-32Fh must be decoded as the chip select signal to the XT hard disk drive. The hard disk controller built into the drive assigns the I/O addresses to programming registers in an XT-compatible manner, as shown in Table 8-2. Refer to the drive manufacturer's data sheets for programming information on these registers.

Table 8-2. *XT-Type Hard Disk Drive Registers*

I/O Address	Type	Register Name
320	R/W	Data Register
321	R	Status Register
321	W	Reset
322	W	Controller Select Register
323	W	DMA, Interrupt Mask Register

AT-Type Hard Disk Drive Signals

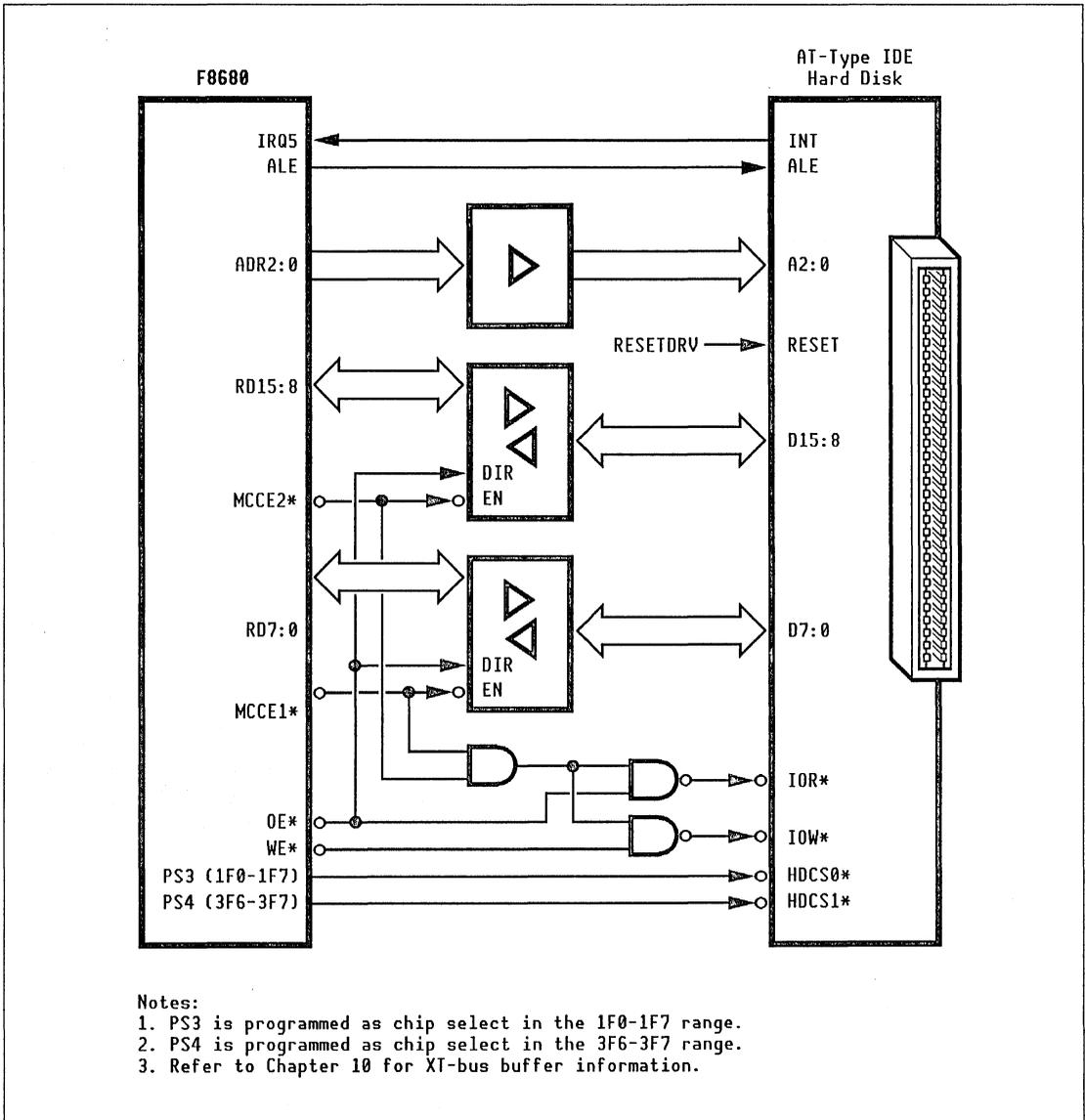
You can use an IDE hard disk drive with an AT-type (16-bit) interface in your system if the drive interface signal timings meet the specifications of the PCMCIA interface (whose timings are provided in Appendix B). In this implementation, the drive would communicate with the system through the memory bus instead of the I/O bus.

You must provide SuperState R code that makes use of the Virtual I/O feature of the F8680 chip to divert I/O instructions to the PCMCIA bank memory locations you have chosen for the drive. Table 8-3 lists the signals associated with the typical AT IDE drive. Figure 8-2 suggests the logic that could be used to implement the interface.

Table 8-3. *AT-Type IDE Hard Disk Drive Interface Signals*

Signal	Direction Relative to Hard Disk Drive	Description
RESET	I	Reset Strobe
D15:0	I/O	Data Bus
ALE	I	Address Enable
IOCS16	O	Hard Disk requires 16-bit transfer
IOW*	I	I/O Write
IOR*	I	I/O Read
IRQ5	O	Interrupt Request
A2:0	I	Address Bus (for I/O address decode)
HDCS0*	I	Primary Chip Select (1F0-1F7)
HDCS1*	I	Secondary Chip Select (3F6-3F7)

Figure 8-2. Connections to an AT-Type IDE Hard Disk Drive



AT-Type Hard Disk Drive Registers

Two I/O address ranges must be decoded as chip select signals to the AT hard disk drive. The hard disk controller built into the drive assigns the I/O addresses to programming registers in an AT-compatible manner, as shown in Table 8-4. Refer to the drive manufacturer's data sheets for programming information on these registers.

Table 8-4. *AT-Type Hard Disk Drive Registers*

I/O Address	Type	Register Name
1F1:1F0	R/W	16-bit Data Register
1F1	R	Error Register
1F1	W	Write Compensation Register
1F2	R/W	Sector Count Register
1F3	R/W	Sector Number Register
1F5:1F4	R/W	Cylinder Number Register
1F6	R/W	Drive/Head Register
1F7	R	Status Register
1F7	W	Command Register
3F6	W	Fixed Disk Register
3F7	R	Digital Input Register

Serial and Parallel Ports

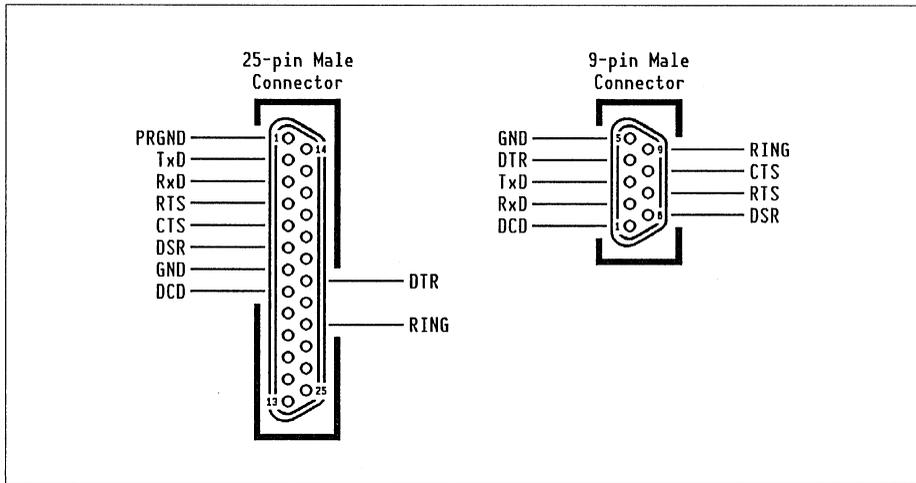
A typical XT-compatible system generally provides a serial communications port and often provides a parallel port as well. The F8680 chip incorporates a serial port, typically used for connection to a modem module. While the F8680 logic does not provide a parallel interface port, you can take advantage of the SuperState R logic to implement the port with a minimal number of components.

RS232-C Signals

The Electronic Industries Association (EIA) defines a signal standard known as RS232-C for use in bidirectional serial communications. The standard defines the set of signal lines necessary for data transmission, reception, control, and status, and defines the voltage level at the signal interface as well.

The signals are shown in Figure 9-1 and described in the list which follows the figure. The descriptions refer to DCE and DTE. Data Communications Equipment (DCE) is the equipment used to transmit and receive the data, and is often a modem. Data Terminal Equipment (DTE) is the equipment used to prepare the data for transmission and interpret received data, and in this case refers to the UART of the F8680 chip.

Figure 9-1. RS232-C Interface Signals



- where:
- PRGND** Protective Ground—usually connected to the shielding of the transmission cable and to the interface connector shell.
 - GND** Signal Ground—common ground for all other signals.
 - TxD** Transmitted Data (O)—data output from the UART to the DCE (modem or interface).
 - RxD** Received Data (I)—data from the DCE (modem or interface) to the UART.
 - RTS** Request To Send (O)—a signal to the DCE (modem or interface) that the UART wants to send data.
 - CTS** Clear To Send (I)—the response of the DCE to the RTS signal from the UART.
 - DTR** Data Terminal Ready (O)—a signal to the DCE (modem or interface) that the UART wants to establish communication.
 - DSR** Data Set Ready (I)—a signal from the DCE that it is ready for data communication.
 - RING** Ring Indicator (I)—a signal from the DCE (usually only from a modem) that the ringing signal has been sensed from an incoming telephone call.

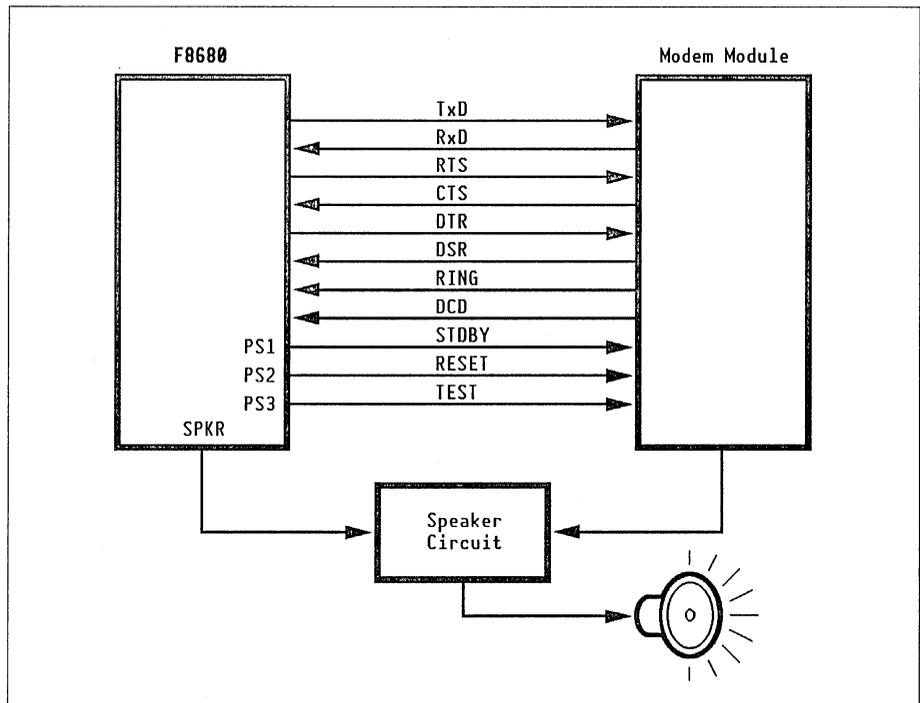
DCD Data Carrier Detected (I)—a signal from the DCE (usually only from a modem) that the data carrier tone from the distant DCE has been detected.

The F8680 chip provides the serial interface signals according to the EIA standards, but at TTL voltage levels instead of the RS232-C interface levels of the specification. TTL voltage levels are more practical for direct connection to a modem module. Line drivers and receivers are required for an external interface (such as the connector shown in Figure 9-1).

Connecting a Modem Module

Because the F8680 serial interface signals operate at TTL voltage levels, interfacing a modem module becomes a simple matter. Figure 9-2 illustrates the connections you need to incorporate a modem module into a system design.

Figure 9-2. *Connecting the Modem Module*



A modem module typically requires additional control lines such as the following:

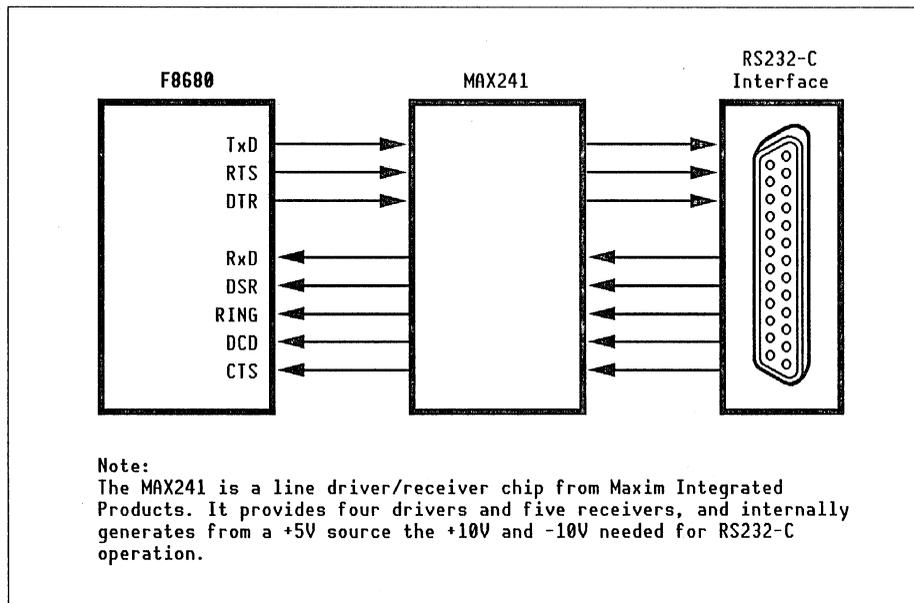
- Modem reset
- Modem power-down/standby
- Modem test.

These signals can be provided through the programmable pins. The modem may also provide an analog output to a speaker circuit. To avoid unnecessary duplication of components, this signal can be combined with the SPKR signal of the F8680 component to form a single input to the speaker.

Connecting Line Drivers

If an interface that operates at RS232-C signal levels is required, you can use drivers and receivers such as those indicated in Figure 9-3.

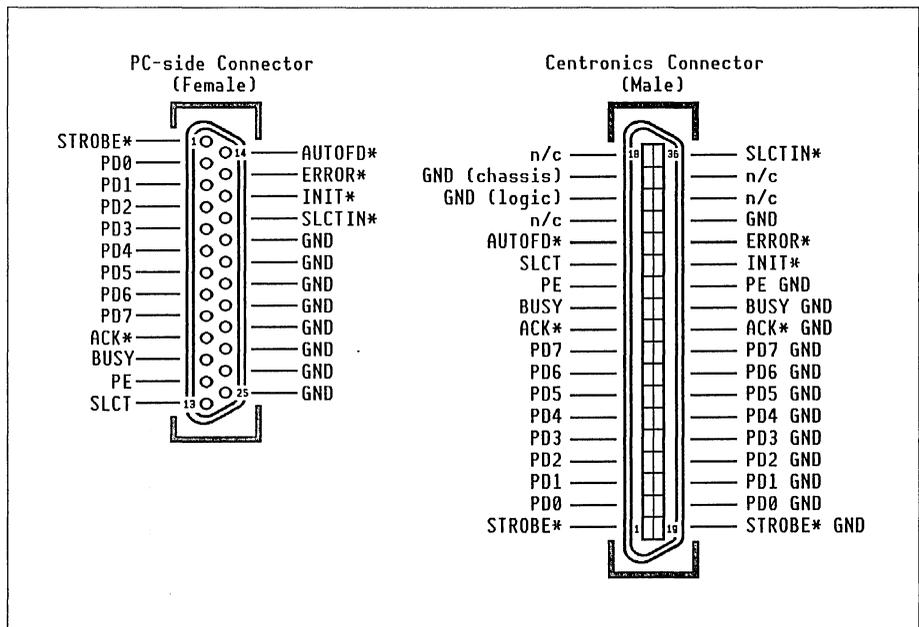
Figure 9-3. *Line Drivers and Receivers for External RS232-C Interface*



Parallel Port Signals

The Centronics® Corporation defines a high-speed parallel output port for connection to its printers. This interface port has become the standard for most printers. The typical Centronics connector and its DB-25 counterpart typically used on PCs are shown in Figure 9-4. The signals are described in the list that follows the figure.

Figure 9-4. Centronics Parallel Port Signals



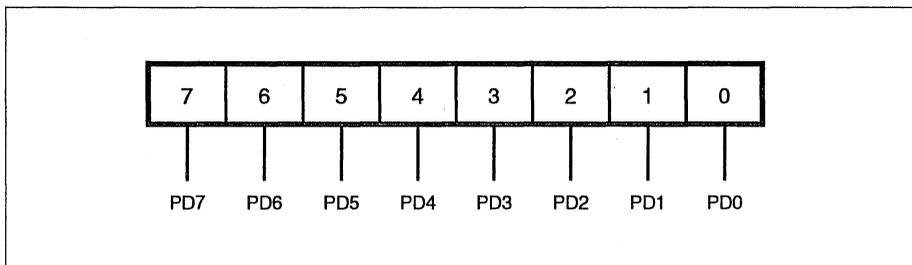
- where: PD7:0 Printer Data (O)—data bus to the printer.
- STROBE* Strobe (O)—pulsed low to indicate that the PD7:0 data is valid.
- ACK* Acknowledge (I)—pulsed low by the printer to indicate that the PD7:0 data has been latched and new data can be sent.
- BUSY Busy (I)—set high by the printer to indicate that no further data can be accepted at the moment.

PE	Paper End (I)—set high by the printer to indicate that there is no more paper to print on.
SLCT	Selected (I)—set high by the printer to indicate that it is “on line.”
AUTOFD*	Auto Feed (O)—set low to indicate to the printer that it should move the paper up each time it prints a line.
ERROR*	Error (I)—set low by the printer to indicate that there is presently an error condition.
INIT*	Initialize (O)—set low to initialize the printer and clear its buffers.
SLCTIN*	Select Input (O)—set low to signal the printer that it is “selected.”

Parallel Interface

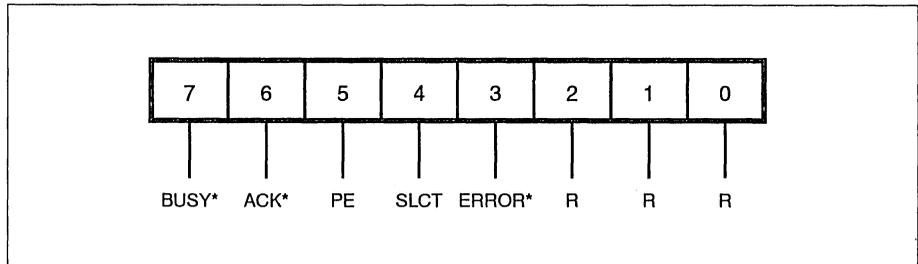
An XT-compatible parallel interface that responds as LPT1 uses the I/O address range 378-37B. I/O address 378 is the Data Register, 379 is the Control Register, and 37A is the Status Register; 37B is not used. The interface interrupts the system on IRQ7. If the parallel interface is to respond as LPT2, the I/O address range 278-27B is used instead. IRQ5 is used if available (if not used for the hard disk). Figures 9-5 through 9-7 show the bits of the parallel port registers.

Figure 9-5. Data Register—I/O Address 378/278 (W/R)



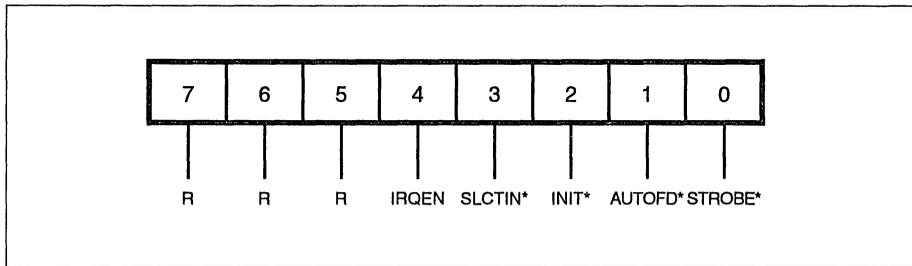
bits: 7:0 PD7:0 Printer Data—data written to this register is sent to the printer. Data returned from reading this register is the same as the data last written.

Figure 9-6. Printer Status Register—I/O Address 379/279 (R)



<i>bits:</i> 7	BUSY*	Busy—a 0 indicates that the printer is busy and cannot accept data.
6	ACK*	Acknowledge—a 0 indicates that the printer has accepted the data sent.
5	PE	Paper End—a 1 indicates that the printer is out of paper.
4	SLCT	Selected—a 1 indicates that the printer is “on-line.”
3	ERROR*	Error—a 0 indicates that an error condition exists in the printer.
2:0	R	Reserved—always mask off for reads.

Figure 9-7. Printer Control Register—I/O Address 37A/27A (W/R)



<i>bits:</i> 7:5	R	Reserved, write as 0 always.
4	IRQEN	IRQ Enable—set to 1 to enable an interrupt to be generated whenever the printer sets ACK* inactive.
3	SLCTIN*	Select Input—set to 1 to select the printer.
2	INIT*	Initialize—set to 0 for 50µs minimum to initialize the printer and clear its input buffer.
1	AUTOFD*	Auto Feed—set to 1 to enable auto line feed function on printer.
0	STROBE*	Strobe—set to 1 to indicate to the printer that the data on PD7:0 is valid.

A parallel interface can be incorporated into system hardware designs in many different ways, as shown in the following sections.

SuperState R Support for Parallel Port

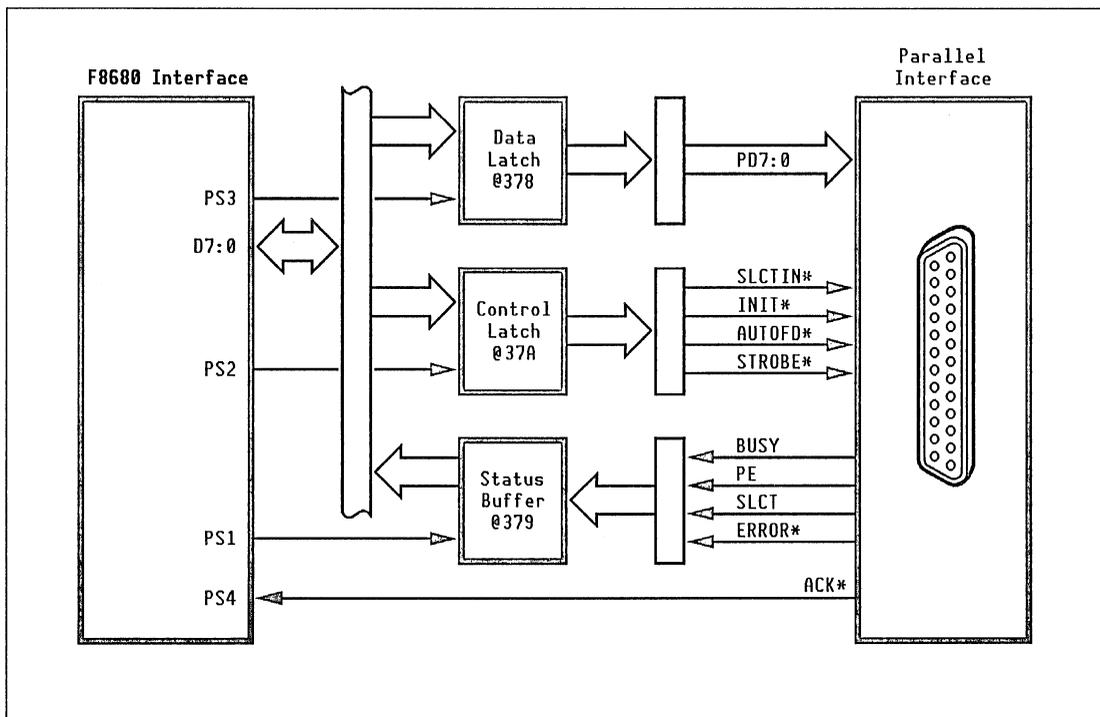
The SuperState R features of the F8680 chip can be used to incorporate a parallel port into system designs with only a few components. To incorporate the interface, you would only have to:

- Provide a buffer to return BUSY, PE, SLCT, and ERROR* status, and use programmable pin PS1 as chip select to decode reads at I/O address 379.
- Provide a control latch to send SLCTIN*, INIT*, AUTOFD*, and STROBE*, and use programmable pin PS2 as chip select to decode writes at I/O address 37A. Shadow these writes and return the value written for reads at this port.

- Provide an output latch for data writes to I/O address 378, using programmable pin PS3 as the chip select. Shadow these writes through Virtual I/O and return the last byte written when a read from I/O address 378 takes place.
- Monitor the ACK* line through programmable pin PS4 programmed for external SuperState R interrupt. The SuperState R interrupt routine can read the IRQEN setting (the control byte is shadowed) to decide whether IRQ7 (or IRQ5) is needed. Remember that ACK* must be reflected in the Printer Status Register.

In summary, you can incorporate a parallel port with only two latches and three buffers. This design is shown in Figure 9-8. However, note that the design requires using all of the programmable pins. For this reason, you may want to use a CHIPS 82C605 Serial/Parallel Interface, which internally decodes the address.

Figure 9-8. *Parallel Port Implementation with Latches*



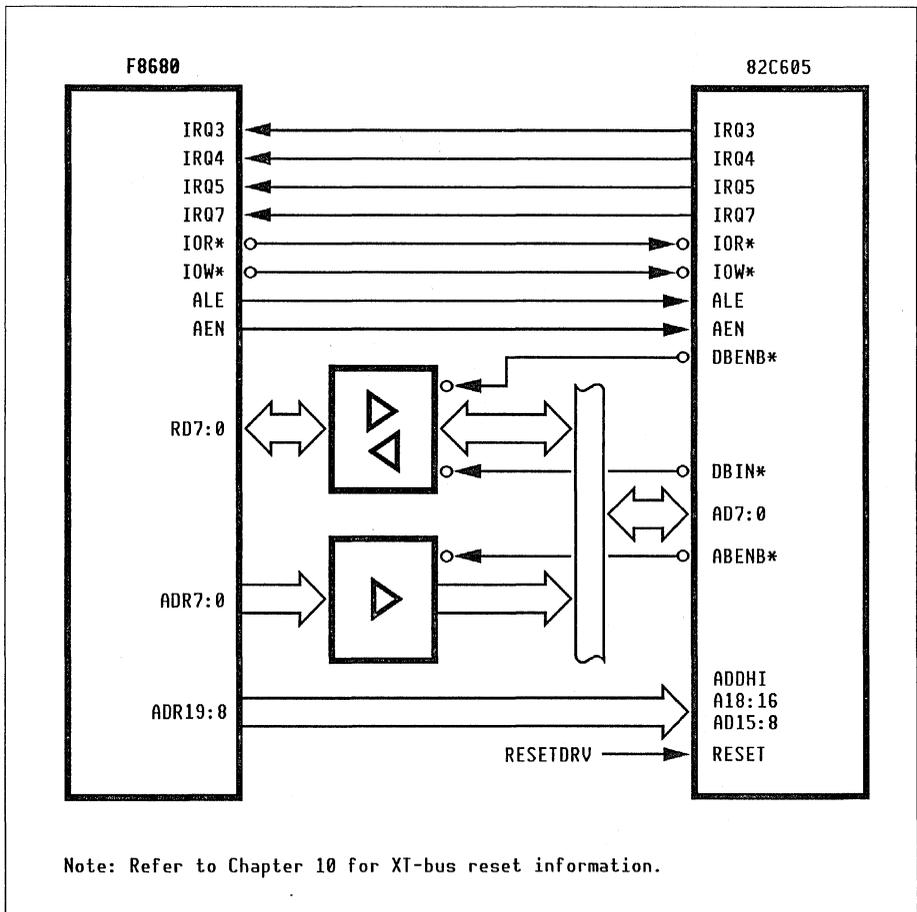
Connecting the CHIPS 82C605 Multifunction Controller

The CHIPS 82C605 Multifunction Controller provides a complete communications subsystem in a single package. It offers:

- Two UARTs
- An enhanced bidirectional parallel port
- Internal I/O address decoding.

Figure 9-9 shows the typical connection of the controller to the F8680 component.

Figure 9-9. Connections to the 82C605 Multifunction Controller



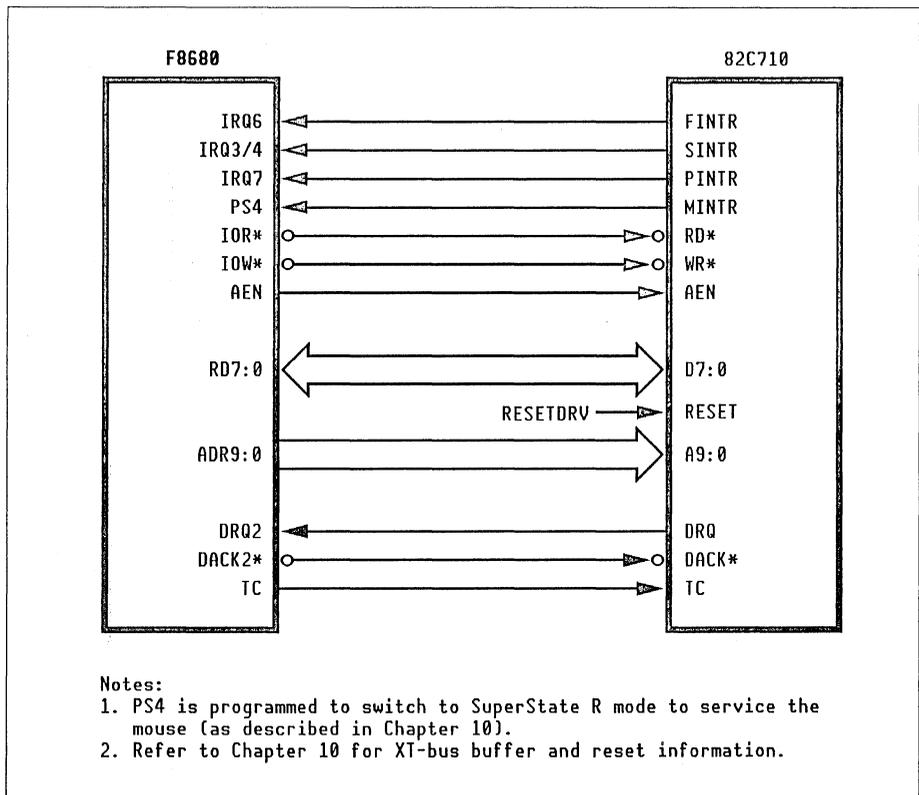
Connecting the CHIPS 82C710 Universal Peripheral Controller

The CHIPS 82C710 Universal Peripheral Controller incorporates the following subsystems in a single CMOS package:

- Enhanced bidirectional parallel port
- 16C450-compatible UART
- PS/2-style mouse port
- NEC 765-compatible floppy disk controller with data separator circuitry
- Interface to an integrated drive electronics (IDE) hard disk drive.

It is a low-power device that draws only 250 μ A when in standby mode, a mode in which all register values are maintained. Figure 9-10 illustrates the connections needed between the 82C710 controller and the F8680 chip.

Figure 9-10. Connections to the 82C710 Universal Peripheral Controller



XT Subsystem

The standard XT bus signals are listed in Table 10-1. The table shows each pin on the XT bus card-edge connector and its corresponding F8680 signal pin.

Table 10-1. *XT Bus Signals*

XT Bus Pin	Signal	Description	Direction Relative to F8680	F8680 Pin
A1	IOCHCK*	I/O Channel Check	I	2
A2	SD7	Data Bus Bit 7	I/O	68
A3	SD6	Data Bus Bit 6	I/O	67
A4	SD5	Data Bus Bit 5	I/O	66
A5	SD4	Data Bus Bit 4	I/O	64
A6	SD3	Data Bus Bit 3	I/O	63
A7	SD2	Data Bus Bit 2	I/O	62
A8	SD1	Data Bus Bit 1	I/O	61
A9	SD0	Data Bus Bit 0	I/O	60
A10	IOCHRDY	I/O Channel Ready	I	6
A11	AEN	Address Enable	O	7
A12	SA19	Address Bus Bit 19	O	9
A13	SA18	Address Bus Bit 18	O	11
A14	SA17	Address Bus Bit 17	O	14
A15	SA16	Address Bus Bit 16	O	16
A16	SA15	Address Bus Bit 15	O	18
A17	SA14	Address Bus Bit 14	O	21
A18	SA13	Address Bus Bit 13	O	23
A19	SA12	Address Bus Bit 12	O	25
A20	SA11	Address Bus Bit 11	O	27
A21	SA10	Address Bus Bit 10	O	29

Table 10-1. XT Bus Signals (continued)

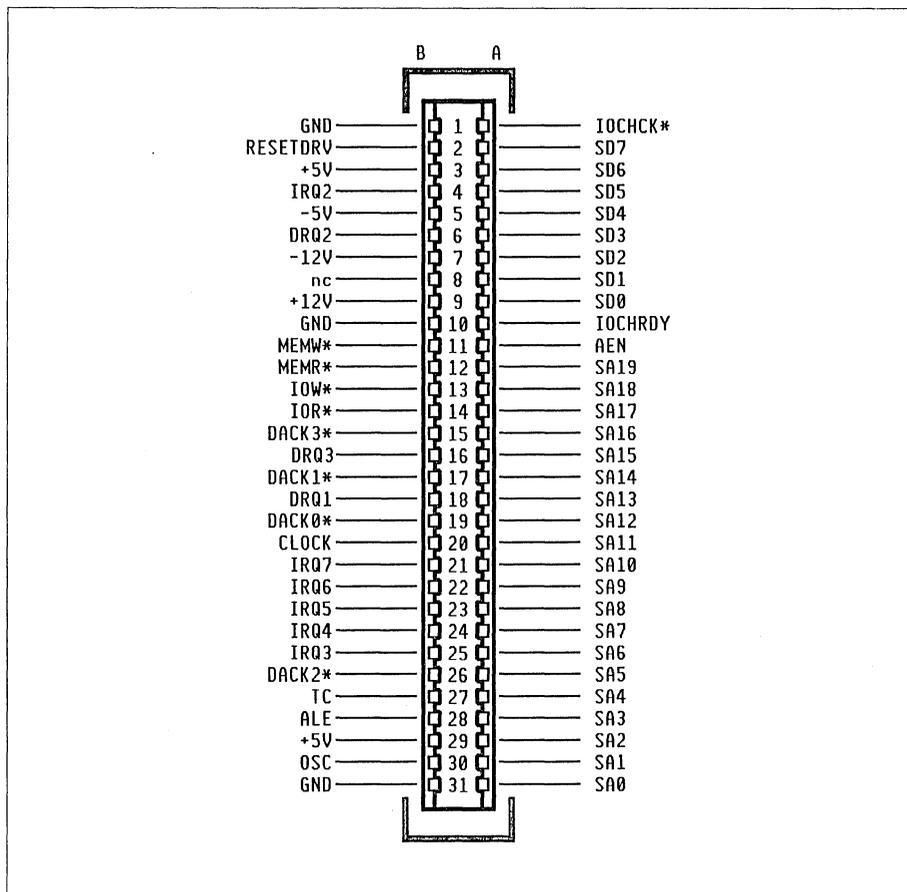
XT Bus Pin	Signal	Description	Direction Relative to F8680	F8680 Pin
A22	SA9	Address Bus Bit 9	O	32
A23	SA8	Address Bus Bit 8	O	34
A24	SA7	Address Bus Bit 7	O	36
A25	SA6	Address Bus Bit 6	O	38
A26	SA5	Address Bus Bit 5	O	40
A27	SA4	Address Bus Bit 4	O	43
A28	SA3	Address Bus Bit 3	O	45
A29	SA2	Address Bus Bit 2	O	47
A30	SA1	Address Bus Bit 1	O	50
A31	SA0	Address Bus Bit 0	O	52
B2	RESETDRV ¹	Bus Reset	—	—
B4	IRQ2	Interrupt Request 2	I	3
B6	DRQ2	DMA Request 2	I	4
B11	MEMW*	Memory Write	O	8
B12	MEMR*	Memory Read	O	10
B13	IOW*	I/O Write	O	12
B14	IOR*	I/O Read	O	15
B15	DACK3*	DMA Acknowledge 3	O	17
B16	DRQ3	DMA Request 3	I	20
B17	DACK1*	DMA Acknowledge 1	O	22
B18	DRQ1	DMA Request 1	I	24
B19	DACK0*	DMA Ack 0 (Refresh)	O	26
B20	CLOCK	XT Bus Clock	O	28
B21	IRQ7	Interrupt Req 7	I	31
B22	IRQ6	Interrupt Req 6	I	33
B23	IRQ5	Interrupt Req 5	I	35
B24	IRQ4	Interrupt Req 4	I	37
B25	IRQ3	Interrupt Req 3	I	39
B26	DACK2*	DMA Acknowledge 2	O	42
B27	TC	Terminal Count	O	44
B28	ALE	Address Latch Enable	O	46
B30	OSC ²	Timer Clock	—	—

¹ The F8680 chip provides no RESETDRV signal. See "Generating RESETDRV for the XT Bus" in this chapter.

² The OSC clock requirements are the same as those for the CLK14 input to the F8680 chip.

Figure 10-1 shows the standard XT-type card-edge connector.

Figure 10-1. XT Bus Card-Edge Connector



Generating RESETDRV for the XT Bus

Since the F8680 chip provides no RESETDRV signal, you should use the RESET input to the F8680 chip for RESETDRV as well as reset. Be sure to design your reset circuit so that it meets the minimum reset time required for both the system and any XT bus expansions.

Connecting the Interrupt Request (IRQ) Pins

The XT architecture defines the assignment of the interrupt request lines according to Table 10-2.

Table 10-2. *XT-Compatible Interrupt Assignments*

Channel	Function
IRQ0	Timer Channel 0
IRQ1	Keyboard interface
IRQ2	Video
IRQ3	Serial port COM2
IRQ4	Serial port COM1
IRQ5	Parallel port LPT2 / XT Hard Disk
IRQ6	Floppy drive controller
IRQ7	Parallel port LPT1

Emulating Extra IRQ Lines Through the Programmable Pins

In an AT-compatible system, a second interrupt controller is connected in cascade with the first, providing seven new interrupt lines. The eighth line replaces the IRQ line lost in cascading the new controller. Table 10-3 lists the assignments of the new interrupt lines.

Table 10-3. *AT-Compatible Assignment for Second Interrupt Controller*

Channel	Function
IRQ8	Real-Time Clock
IRQ9	Reserved ¹
IRQ10	Available for AT bus cards
IRQ11	Available for AT bus cards
IRQ12	Mouse
IRQ13	Numeric coprocessor
IRQ14	AT-compatible hard disk
IRQ15	Available for AT bus cards

¹ IRQ9 replaces IRQ2, which is used to cascade the second interrupt controller in an AT-compatible system.

You can use the F8680 programmable pins feature to simulate the presence of some of these interrupt lines in your system. For example, suppose you want to accommodate a PS/2-compatible mouse, which is assigned to IRQ12. You could approach the design using the CHIPS 82C710 Universal Peripheral Controller. This device provides a PS/2-compatible mouse interface that generates an interrupt on its MINTR line. See Chapter 9 for full connection details.

You would first choose a programmable pin to use; PS4 will be used for this example. You need only connect MINTR to PS4 to complete the hardware phase of the design. Program PS4 so it will generate a switch to SuperState R mode when PS4 goes active. The polarities of MINTR and PS4 are both programmable, so you must be sure they are both set the same.

Your SuperState R routine that handles external switch requests can then call INT 14h, the routine corresponding to IRQ12, to handle the mouse interrupt through a standard driver.

The *F8680 PC/CHIP Programmer's Reference Manual* provides the programming details you would need for this design.

Connecting the DREQ/DACK Pins

The XT architecture defines the assignment of the DMA request/acknowledge lines as listed in Table 10-4. The DRQ input lines are active when high; the DACK output lines are active when low.

Table 10-4. *XT-compatible DMA Request Line Assignment*

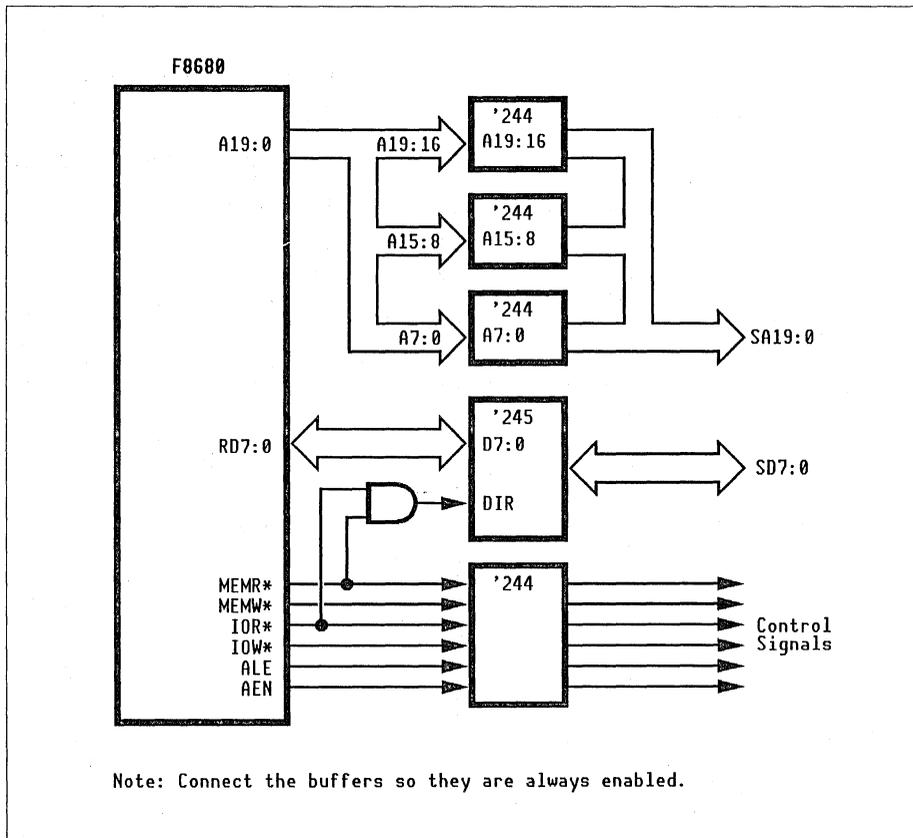
DMA Lines	Assignment
DRQ0 ¹ /DACK0*	Refresh
DRQ1/DACK1*	Available for XT bus expansions
DRQ2/DACK2*	Floppy disk drive
DRQ3/DACK3*	XT-compatible hard disk drive

¹ The DRQ0 line is not available external to the F8680 chip.

Driving Peripheral Devices on the XT Bus

The drive capability of the F8680 outputs is adequate for direct connection of a few peripheral devices on the XT bus, depending on their requirements. However, if you intend to provide adequate drive for unknown adapter cards, you should isolate the XT bus from the system buses with buffers. Figure 10-2 illustrates the type of buffering you should use.

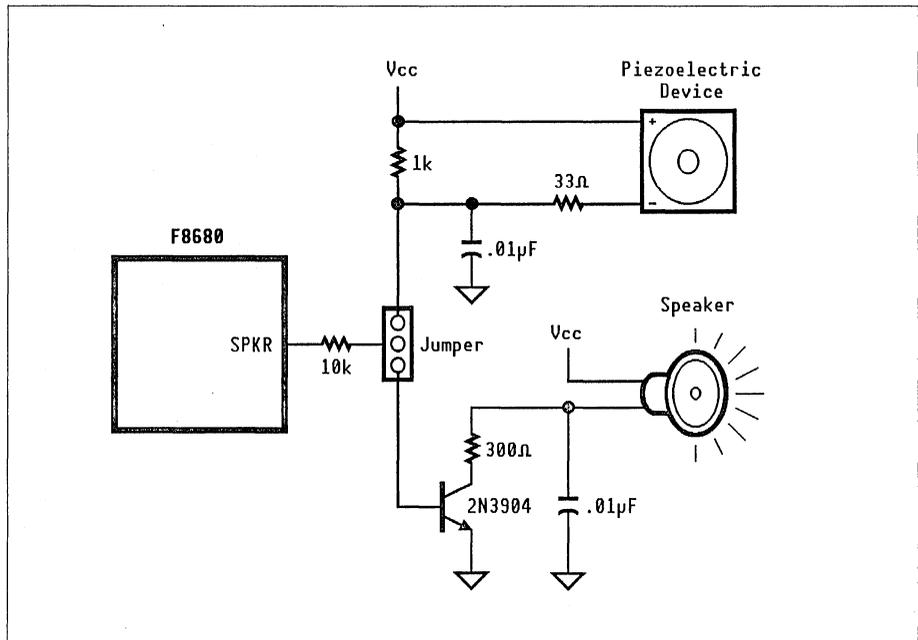
Figure 10-2. *Buffers for the XT Bus*



Connecting a Speaker

The SPKR output of the F8680 chip provides a digital signal at TTL voltage levels. This is adequate for driving certain sounding devices such as a piezoelectric device, but will not work if you want to drive a speaker. Figure 10-4 illustrates the circuits necessary for driving a piezoelectric device or a speaker.

Figure 10-3. *Driver Circuits for Sounding Devices*



Special Applications

The F8680 chip is ideal for non-traditional applications such as pen-based PCs, supervisory control and data acquisition (SCADA) systems, and embedded control systems. In the past you might have developed these applications using specialized processors and proprietary software development kits. Now you can use the F8680 component along with off-the-shelf software development tools to design powerful, low-cost systems that can be programmed as easily as a PC.

Incorporating a Display Overlay

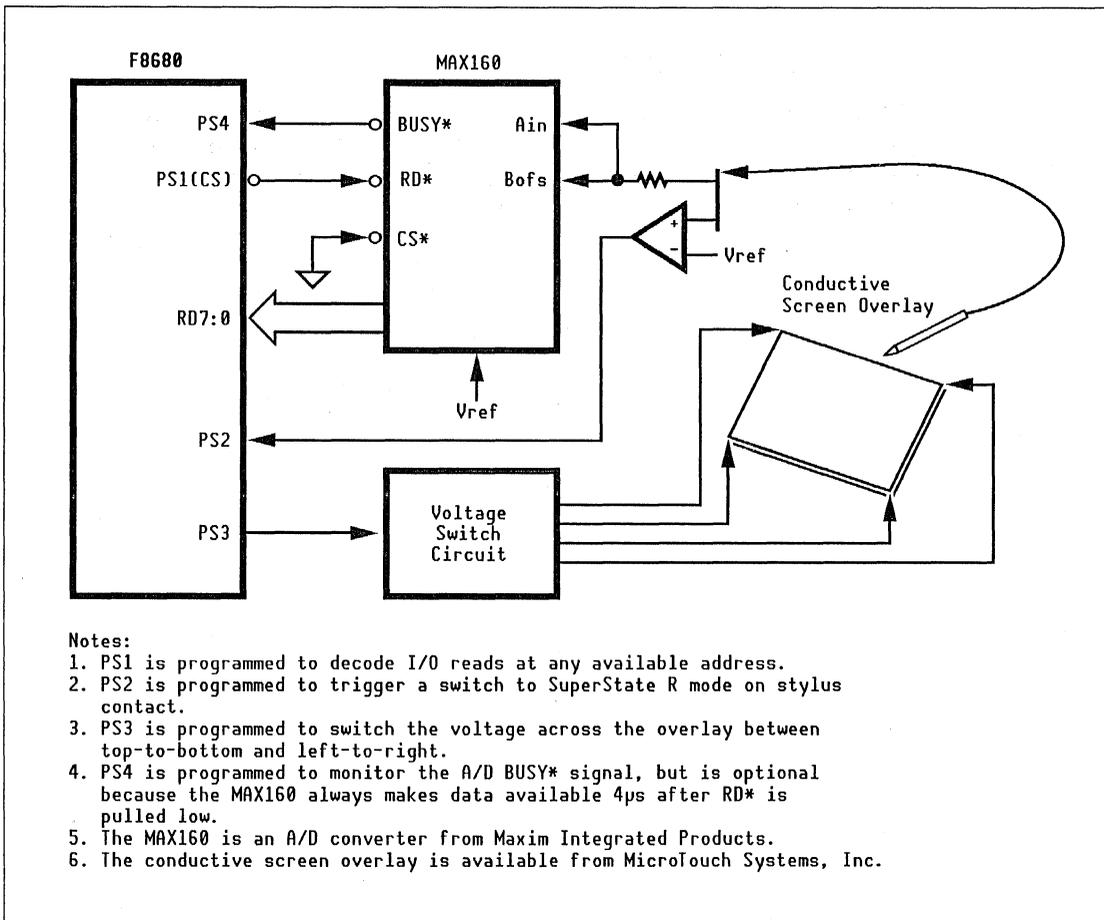
The display overlay is a relatively new input device for the PC. It is generally a transparent panel with an electrically conductive coating that is placed on top of the LCD panel. One edge of the overlay is connected to ground and the other to a supply voltage, resulting in a voltage that varies linearly from edge to edge. If you touch the conductive coating with a metal stylus, the voltage measured at the stylus corresponds to the distance the stylus is from the edge.

If you connect the left edge of the overlay to ground and the right edge to supply voltage, you can measure a horizontal position voltage x . Connecting the bottom edge to ground and the top edge to supply voltage allows you to measure a vertical position voltage y . You need only an analog-to-digital (A/D) converter to convert these voltages to digital values for input to the CPU. The CPU can then interpret these values as it would for a mouse or other positioning device.

Connecting the Overlay

A practical display overlay design requires that multiple samples be taken and averaged at each point where the stylus touches the overlay. The overlay for this design is available from MicroTouch Systems which supplies a conductive overlay and the code needed for sampling. The component layout is shown in Figure 11-1.

Figure 11-1. Connecting an Overlay



Monitoring for Stylus Contact

The F8680 external SuperState R interrupt feature is fundamental to overlay management. Every time the user puts the stylus down on the conductive overlay, the comparator generates a “stylus contact” interrupt. This interrupt causes a switch to SuperState R mode, which then reads the x-y coordinates from the A/D converter.

Typically, the F8680 chip will request 15-25 coordinate pairs per second, allowing the CPU enough time to process each set of coordinates and act appropriately (often by drawing pixels on the LCD panel to match the stylus movement).

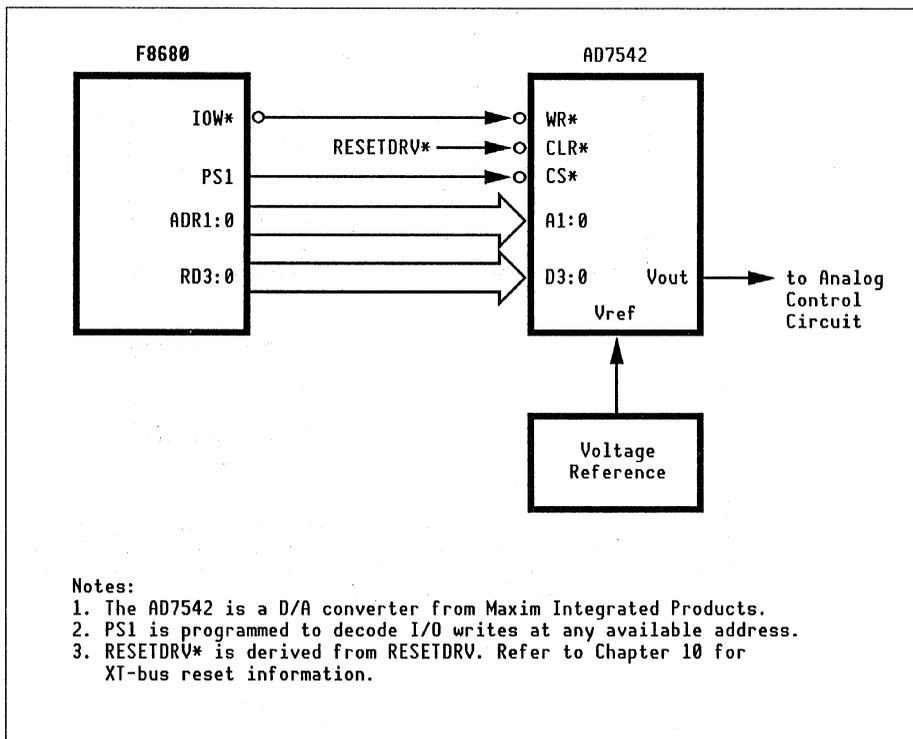
Monitoring a Control Panel

The pen-based PC will generally provide several function buttons to activate special operations. These buttons can be connected to spare PS pins that are programmed as external SuperState R inputs. In this way, pressing a button can cause a switch to SuperState R mode, which will service the event appropriately.

Connecting a Digital-to-Analog Converter

You must incorporate a digital-to-analog (D/A) converter in your design when you need to interface with an analog control circuit. The design of the analog circuit itself dictates what type of driver circuit you will have to provide. However, the interfacing of the D/A converter itself is predictable and is typically done as shown in Figure 11-2.

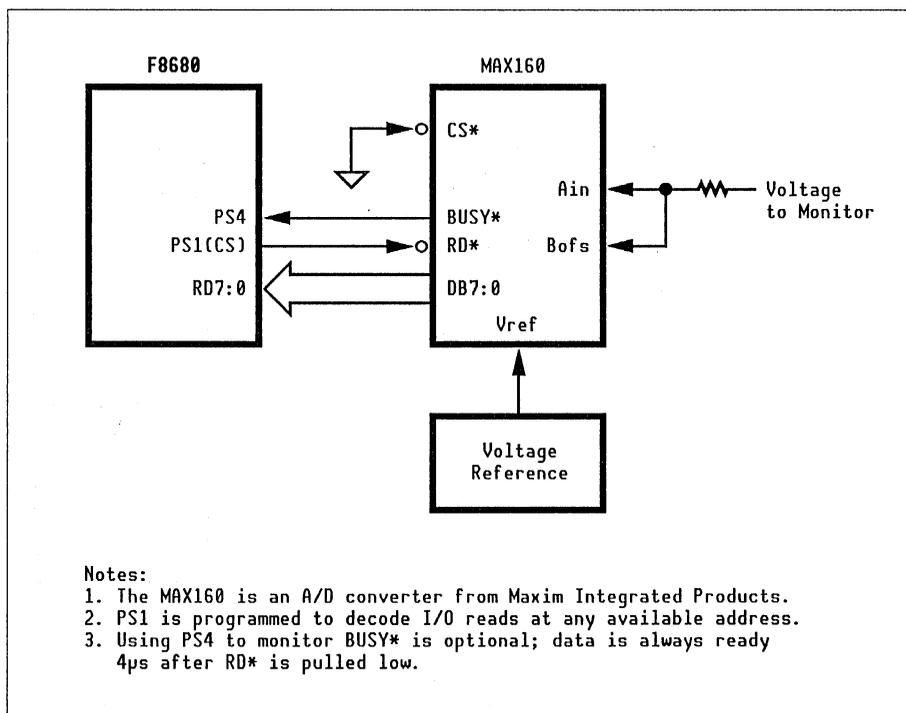
Figure 11-2. *Incorporating the D/A Converter*



Connecting an Analog-to-Digital Converter

You can use an analog-to-digital (A/D) converter circuit for a variety of monitoring functions, such as the battery voltage monitor circuit shown in Chapter 5. Connecting an A/D converter is typically done as shown in Figure 11-3.

Figure 11-3. Incorporating the A/D Converter



Emulating the Real-Time Clock

You can use the continuous time-of-day clock of the F8680 chip to simulate the presence of the Motorola 6818-type Real-Time Clock (RTC) with CMOS RAM. In fact, the CHIPS SuperState R code that provides the basic DMA handling for the F8680 chip also provides basic RTC/CMOS RAM emulation at I/O ports 070-071, as in AT-type systems. An applications note in Chapter 10 of the *F8680 PC/CHIP Programmer's Reference Manual* provides information on expanding the RTC emulation.

Processor Architecture

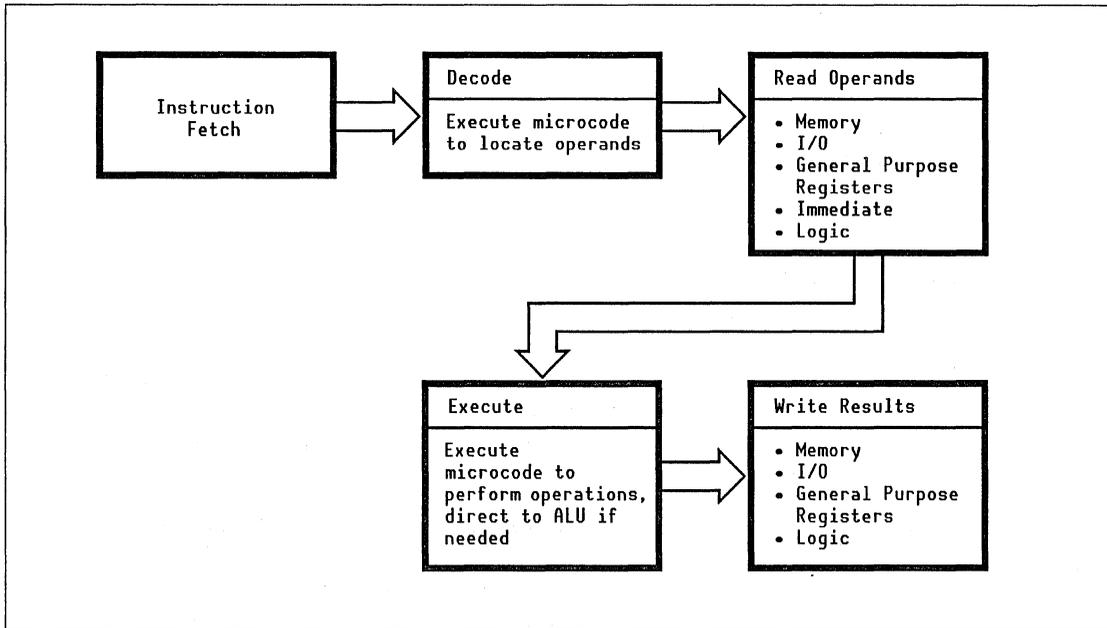
Chips and Technologies, Inc. has engineered the central processor of the F8680 single-chip PC to provide compatibility with the 8086 microprocessor yet perform like an 80286 microprocessor. The SuperState R extensions to the 8086 architecture are extensions in concept only, as the F8680 architecture includes these extensions as an integral part of the chip design. The CMOS processor provides operation at clock rates from DC to full speed.

Processor Pipeline

The F8680 CPU uses a pipeline to streamline the flow of instruction execution. Once the pipeline is loaded, the execution of one instruction overlaps with the fetching and decoding of other instructions and instruction operands. This process significantly reduces the overall time required to execute each instruction.

The instructions that pass through the pipeline are processed in four stages: decode, operand read, instruction execution, and operand write. The relationship among these stages is illustrated in Figure A-1.

Figure A-1. Processor Pipeline



The pipeline stages operate as follows:

- The CPU fetches the instruction opcode pointed to by CS:IP any time the system buses are free and the instruction fetch register is empty.
- Once an instruction opcode is available, it is decoded to determine whether operands are needed. If a memory operand is required, it is fetched.
- Once the instruction opcode and memory operand are available, the processor branches to the appropriate routine in microcode and executes the instruction.
- If required, the result of the operation is written to memory.

These operations overlap from one instruction to the next. For example, while one instruction is being decoded, the next is being fetched. Therefore, the instruction execution rate cannot be determined simply by counting CPU clock cycles. The actual instruction throughput depends on the type and sequence of instructions and how well their execution can be made to overlap.

For this reason, the *F8680 PC/CHIP Programmer's Reference* describes instruction processing in terms of *flows*, that is, the number of cycles from the start of the pipeline to the end. Actual instruction processing time will almost always be considerably less than the sum of all discrete operations taking place.

A configuration bit is provided in CREG 0B to disable overlap in instruction execution. This bit is generally set only for diagnostic purposes. However, it can also effect a performance reduction similar to that offered by CREG 0A, as described below.

Performance Reduction

A configuration register is provided at CREG 0A for performance reduction. This feature is available as a power-saving measure that is most effective when used with DRAM. The feature inserts the programmed number of idle cycles between execution of each instruction. The delay is introduced only between instructions, not between iterations of the same instruction (such as one with a REPEAT prefix).

The performance reduction that results depends heavily on the instruction mix being executed. For example, if a series of string move instructions is being executed, the programmed performance reduction has little effect. In this case, a more effective means of decreasing DRAM accesses might be to disable the processing pipeline overlap feature (CREG 0B).

Index

8086, 1-1, 1-3, 1-6, 1-11, 2-12
8254, 1-9
8259, 1-9
82C450, 2-8, 4-10
82C455, 2-6, 4-8
82C605, 9-9-9-10
82C710, 2-8, 6-7, 7-1, 7-3, 9-11, 10-6
82C765, 2-6, 7-1, 7-3

A

ACDCLK, 4-4, 5-7
ACK, 9-5, 9-7-9-9
Acknowledge, 7-3, 8-1, 9-5, 9-7, 10-6
Addressing
 I/O, 1-5, 1-11, 5-7, 6-6, 7-1-7-3, 8-1, 8-3-8-4,
 8-6, 9-6, 9-8-9-10
 memory, 2-8, 3-5, 3-7, 3-13, 3-17
ALE, 8-4
Application Programs, 1-1, 1-3-1-5, 1-8-1-10,
 1-12, 2-1, 2-10, 3-5, 3-16
Architecture, 1-1, 1-4, 1-10, 10-4, 10-6
AUTOFD, 9-6, 9-8

B

Background, 1-9
Backlighting, 2-4, 4-6, 5-2
Bank switching, 1-8, 3-2-3-3, 3-22
Batteries, 2-2, 2-4, 2-6, 3-20, 5-2, 5-9-5-10, 11-4
BIOS, 1-4-1-5, 2-2, 3-16-3-17

Boundary, 3-7, 3-13, 3-17, 3-20
BSY, 3-21
Buffering, 3-21-3-22, 3-24, 6-4, 9-8, 10-6
Buses, 1-4, 1-6, 1-8, 1-10, 1-13, 2-12, 3-1, 3-5,
 3-7-3-8, 3-14, 3-16-3-17, 3-21-3-22,
 3-24, 4-1-4-2, 6-4, 6-7, 7-1-7-3, 8-1, 8-4,
 9-5, 10-1, 10-3, 10-6, 11-3

C

Carrier, 9-3
CAS, 3-9
Cascading, 10-5
Cells, 4-4
Centronics, 9-5
CGA, 1-8-1-9, 4-5, 4-8
CHS, 4-5
CLKCANRUN, 5-7
Configuration
 memory, 3-1
Controller
 CRT, 4-4-4-5
 Graphics, 1-9, 3-6, 4-8, 5-7
 Interrupt, 1-9, 10-5
 keyboard, 6-6
 Memory, 1-8, 3-1, 3-5, 3-9, 3-14, 3-16, 3-26
CPU, 1-5-1-6, 1-8, 1-11, 3-2-3-3, 3-5, 11-1,
 11-3
CPUCLK, 3-7
CREG, 3-3, 3-5, 3-7, 3-9, 3-14, 3-20, 3-26,
 4-2-4-5, 4-8, 5-2-5-3
CRT, 1-9, 4-1, 4-7, 4-10

D

Decoding, 1-11, 2-4, 2-6, 3-1-3-3, 3-5, 3-8, 5-7,
6-4, 7-2-7-3, 8-3-8-4, 8-6, 9-8-9-10, 11-3
Desktop, 2-8
Digital-to-analog (D/A) converter, 11-3
Digitizing, 2-10
Display, 4-2, 4-7
DMA
 See Controller, DMA
DOTCLOCK, 4-2, 4-5
Drivers, 3-18, 6-7, 9-3, 10-7
Duty, 4-4, 5-7

E

EGOUT, 4-5
Embedded controller, 2-12, 11-1
EMS, 1-8, 2-2, 2-4, 2-8
Emulation, 6-6
Execute-In-Place, 2-2
EXTSS, 5-7

F

FDC, 7-1-7-3
FET, 4-6
Fetching, 1-6, 1-11
Flags, 1-11
Flat-pack, 4-8, 4-10
Floppy, 2-6, 2-8, 7-1, 7-3, 9-11, 10-4
Foreground, 1-9
Frequencies, 1-11, 2-6, 4-4, 5-7

G

GENB, 4-5, 4-8
GRACS, 4-1
GRAOE, 4-1
Graphics
 See Controller, graphics
GRAWE, 4-2

H

Handler
 interrupt, 1-5
Hard disk, 1-5, 2-4, 2-6, 2-8, 3-27, 7-1, 8-1,
8-3-8-4, 8-6, 9-6, 9-11
High-resolution, 2-8, 4-1, 4-8, 4-10
HS, 4-2, 4-5, 4-7

I

I/O operations, 1-3, 1-5
IBM, 1-9, 6-6
IFACTIVE, 5-7
INIT, 9-6, 9-8
Initialization, 1-4, 6-1, 9-6, 9-8
INT, 1-5, 7-2, 10-5
Intercepting, 1-3-1-5, 6-6
Interfacing, 1-8, 1-11, 2-6, 2-8, 2-10, 3-6, 3-9,
3-13-3-14, 3-16, 3-18, 3-20-3-21,
3-26-3-27, 4-1, 6-1, 6-3, 6-6-6-7, 7-1,
8-1, 8-4, 9-1-9-3, 9-5, 9-8, 11-3
Interference, 1-13
Interrupts
 See Controller, interrupt
IOCHCK, 1-10
IOR, 7-2, 8-1, 8-4
IOW, 7-2, 8-1, 8-4
IRQ, 1-5, 2-10, 9-8, 10-4
IRQEN, 9-8-9-9

K

Keyboard, 2-2, 2-6, 5-8, 6-1-6-5, 10-4

L

Laptop, 2-4
LCD panels, 1-9, 1-11, 4-2, 4-4
LFEAT, 4-3, 4-6, 7-3, 8-2
LP, 4-2, 4-4-4-5

M

MAD24, 5-7
 Management
 memory, 1-8
 power, 2-2, 2-4, 5-1
 system, 1-3
 MaskROM, 3-26
 Maxim, 11-3
 MCCE1, 3-21
 MCCE2, 3-21
 MCRDY, 3-21
 Memory
 conventional, 2-2, 2-4
 system, 1-8, 2-6, 3-8, 3-14, 3-17, 3-22
 Memory cards, 1-4, 1-8, 2-2, 3-18, 3-20-3-24,
 3-26-3-27, 5-7
 Memory controller
 See Controller, memory
 Memory mapping, 3-1-3-3
 MEMR, 3-17
 MEMW, 3-17
 Microcode, 1-6, 1-9
 MicroTouch, 11-2
 MINTR, 10-5
 MIPS, 1-11, 5-7
 Mode
 graphics, 1-9
 page, 3-6, 3-9
 text, 1-9, 4-8
 Modem, 9-1-9-3
 Monitoring, 1-3-1-5, 1-10-1-11, 2-2, 2-4, 2-10,
 2-12, 3-21, 5-1-5-2, 5-6, 5-9, 6-2, 11-4
 Motorola, 11-5
 Mouse, 6-7, 7-1, 9-11, 10-5, 11-1
 MS-DOS, 1-8, 2-10

N

N-key rollover, 6-5
 NEC, 7-1, 9-11
 NMI, 1-10

O

Oscillator, 5-2, 5-5
 OSCPW, 5-2-5-3, 5-5
 OTPROM, 3-26
 Overhead, 6-7
 Overlap, 3-5
 Overlay, 2-10, 11-1-11-3

P

Package, 1-6, 4-8, 4-10
 Page frames, 1-8
 Palmtop, 2-2
 Parallel, 2-8, 7-1, 9-5-9-6, 9-8, 9-10-9-11, 10-4
 PCMCIA, 1-4, 1-8, 1-11, 2-2, 3-5-3-6, 3-16,
 3-18-3-24, 3-26-3-27, 8-4
 PCS, 5-7
 PE, 9-6-9-8
 Pen-based, 2-10, 11-1, 11-3
 Performance, 1-6, 2-2, 2-4, 2-6, 2-8, 2-12, 3-1,
 3-5, 3-13, 3-16, 3-26, 5-11
 Periodic, 1-4, 1-10, 3-9, 3-13, 5-7-5-8, 5-10, 6-4
 Peripheral devices, 1-5, 5-6, 5-8, 10-6
 PGM, 3-21
 Piezoelectric, 10-7
 Pixels, 1-9, 4-2, 4-4-4-5, 11-3
 POFF, 5-2
 Polling, 1-11
 Portable, 2-6, 6-1
 Ports, 1-5, 2-8, 2-12, 3-27, 6-6-6-7, 9-5, 9-8,
 11-6
 Power-down, 1-4, 4-8, 5-1, 5-3-5-4, 9-4
 Power-up, 1-4, 3-3, 5-1-5-3, 5-7, 6-1
 Printer, 9-5-9-9
 Programmable chip select, 7-1, 8-1
 Programmable pins, 1-10-1-11, 2-2, 2-4, 2-6,
 2-10, 3-23, 3-27, 4-2, 4-6, 4-8, 5-2,
 5-6-5-9, 8-1, 9-4, 9-8-9-9, 10-5
 PROM, 3-16-3-17, 3-27
 Pseudo-SRAM (PSRAM), 1-8, 3-5, 3-14
 PWRUP, 1-10, 5-1-5-4, 6-4

R

RAM, 1-8, 2-2, 2-4, 2-8, 3-2-3-3, 3-5-3-8, 3-14, 3-16-3-18, 3-22-3-23, 3-26, 4-10, 11-6
 RAS, 3-9
 RDY, 3-21
 Receivers, 6-7, 9-3
 REFRESH, 3-14, 3-21
 REG, 3-21-3-24
 Registers
 configuration, 1-5, 5-3
 segment, 1-6
 Relax/Retire, 2-6, 4-8
 Reset, 1-4, 4-8, 7-2, 8-1, 8-3-8-4, 9-4, 10-3, 11-4
 RESETDRV, 10-1-10-3, 11-3
 RFSH, 3-21
 RING, 9-2
 ROM, 1-8, 2-2, 2-4, 3-16-3-18
 ROMCS, 3-5, 3-17
 RTC, 11-6
 RTS, 9-2
 RxD, 9-2

S

SCADA, 2-12, 11-1
 Scan, 2-2, 6-1-6-5
 Scan codes, 2-6, 6-1-6-2
 Schmitt-triggered, 1-13
 SDATA, 1-10, 3-20, 3-26
 Segments, 1-8, 3-6
 Serial, 2-8, 2-12, 6-1, 6-7, 9-1, 10-4
 Shadowing, 2-2, 2-4, 3-16-3-17, 9-9
 SLCT, 9-6-9-8
 SLCTIN, 9-6, 9-8
 slew-rate, 1-13
 SMARTMAP, 4-8
 Software, 1-4, 1-9, 2-12, 3-1, 3-16, 3-18, 3-26, 4-5, 4-8, 5-1-5-4, 5-6, 5-11, 6-2, 6-5-6-6, 11-1
 Speaker, 9-4, 10-7
 Speeds, 1-11-1-12, 3-26, 9-5
 Spin-down, 2-4
 SPKR, 9-4, 10-7

SRAM, 1-8, 2-2, 3-5-3-6, 3-13-3-14, 3-16, 3-26, 4-1-4-2
 State machine, 5-1-5-3, 5-5, 5-7
 Status, 1-10-1-11, 3-19-3-20, 3-26, 5-6, 6-6, 7-3, 8-3, 8-6, 9-1, 9-6, 9-8-9-9
 Strobe, 8-1, 8-4, 9-5, 9-8
 Stylus, 2-10, 11-1-11-3
 SuperState R code, 1-3, 1-5, 1-10, 2-10, 2-12, 5-10, 6-1, 6-4, 6-6, 8-4, 11-3
 SuperState R instruction set, 1-6
 Suspend mode, 1-11, 3-14, 5-1-5-3
 Switches, 5-3-5-4, 5-7, 6-2
 Synchronization, 4-7
 System
 operating, 1-10

T

TC, 7-3
 Tics, 5-8, 6-4
 Time-of-day, 5-1-5-2, 11-6
 Timers, 1-9, 3-9, 5-8, 6-4
 Traps, 1-5, 6-6
 Triggering, 5-7
 TSR, 1-5
 TTL, 9-3, 10-7
 TxD, 9-2

U

UART, 1-9, 5-2, 5-5, 9-1-9-2, 9-11

V

Vcc, 1-13, 5-1, 5-4
 VCCPAD, 1-13
 VGA, 2-6, 2-8, 4-10
 Virtual I/O, 1-4-1-5, 6-6, 8-4, 9-9
 Virtual Interrupts, 1-4
 Visual Map, 1-9
 Vpp, 3-19, 3-27

W

Waveform, 1-10

WID, 3-8

WP, 3-21

WR, 7-2

Write-protection, 3-21

X

XIP

See Execute-in-place

Regional Sales Offices

United States

California, Irvine
Chips and Technologies, Inc.
Phone: 714-852-8721

California, San Jose
Chips and Technologies, Inc.
Phone: 408-437-3300

Georgia, Norcross
Chips and Technologies, Inc.
Phone: 404-662-5098

Illinois, Schaumburg
Chips and Technologies, Inc.
Phone: 708-397-4300

Massachusetts, Andover
Chips and Technologies, Inc.
Phone: 508-688-4600

Texas, Dallas
Chips and Technologies, Inc.
Phone: 214-702-9855

International

Germany
Munich
Chips and Technologies, Inc.
011-49-89-46-3074

Japan
Tokyo
Chips and Technologies, Inc.
011-813-3797-4311

Korea
Seoul
Chips and Technologies, Inc.
011-822-558-5559

Switzerland
Marin
Chip and Technologies, Inc.
011-41-38-33-8490 or
011-41-38-33-8206

Taiwan
Taipei
Chips and Technologies, Inc.
011-886-2-717-5595

Sales Representatives

United States

Arizona, Scottsdale
AzTECH Component Sales
Phone: 602-991-6300

California, Santa Clara
Magna Sales
Phone: 408-727-8753

California, San Diego
S.C. Cubed
Phone: 619-458-5808

California, Thousand Oaks
S.C. Cubed
Phone: 805-496-7307

California, Tustin
S.C. Cubed
Phone: 714-731-9206

Colorado, Wheat Ridge
Wescom Marketing, Inc.
Phone: 303-422-8957

Connecticut, Guilford
DataMark Inc.
Phone: 203-453-0575

Florida, Casselberry
Dyne-A-Mark Corp.
Phone: 407-831-2811

Florida, Ft. Lauderdale
Dyne-A-Mark Corp.
Phone: 305-771-6501

Georgia, Norcross
Bits, Inc.
Phone: 404-446-1155

Illinois, Hoffman Estates
Micro-Tex, Inc.
Phone: 312-382-3001

Indiana, Carmel
Giesting & Associates
Phone: 317-844-5222

Kentucky, Versailles
Giesting & Associates
Phone: 606-873-2330

Maryland, Annapolis
EES
Phone: 301-269-4234

Massachusetts, Woburn
Mill-Bern Assoc.
Phone: 617-932-3311

Michigan, Coloma
Giesting & Associates
Phone: 616-468-3308

Michigan, Comstock Park
Giesting & Associates
Phone: 616-784-9437

Michigan, Livonia
Giesting & Associates
Phone: 313-478-8106

Minnesota, Eden Prairie
High Tech Sales Assoc.
Phone: 612-944-7274

Missouri, Bridgeton
Centech
Phone: 314-291-4230

Missouri, Raytown
Centech
Phone: 816-358-8100

New Jersey, Moorestown
T.A.I.
Phone: 609-778-5353

New York, Commack
ERA, Inc.
Phone: 516-543-0510

New York, Syracuse
Pitronics
Phone: 315-455-7346

New York, Williamsville
Pitronics
Phone: 716-689-2378

Ohio, Cincinnati
Giesting & Associates
Phone: 513-385-1105

Ohio, Cleveland
Giesting & Associates
Phone: 216-261-9705

Oregon, Beaverton
L-Squared Limited
Phone: 503-629-8555

Pennsylvania, Pittsburg
Giesting & Associates
Phone: 412-828-3553

Texas, Austin
OM-Austin
Phone: 512-794-9971

Texas, Houston
OM Assoc., Inc.
Phone: 713-789-4426

Texas, Richardson
OM Assoc., Inc.
Phone: 214-690-6746

Utah, Salt Lake City
Wescom Marketing, Inc.
Phone: 801-269-0419

Washington, Kirkland
L-Squared Limited
Phone: 206-827-8555

Wisconsin, Waukesha
Micro-Tex, Inc.
Phone: 414-542-5352

Canada

Ontario, Kanata
Electro Source, Inc.
Phone: 613-592-3214

Ontario, Rexdale
Electro Source, Inc.
Phone: 416-675-4490

Quebec, Pointe Claire
Electro Source, Inc.
Phone: 514-630-7846

British Columbia, Vancouver
Electro-Source
Phone: 604-948-2314

Distributors

North America

United States

Alabama, Huntsville
 Reptron Electronics, Inc.
 Phone: 205-722-9500

Arizona, Tempe
 Anthem
 Phone: 602-966-6600

California, Chatsworth
 Anthem
 Phone: 818-700-1000

California, Fountain Valley
 Bell Microproducts
 Phone: 714-963-0667

California, Irvine
 Anthem
 Phone: 714-768-4444

California, Milpitas
 Bell Microproducts
 Phone: 408-434-1150

California, Rocklin
 Anthem
 Phone: 916-624-9744

California, San Diego
 Anthem
 Phone: 619-453-9005

California, San Jose
 Anthem
 Phone: 408-453-1200

Colorado, Englewood
 Anthem
 Phone: 303-790-4500

Colorado, Wheat Ridge
 Added Value
 Phone: 303-422-1701

Connecticut, Waterbury
 Anthem
 Phone: 203-575-1575

Florida, Ft. Lauderdale
 Reptron Electronics, Inc.
 Phone: 305-735-1112

Florida, Tampa
 Reptron Electronics
 Phone: 813-854-2351

Georgia, Norcross
 Reptron Electronics, Inc.
 Phone: 404-446-1300

JACO/QC
 Phone: 404-449-9508

Illinois, Schaumburg
 Anthem
 Phone: 708-884-0200

Reptron Electronics, Inc.
 Phone: 708-882-1700

Massachusetts, Wilmington
 Anthem
 Phone: 508-657-5170

Bell Microproducts
 Phone: 508-658-0222

Maryland, Columbia
 Anthem
 Phone: 301-995-6640

JACO/QC
 Phone: 301-995-6620

Michigan, Livonia
 Reptron Electronics, Inc.
 Phone: 313-525-2700

Minnesota, Eden Prairie
 Anthem
 Phone: 612-944-5454

Minnesota, Minnetonka
 Reptron Electronics, Inc.
 Phone: 612-938-0000

New Jersey, Pinebrook
 Anthem
 Phone: 201-227-7960

New York, Commack
 Anthem
 Phone: 516-864-6600

New York, Long Island
 JACO/QC
 Phone: 516-273-5500

North Carolina, Raleigh
 Reptron Electronics, Inc.
 Phone: 919-870-5189

JACO/QC
 Phone: 919-876-7767

Ohio, Columbus
 EMC
 Phone: 614-299-4161

Ohio, Solon
 Reptron Electronics, Inc.
 Phone: 216-349-1415

Ohio, Worthington
 Reptron Electronics, Inc.
 Phone: 614-436-6675

Oregon, Beaverton
 Anthem
 Phone: 503-643-1114

Pennsylvania, Horsham
 Anthem
 Phone: 215-443-5150

Texas, Addison
 JACO/QC
 Phone: 214-733-4300

Texas, Austin
 JACO/QC
 Phone: 512-835-0220

Texas, Richardson
 Anthem
 Phone: 214-238-7100

Texas, Sugarland
 JACO/QC
 Phone: 713-240-2255

Utah, Salt Lake City
 Anthem
 Phone: 801-973-8555

Utah, West Valley
 Added Value
 Phone: 801-975-9500

Washington, Bothell
 Anthem
 Phone: 206-483-1700

Canada

Ontario, Rexdale
 Valtrie Marketing
 Phone: 416-798-2555

Distributors

International

Asia/Pacific

Australia

Melbourne
Zatek Components, Ltd.
3-763-8899

Sidney

Zatek Components, Ltd.
2-895-5534

Hong Kong

Kwung Tong
Wong's KK Ltd.
85-2-345-0121

India

Bombay
Silicon Electronics
91-22-243-460

New Dehli

Ajay Jain
91-11-686-3044

Israel

Tel-Aviv
CVS
97-23-544-7475

Japan

Kawasaki
CTC Components Systems Co., Ltd.
81-44-852-1825

Tokyo

ASCII Corporation
81-3-486-9188

Korea

Seoul
Nae Wae Semiconductor
82-2-704-0103

Malaysia

Penang
Dynamar
60-4-637-292

Selangor

03-776-7140/199

Singapore

Singapore
Technology Distribution
PTE Ltd.
65-368-6065

Taiwan

Taipei
Ally, Inc.
886-2-562-0828

World Peace

Industrial Co., Ltd.
88-62-786-5311

Thailand

Bangkok
Grawinner
66-2-215-8742

Europe

Belgium

Zaventem
ACAL Auriema Belgium
32-2-720-5983

Denmark

Herlev
Nordisk, Elektronik A/S
454-284-2000

Finland

Helsinki
OY Fintronic AB
35-806-926022

France

Le Chesnay
A2M
33-1-395-49113

Germany

Neteltal
Rein Elektronik GmbH
49-215-373300

Italy

Milano
Moxel S.r.L.
39-2-612-90521

Netherlands

Eindhoven
ACAL Auriema Nederland B.V.
31-40-816-565

Norway

Hvalstad
Nordisk Elektronik A/S
47-284-6210

Spain

Madrid
Compania Electronica de Tecnicas
Aplicadas, S.A.
34-1-754-3001

Barcelona

Compania Electronica de Tecnicas
Aplicadas, S.A.
34-3-300-7712

Sweden

Kista
Nordisk Elektronik A.B.
46-8-703-4630

Switzerland

Dietikon
DataComp AG
41-1-740-5140

United Kingdom

Berkshire, Maidenhead
Magna Technology
44-734-880211

Oxfordshire

Thames Components
8-44-261188

Americas

Mexico

Gaudalajara
Sonika
52-364-74250



Chips and Technologies

3050 Zanker Road

San Jose, California 95134

Phone: 408/434-0600

Telex: 272929 CHIPS UR

FAX: 408/434-6452

Publication No.: UG80

Stock No.: 050080-001

Revision No.: 1.0

F8680 PC/CHIP™

Small LCD Panel Support Application Note

PC/CHIP™

CHIPS®



F8680 PC/CHIP™

Small LCD Panel Support Application Note

Copyright Notice

Software Copyright © 1991, Chips and Technologies, Inc.

Manual Copyright © 1991, Chips and Technologies, Inc.

All Rights Reserved.

Printed in U.S.A.

Trademark Acknowledgement

CHIPS® and the CHIPS logotype are registered trademarks of Chips and Technologies, Inc.

PC/CHIP™, SuperState™, Virtual I/O™, Virtual Interrupts™ and Visual Map™ are trademarks of Chips and Technologies, Inc.

Disclaimer

This publication is copyrighted by Chips and Technologies, Inc. You may not reproduce, transmit, transcribe, store in a retrieval system, or translate into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, any part of this publication without express written permission of Chips and Technologies, Inc.

The information contained in this document is being issued in advance of the production cycle for the device(s). The parameters for the device(s) may change before final production.

Chips and Technologies, Inc. makes no representations or warranties regarding the contents of this document. We reserve the right to revise the document or make changes in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or changes.

The information contained in this document is provided for general use by our customers. Our customer should be aware that the personal computer field is the subject of many patents. Our customers should ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Chips and Technologies, Inc. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

Restricted Rights and Limitations

Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.277-7013

Chips and Technologies, Inc.

3050 Zanker Road

San Jose, California 95134

Phone: 408-434-0600



F8680 PC/CHIP™ Small LCD Panel Support

The F8680 PC/CHIP™ microchip is a highly integrated processor containing all peripheral devices necessary to implement a PC subsystem. One of the peripherals included as part of the single-chip solution is a graphics controller that supports both LCD and CRT modes. This application note addresses one aspect of LCD panel support: panels smaller than the standard 640 pixel x 200 line panel.

Overview

The CGA display memory is sized for a 640 x 200 panel; a smaller panel allows only a portion of this memory to be displayed. Therefore, this application note deals with making the smaller panel serve as a *window* into the larger display memory. In order to implement this window, a start address from which to display memory must be determined. Also, the clock frequency provided to the panel must be adjusted to the size of the panel. Both of these issues will be discussed.

An LCD panel subsystem accepts display data one line of pixels at a time. These bits are transmitted serially to the panel, and are received in a shift register whose capacity corresponds to the size of the panel it supports. With a 640 x 200 panel, 640 bits of display data are shifted in before the graphics controller generates a latch pulse to load the shift register pixels to the current line on the LCD panel.

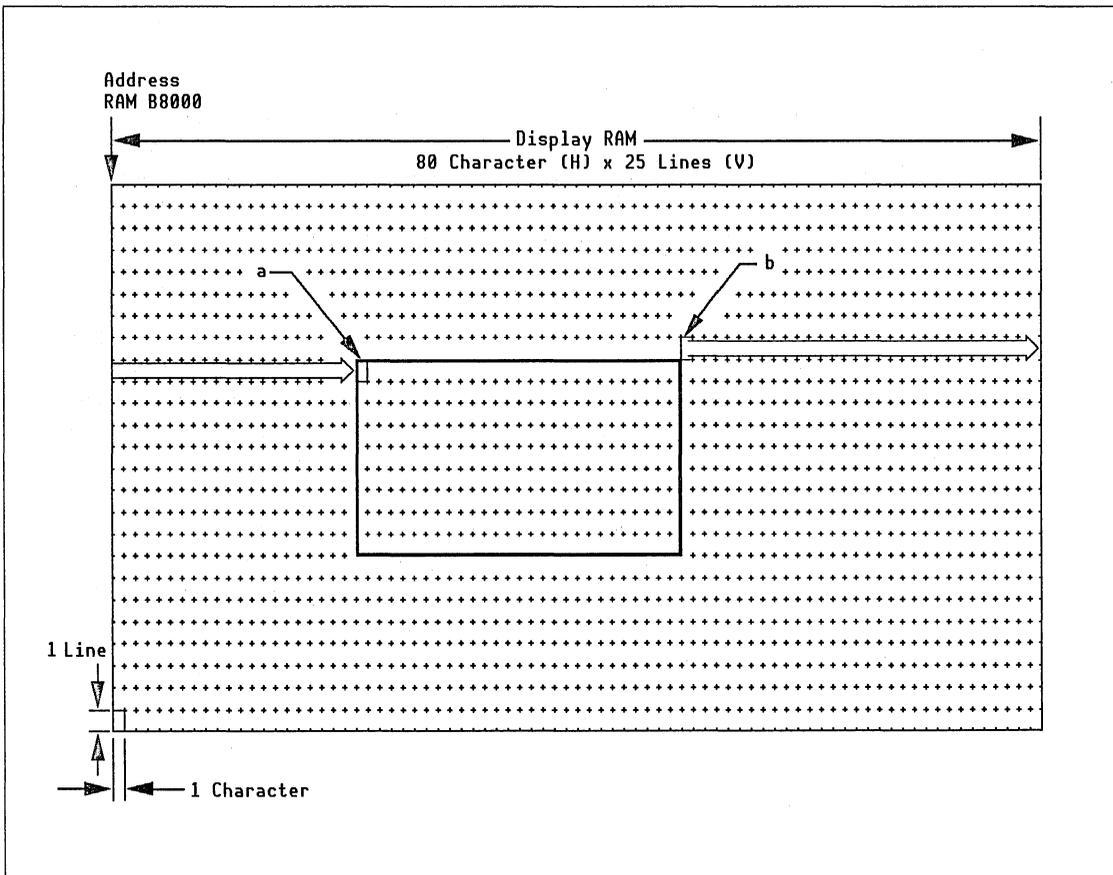
If a smaller panel of dimensions H pixels x V lines is used, the shift register capacity is only H bits; it cannot hold a full line of 640 pixels. It is important to note, however, that if 640 pixels *are* shifted in to a smaller shift register, the first pixels are ignored; only the last H pixels are retained. It is this fact that permits the following scheme to work.

Note: Some panels do not incorporate the shift register design described here. Instead, they use a memory and a pointer to a memory location. For panels without shift registers, the last H pixels are not retained; the first H pixels are. However, the scheme presented will work for them also, if an appropriate modification is made to the start address.

Implementation

If a panel size of H pixels x V lines is assumed, Figure 1 illustrates the graphics memory situation.

Figure 1. LCD Memory Map



It would be ideal to start displaying memory at point a , display for a duration of H pixels, increment the start address by 640, and repeat the process for the next line. Unfortunately, CGA-compatible graphics provide no mechanism to offset the start address by a specific amount (640 in this case) for every line.

However, it is possible to use a mid-line start address in order to take advantage of the panel shift register design. Instead of using a as the start address, point b on the previous line is taken as the start address. Since the last H pixels of data before the point exactly one row down from b comprise the valid data to display, only these last H pixels must be in the horizontal drivers when the latch pulse signal is generated.

Note that if a panel without a shift register is used, the start address stays at point a , because the panel saves the first H pixels.

Determining Start Address, Graphics Mode (640 x 200 and 320 x 200)

Suppose the window begins displaying at pixel location (X, Y) . Logically, the start address would be defined as follows:

$$\text{Start Address} = \frac{(640 \times Y) + X}{16}$$

However, if the above scheme is used to display data, the starting address becomes:

$$\text{Start Address} = \frac{(640 \times Y) + X - (640 - H)}{16}$$

The quantity $(640 - H)$ is subtracted from the start address so that by the time the next latch pulse is generated, the correct data is in the horizontal drivers, i.e. the H pixels starting at location (X, Y) . If the panel without a shift register is used, this quantity would not need to be subtracted.

The address is divided by 16, as the display memory is addressable only on word boundaries.

Determining Start Address, Text Mode

In text mode, for each character there are two corresponding bytes: one for the character code and one for the character attribute. Therefore, for 80 x 25 text mode the start address is defined as:

$$\text{Start Address} = (Y \times 80 + X - (80 + \frac{H}{8})) \times 2$$

For 40x25 mode, this becomes:

$$\text{Start Address} = (Y \times 40 + X - (40 + \frac{H}{16})) \times 2$$

Again, the quantities $(80 + H/8)$ and $(40 + H/16)$ would not need to be subtracted if the panel without a shift register were used.

Once determined, the start address must be written to ports 3D4/3D5 at indexes 0C and 0D.

Timing Considerations

Pixels are transmitted serially, in groups of four pixels. If the panel were H pixels wide, H/4 clock cycles would be needed between latch pulses (4 pixels/clock). However, using point *b* as the start address, H/4 clock cycles between line pulses results in unwanted data on the panel.

Therefore, the clock timing for a 640-pixel panel must be maintained so that the incorrect data is shifted out and the correct data is shifted in. This means clocking the panel 160 times (640/4) between latch pulses.

Determining Clock Frequency

The clock frequency will differ depending on the size of panel used. The following formula should be applied:

$$f_{\text{clock}} = R_{\text{rfs}} \times V \times \frac{640}{4}$$

where: R_{rfs} is the panel refresh rate.

This clock may be supplied from the DOTCLOCK output of the F8680 microprocessor if the corresponding frequency is available on PS4.

Regional Sales Offices

United States

California, Irvine
Chips and Technologies, Inc.
Phone: 714-852-8721

California, San Jose
Chips and Technologies, Inc.
Phone: 408-437-3300

Georgia, Norcross
Chips and Technologies, Inc.
Phone: 404-662-5098

Illinois, Schaumburg
Chips and Technologies, Inc.
Phone: 708-397-4300

Massachusetts, Andover
Chips and Technologies, Inc.
Phone: 508-688-4600

Texas, Dallas
Chips and Technologies, Inc.
Phone: 214-702-9855

International

Germany
Munich
Chips and Technologies, Inc.
011-49-89-46-3074

Japan
Tokyo
Chips and Technologies, Inc.
011-813-3797-4311

Korea
Seoul
Chips and Technologies, Inc.
011-822-558-5559

Switzerland
Marin
Chip and Technologies, Inc.
011-41-38-33-8490 or
011-41-38-33-8206

Taiwan
Taipei
Chips and Technologies, Inc.
011-886-2-717-5595

Sales Representatives

United States

Arizona, Scottsdale
AzTECH Component Sales
Phone: 602-991-6300

California, Santa Clara
Magna Sales
Phone: 408-727-8753

California, San Diego
S.C. Cubed
Phone: 619-458-5808

California, Thousand Oaks
S.C. Cubed
Phone: 805-496-7307

California, Tustin
S.C. Cubed
Phone: 714-731-9206

Colorado, Wheat Ridge
Wescom Marketing, Inc.
Phone: 303-422-8957

Connecticut, Guilford
DataMark Inc.
Phone: 203-453-0575

Florida, Casselberry
Dyne-A-Mark Corp.
Phone: 407-831-2811

Florida, Ft. Lauderdale
Dyne-A-Mark Corp.
Phone: 305-771-6501

Georgia, Norcross
Bits, Inc.
Phone: 404-446-1155

Illinois, Hoffman Estates
Micro-Tex, Inc.
Phone: 312-382-3001

Indiana, Carmel
Giesting & Associates
Phone: 317-844-5222

Kentucky, Versailles
Giesting & Associates
Phone: 606-873-2330

Maryland, Annapolis
EES
Phone: 301-269-4234

Massachusetts, Woburn
Mill-Bern Assoc.
Phone: 617-932-3311

Michigan, Coloma
Giesting & Associates
Phone: 616-468-3308

Michigan, Comstock Park
Giesting & Associates
Phone: 616-784-9437

Michigan, Livonia
Giesting & Associates
Phone: 313-478-8106

Minnesota, Eden Prairie
High Tech Sales Assoc.
Phone: 612-944-7274

Missouri, Bridgeton
Centech
Phone: 314-291-4230

Missouri, Raytown
Centech
Phone: 816-358-8100

New Jersey, Moorestown
T.A.I.
Phone: 609-778-5353

New York, Commack
ERA, Inc.
Phone: 516-543-0510

New York, Syracuse
Pitronics
Phone: 315-455-7346

New York, Williamsville
Pitronics
Phone: 716-689-2378

Ohio, Cincinnati
Giesting & Associates
Phone: 513-385-1105

Ohio, Cleveland
Giesting & Associates
Phone: 216-261-9705

Oregon, Beaverton
L-Squared Limited
Phone: 503-629-8555

Pennsylvania, Pittsburg
Giesting & Associates
Phone: 412-828-3553

Texas, Austin
OM-Austin
Phone: 512-794-9971

Texas, Houston
OM Assoc., Inc.
Phone: 713-789-4426

Texas, Richardson
OM Assoc., Inc.
Phone: 214-690-6746

Utah, Salt Lake City
Wescom Marketing, Inc.
Phone: 801-269-0419

Washington, Kirkland
L-Squared Limited
Phone: 206-827-8555

Wisconsin, Waukesha
Micro-Tex, Inc.
Phone: 414-542-5352

Canada

Ontario, Kanata
Electro Source, Inc.
Phone: 613-592-3214

Ontario, Rexdale
Electro Source, Inc.
Phone: 416-675-4490

Quebec, Pointe Claire
Electro Source, Inc.
Phone: 514-630-7846

British Columbia, Vancouver
Electro-Source
Phone: 604-948-2314

Distributors

North America

United States

Alabama, Huntsville
Reptron Electronics, Inc.
Phone: 205-722-9500

Arizona, Tempe
Anthem
Phone: 602-966-6600

California, Chatsworth
Anthem
Phone: 818-700-1000

California, Fountain Valley
Bell Microproducts
Phone: 714-963-0667

California, Irvine
Anthem
Phone: 714-768-4444

California, Milpitas
Bell Microproducts
Phone: 408-434-1150

California, Rocklin
Anthem
Phone: 916-624-9744

California, San Diego
Anthem
Phone: 619-453-9005

California, San Jose
Anthem
Phone: 408-453-1200

Colorado, Englewood
Anthem
Phone: 303-790-4500

Colorado, Wheat Ridge
Added Value
Phone: 303-422-1701

Connecticut, Waterbury
Anthem
Phone: 203-575-1575

Florida, Ft. Lauderdale
Reptron Electronics, Inc.
Phone: 305-735-1112

Florida, Tampa
Reptron Electronics
813-854-2351

Georgia, Norcross
Reptron Electronics, Inc.
Phone: 404-446-1300

JACO/QC
Phone: 404-449-9508

Illinois, Schaumburg
Anthem
Phone: 708-884-0200

Reptron Electronics, Inc.
Phone: 708-882-1700

Massachusetts, Wilmington
Anthem
Phone: 508-657-5170

Bell Microproducts
Phone: 508-658-0222

Maryland, Columbia
Anthem
Phone: 301-995-6640

JACO/QC
Phone: 301-995-6620

Michigan, Livonia
Reptron Electronics, Inc.
Phone: 313-525-2700

Minnesota, Eden Prairie
Anthem
Phone: 612-944-5454

Minnesota, Minnetonka
Reptron Electronics, Inc.
Phone: 612-938-0000

New Jersey, Pinebrook
Anthem
Phone: 201-227-7960

New York, Commack
Anthem
Phone: 516-864-6600

New York, Long Island
JACO/QC
Phone: 516-273-5500

North Carolina, Raleigh
Reptron Electronics, Inc.
Phone: 919-870-5189

JACO/QC
Phone: 919-876-7767

Ohio, Columbus
EMC
Phone: 614-299-4161

Ohio, Solon
Reptron Electronics, Inc.
Phone: 216-349-1415

Ohio, Worthington
Reptron Electronics, Inc.
Phone: 614-436-6675

Oregon, Beaverton
Anthem
Phone: 503-643-1114

Pennsylvania, Horsham
Anthem
Phone: 215-443-5150

Texas, Addison
JACO/QC
Phone: 214-733-4300

Texas, Austin
JACO/QC
Phone: 512-835-0220

Texas, Richardson
Anthem
Phone: 214-238-7100

Texas, Sugarland
JACO/QC
Phone: 713-240-2255

Utah, Salt Lake City
Anthem
Phone: 801-973-8555

Utah, West Valley
Added Value
Phone: 801-975-9500

Washington, Bothell
Anthem
Phone: 206-483-1700

Canada

Ontario, Rexdale
Valtrie Marketing
Phone: 416-798-2555

Distributors

International

Asia/Pacific

Australia
Melbourne
Zatek Components, Ltd.
3-763-8899

Sidney
Zatek Components, Ltd.
2-895-5534

Hong Kong
Kwung Tong
Wong's KK Ltd.
85-2-345-0121

India
Bombay
Silicon Electronics
91-22-243-460

New Dehli
Ajay Jain
91-11-686-3044

Israel
Tel-Aviv
CVS
97-23-544-7475

Japan
Kawasaki
CTC Components Systems Co., Ltd.
81-44-852-1825

Tokyo
ASCII Corporation
81-3-486-9188

Korea
Seoul
Nae Wae Semiconductor
82-2-704-0103

Malaysia
Penang
Dynamar
60-4-637-292

Selangor
03-776-7140/199

Singapore
Singapore
Technology Distribution
PTE Ltd.
65-368-6065

Taiwan
Taipei
Ally, Inc.
886-2-562-0828

World Peace
Industrial Co., Ltd.
88-62-786-5311

Thailand
Bangkok
Grawinner
66-2-215-8742

Europe

Belgium
Zaventem
ACAL Auriema Belgium
32-2-720-5983

Denmark
Herlev
Nordisk, Elektronik A/S
454-284-2000

Finland
Helsinki
OY Fintronic AB
35-806-926022

France
Le Chesnay
A2M
33-1-395-49113

Germany
Neteltal
Rein Elektronik GmbH
49-215-373300

Italy
Milano
Moxel S.r.L.
39-2-612-90521

Netherlands
Eindhoven
ACAL Auriema Nederland B.V.
31-40-816-565

Norway
Hvalstad
Nordisk Elektronik A/S
47-284-6210

Spain
Madrid
Compania Electronica de Tecnicas
Aplicadas, S.A.
34-1-754-3001

Barcelona
Compania Electronica de Tecnicas
Aplicadas, S.A.
34-3-300-7712

Sweden
Kista
Nordisk Elektronik A.B.
46-8-703-4630

Switzerland
Dietikon
DataComp AG
41-1-740-5140

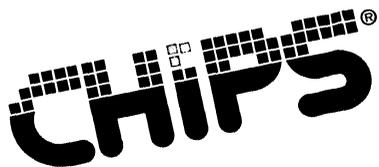
United Kingdom

Berkshire, Maidenhead
Magna Technology
44-734-880211

Oxfordshire
Thames Components
8-44-261188

Americas

Mexico
Gaudalajara
Sonika
52-364-74250



Chips and Technologies

3050 Zanker Road

San Jose, California 95134

Phone: 408/434-0600

Telex: 272929 CHIPS UR

FAX: 408/434-6452

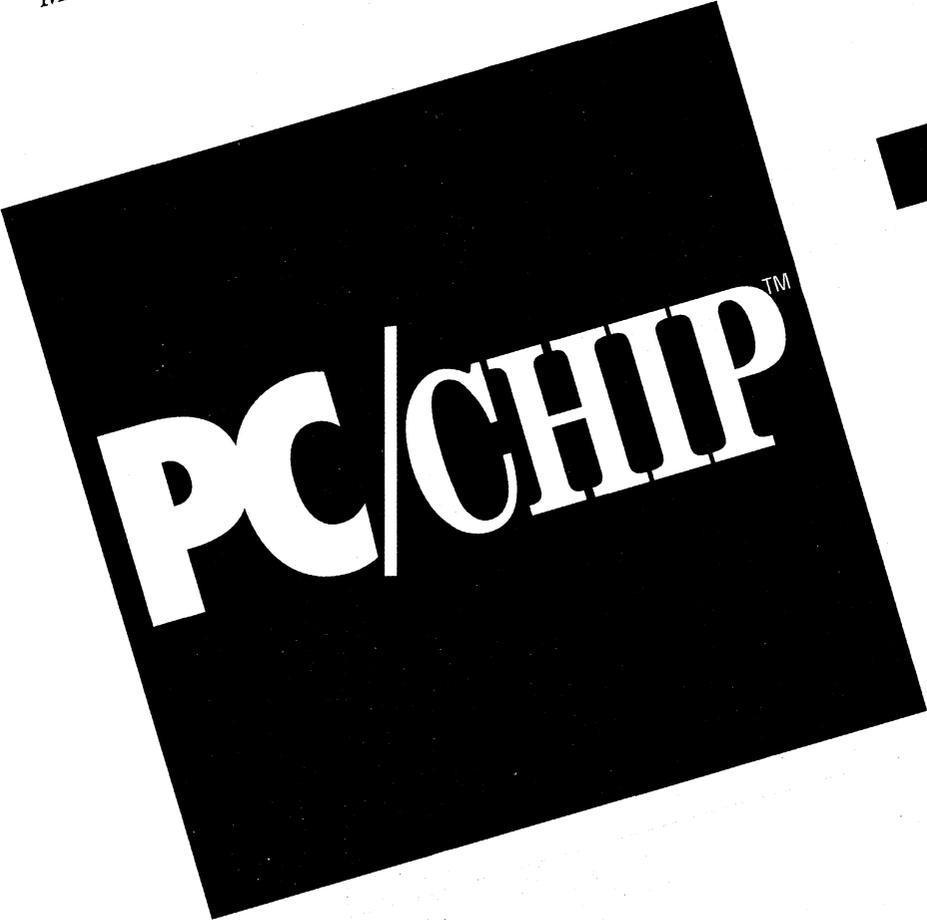
Publication No.: AN58

Stock No.: 02058-001

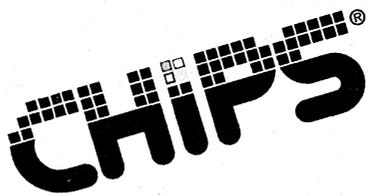
Revision No.: 1.0

F8680 PC/CHIP™

Memory Configuration Application Note



PC/CHIP™



CHIPS®



F8680 PC/CHIP™

Memory Configuration Application Note

Copyright Notice

Software Copyright © 1991, Chips and Technologies, Inc.

Manual Copyright © 1991, Chips and Technologies, Inc.

All Rights Reserved.

Printed in U.S.A.

Trademark Acknowledgement

CHIPS® and the CHIPS logotype are registered trademarks of Chips and Technologies, Inc.

PC/CHIP™, SuperState™, Virtual I/O™, Virtual Interrupts™ and Visual Map™ are trademarks of Chips and Technologies, Inc.

Disclaimer

This publication is copyrighted by Chips and Technologies, Inc. You may not reproduce, transmit, transcribe, store in a retrieval system, or translate into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, any part of this publication without express written permission of Chips and Technologies, Inc.

The information contained in this document is being issued in advance of the production cycle for the device(s). The parameters for the device(s) may change before final production.

Chips and Technologies, Inc. makes no representations or warranties regarding the contents of this document. We reserve the right to revise the document or make changes in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or changes.

The information contained in this document is provided for general use by our customers. Our customer should be aware that the personal computer field is the subject of many patents. Our customers should ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Chips and Technologies, Inc. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

Restricted Rights and Limitations

Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.277-7013

Chips and Technologies, Inc.
3050 Zanker Road
San Jose, California 95134
Phone: 408-434-0600



F8680 PC/CHIP™ Memory Configuration

The F8680 PC/CHIP™ microchip is designed to meet the special needs of diverse market segments. To this end, the chip uses an extremely flexible memory controller to support the various configurations that may be used. This section describes the features of this memory controller through examples of its use in two different memory configurations.

The major components of the memory controller are the Bank Switching Registers, the 34 Memory Mapping Registers, and the Bank Select Registers. The Bank Switching Registers are used to generate the 26-bit address that allows access to any location in the 64MB address space. Bank switching must first be enabled (CREG 0C) and is valid only at certain segments of the upper 384kB in the first 1MB. The 34 Memory Mapping Registers describe the memory in the first 2MB (with 32kB blocks per register) or 4MB (with 64kB blocks per register). These registers also point to one of five Bank Select Registers. The Bank Select Registers are only concerned with types of memory and memory cycles to be performed. It is through a combination of Memory Mapping and Bank Select Registers that the address space is merged with the memory types to define fully the memory in the system.

The following description discusses only the memory controller configuration. None of the other configuration registers that must be set to produce a working system are discussed.

The following two examples show one typical and one somewhat atypical memory configuration. Both configurations require the use of a 32k x 8 120ns SRAM for graphics. This graphics RAM has its own address and data lines, but its Bank Switching Register must still be loaded.

Both examples use 64kB memory blocks (selected through CREG 0D).

Memory Configuration Example 1

This system will use the following memory types:

- Two 512k x 8 DRAMs
- One 128k x 8 ROM
- One PCMCIA Memory Card.

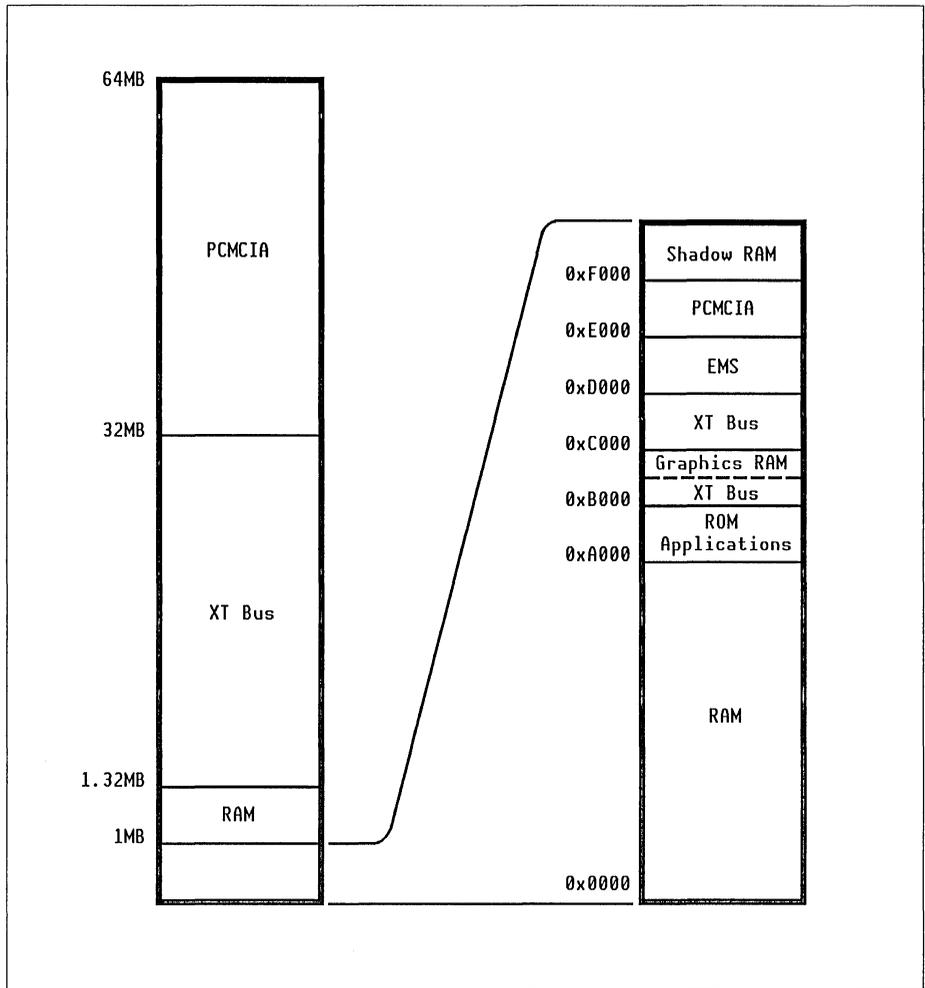
The lower 64kB of the ROM will be reserved for applications. The upper 64kB is used for BIOS and extensions and will be shadowed at segment 0xF000.

There is 640kB of conventional memory, 64kB for shadow ROM, and the remaining 320kB above 1MB. The ROM applications will be mapped into segment 0xA000. The PCMCIA card is mapped into the upper 32MB of the address space. Access to the card is made through segment 0xE000 with the bank switching feature enabled. Bank switching allows access to any byte in the 64MB address space of the chip.

Memory Map

Figure 1 shows how the memory is mapped.

Figure 1. Memory Map

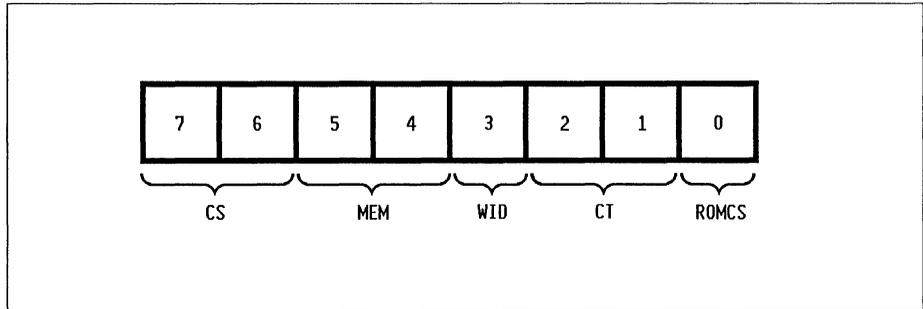


The following sections explain the set-up procedure.

Bank Select Registers Setup

The first step is to set up the five Bank Select Registers to support the configuration. These registers are mapped into the configuration space at CREGs 20-24. Their format is shown in Figure 2.

Figure 2. Bank Select Register Format



<i>bits:</i> 7:6	CS1:0	Chip Select to activate 00 = Chip select 1 01 = Chip select 2 10 = Chip select 3 11 = No chip select (XT bus - default)
5:4	MEM1:0	Memory Address Multiplexing Used 00 = 256Kb deep (256Kb x 1 or 256Kb x 4 - default) 01 = 1Mb deep 10 = 4Mb deep 11 = Reserved
3	WID	Bank Width 0 = 1 byte wide (default) 1 = 2 bytes wide
2:1	CT1:0	Cycle Type to Perform 00 = XT bus cycle (default) 01 = DRAM cycle 10 = SRAM/pseudo-SRAM cycle 11 = Memory card
0	ROMCS	ROM Chip Select Enable 0 = No ROM chip select 1 = Activate ROMCS line (default)

Because there are two banks of RAM, two of the Bank Select Registers need to have chip selects associated with them. The others can point to the XT bus with ROMCS either enabled or disabled. For the system in this example, the Bank Select Registers are set up as shown in Table 1.

Table 1. *Bank Select Register Settings*

CREG	Bank Select	Setting	Function
0x20	BS0	11000001	XT bus cycle, ROMCS active
0x21	BS1	00000100	DRAM cycle, CS20 active
0x22	BS2	01000100	DRAM cycle, CS21 active
0x23	BS3	11000000	XT bus cycle, no CS
0x24	BS4	11000000	XT bus cycle, no CS

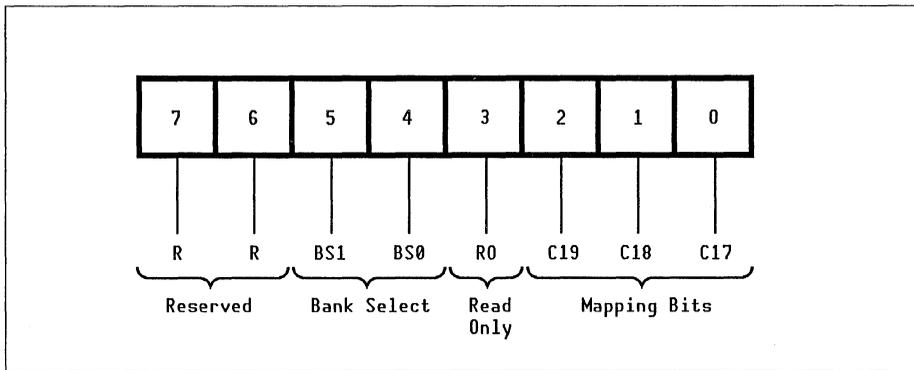
Note: There is no special MEM setting for 512kB deep DRAM. However, the preliminary specifications for the Hitachi HM514800 and Toshiba TC514800 512k x 8 DRAMs indicate that the row and column addresses the DRAM needs will be compatible with the 256kB deep setting.

The 34 Memory Mapping registers select one of Bank Select registers 0-3. Bank Select 4 is the *out-of-range* register, which specifies the memory type in the range from 4MB to 16MB.

Memory Mapping Registers Setup

After the Bank Select Registers are set up, the 34 Memory Mapping Registers must be programmed to point to them. The format of these registers is shown in Figure 3.

Figure 3. Memory Mapping Register Format



<i>bits:</i> 7:6	R	Reserved
5:4	BS1:0	Bank Select 00 = Use Bank Select Register 0 at CREG 20h 01 = Use Bank Select Register 1 at CREG 21h 10 = Use Bank Select Register 2 at CREG 22h 11 = Use Bank Select Register 3 at CREG 23h
5:3	RO	Read-Only 0 = Read/write memory range 1 = Read-only memory range
2:0	XOR	Complement A19-A17 000 = Invert no bits 100 = Invert bit 19, but not 18 or 17 010 = Invert bit 18, but not 19 or 17 110 = Invert bits 19 and 18, but not 17 . . 111 = Invert bits 19, 18 and 17

64kB blocks will be used, so each register corresponds to one 64kB segment. Table 2 shows how the Memory Mapping Registers are set up.

Table 2. *Memory Mapping Register Settings*

CREG	Segment	Register Values	Access Location
0xD4	0x14000	100011	2nd bank
0xD3	0x13000	100000	2nd bank
0xD2	0x12000	100011	2nd bank
0xD1	0x11000	100010	2nd bank
0xD0	0x10000	100011	2nd bank
0xCF	0x0F000	101000	2nd bank (shadow RAM)
0xCE	0x0E000	000000	PCMCIA (bank switched)
0xCD	0x0D000	000000	EMS (bank switched)
0xCC	0x0C000	110000	XT bus
0xCB	0x0B000	110000	XT bus at B000, graphics RAM at B800
0xCA	0x0A000	001000	Application programs (ROM)
0xC9	0x09000	100000	2nd bank
0xC8	0x08000	100000	2nd bank
0xC7	0x07000	010000	1st bank
0xC6	0x06000	010000	1st bank
0xC5	0x05000	010000	1st bank
0xC4	0x04000	010000	1st bank
0xC3	0x03000	010000	1st bank
0xC2	0x02000	010000	1st bank
0xC1	0x01000	010000	1st bank
0xC0	0x00000	010000	1st bank

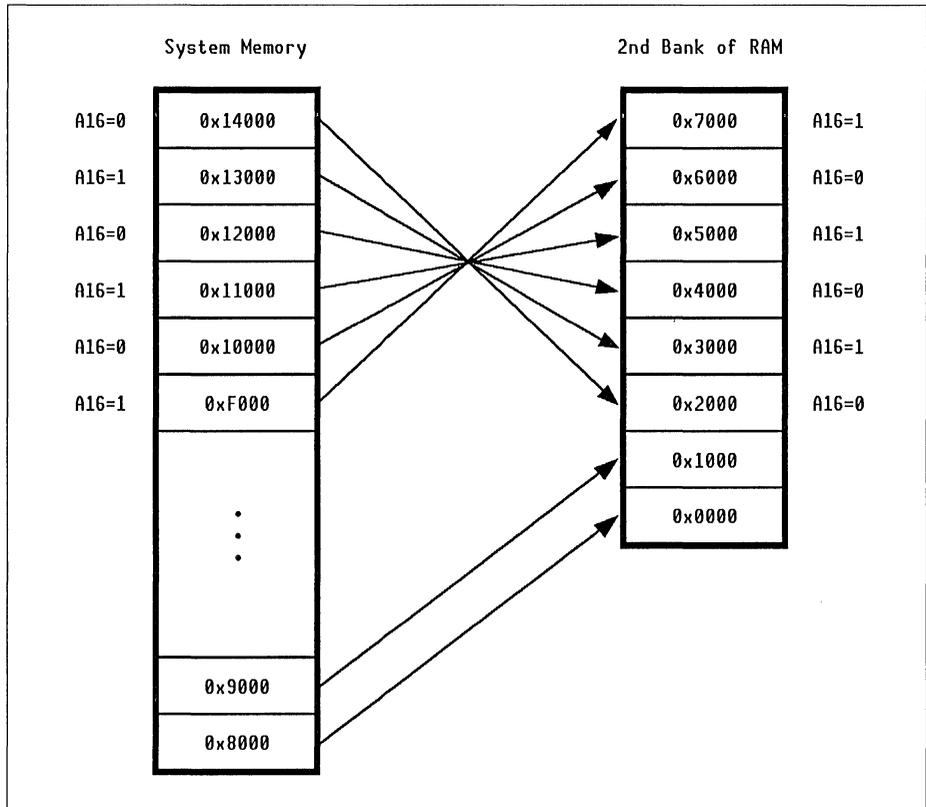
Note that the RO (read-only) bit is activated for those segments that have a physical ROM attached or need write protection (as with BIOS shadowing). For segment 0xF000, this would mean setting the RO bit once the ROM has been shadowed. Also note that although segment 0xB000 has the configuration bits indicating an XT bus cycle, a graphics RAM cycle is forced for any address in the 0xB8000 to 0xC0000 range.

Accesses to locations in segment 0xA000 activate the ROM, which tends to be slower than RAM. One way to speed up the system is to map one of the segments of RAM above 1MB to segment 0xA000, copy the applications to that segment during system initialization, and make the segment read-only. This setup would enable the user to run the applications in faster memory (RAM vs. ROM) while still providing the necessary write protection.

The PCMCIA card will be accessed through bank switching. Since the operation is being performed on a 64kB block, the upper 10 bits (A25:16) are substituted by the bits written to CREGs A3 and AF. A15:0 are unchanged.

The XOR of address bits 17, 18, and 19 is used to map the ROM BIOS and addresses above the 1MB limit to the second bank of RAM. However, because address bit 16 cannot be changed, each block in the address space can be mapped only to certain blocks in the physical RAM. The mapping shown in Figure 4 obeys this rule. Address bit 16 is the same for the destination segment to which the block is mapped.

Figure 4. Mapping of Second Bank



The remaining Memory Mapping Registers are set as shown in Table 3.

Table 3. Memory Mapping Register Settings

CREG	Segment	Register Values	Access Location
0xD5-0xDF	0x15000-0x1F000	110000	XT bus
0xE0	0x20000-0x2F000	110000	XT bus
0xF0	0x30000-0x3F000	110000	XT bus

With the Memory Mapping Registers set as noted, this system will use the mapping shown in Table 4.

Table 4. *Memory Mapping*

Memory Range	Usage
32-64MB	PCMCIA Memory Card
16-32MB	XT Bus (default)
4-16MB	XT Bus (through BS4)
1.32-4MB	XT Bus (through Mapping Register)
1-1.32MB	System RAM
960kB-1MB	ROM Shadow
704-960kB	Graphics RAM, XT Bus, Bank Switching
640-704kB	ROM Applications
0-640kB	System RAM

Memory Configuration Example 2

This example will use the following memory types:

- Two 512k x 8 PSRAMs
- One 128k x 8 PSRAM
- One 64k x 8 ROM
- One PCMCIA Card.

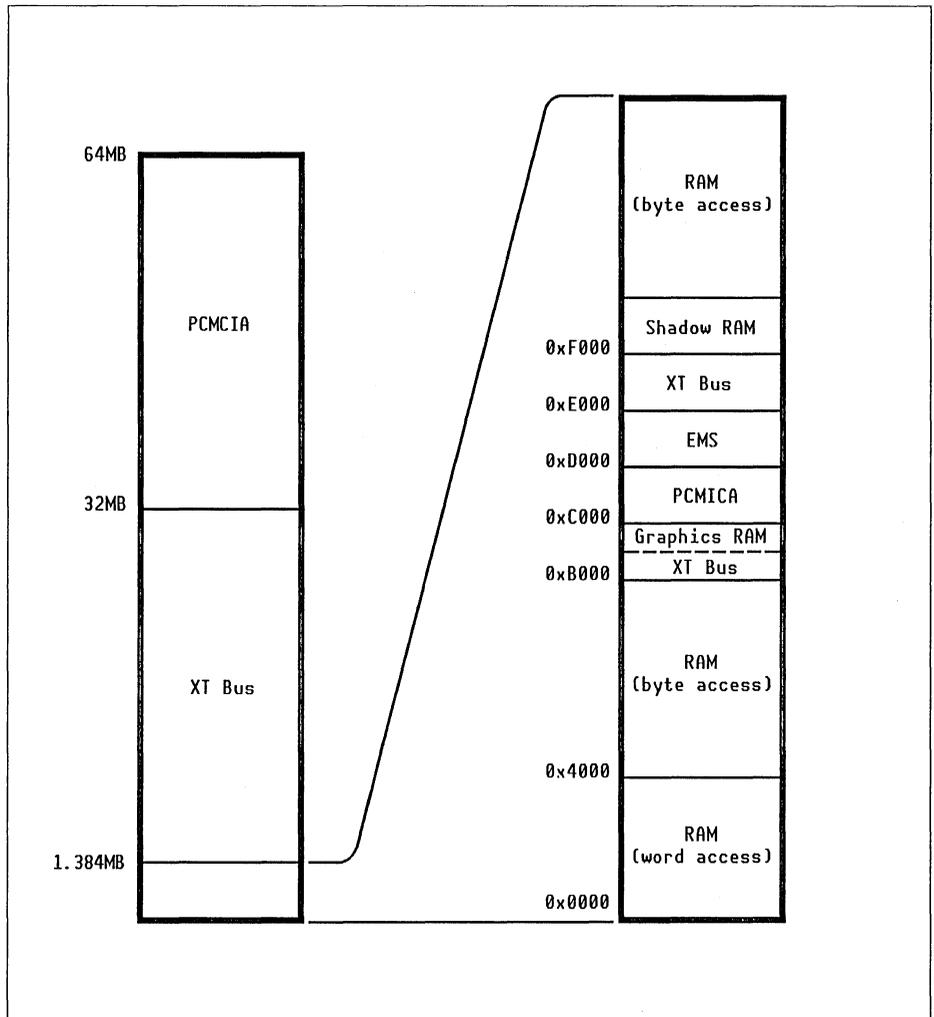
This system is somewhat out of the ordinary. It is used to show the flexibility of the memory controller and is not a typical system.

The system will be configured such that the first 256kB of RAM will be set up to allow for word accesses. The next 384kB will be configured for single-byte accesses. The remaining 512kB of RAM will be used for ROM shadow, HIMEM support, and expanded memory.

Memory Map

Figure 5 illustrates how the memory is mapped.

Figure 5. *Memory Mapping*



Bank Select Registers Setup

Because this system supports both word and byte accesses, assigning chip selects to banks is a bit trickier than for a typical configuration. One way to make this assignment is the following:

1. Assign CS20 (and BS1) to be active in the word access area (the first 256kB).
2. Assign CS21 (and BS2) to be active in the byte access area (the next 384kB).
3. Assign CS22 (and BS3) to be active during access of shadow RAM, HIMEM, and expanded memory (at segments 0xA000, 0xF000—0x15000).

With this setup, there will be two chip selects (CS20 and CS21) associated with the 512k x 8 PSRAM. Logically, the PSRAM can be viewed as being part of two banks, one for word accesses and the other for byte accesses. The 128k x 8 PSRAM will be used on the word accesses in the first 256kB, and will be hooked up to CS20. The other 512k x 8 PSRAM will be used for shadow RAM, HIMEM, and expanded memory and will be connected to CS22.

Actually, the 512k x 8 PSRAM could be connected to CS20 alone to produce a working system, because the high-byte memory control signals would not be activated on single-byte accesses. However, assigning two chip selects to the PSRAM allows one to be used as the output control selector of the 2-input MUX used on the high address bit of the RAM. This MUX is needed to substitute A0 for A19 when making byte accesses. See Figure 7 for details on the memory interface.

The Bank Select Registers are written as shown in Table 5.

Table 5. *Bank Select Register Settings*

CREG		Setting	Function
0x20	BS0	11000000	XT bus cycle, no CS
0x21	BS1	00001100	SRAM/PSRAM cycle, CS20, 2 byte
0x22	BS2	01000100	SRAM/PSRAM cycle, CS21, 1 byte
0x23	BS3	10000100	SRAM/PSRAM cycle, CS22
0x24	BS4	11000000	XT bus cycle, no CS

On reset Bank Select Register 0 contains 01100001. Change this register to the above value only when all memory and other configuration registers have been set and the BIOS has been shadowed. If this change is made any earlier, the system will crash.

Memory Mapping Register Setup

The setup for the 34 Memory Mapping Registers follows logically from the Bank Select Registers. The first four registers (segments) will point to BS1, thereby indicating word accesses. All of the other registers correspond to segments with only single-byte accesses.

Table 6 shows the values for all Memory Mapping Registers.

Table 6. *Memory Mapping Register Settings*

CREG	Segment	Register Values	Access Location
0xD5	0x15000	110000	Third bank
0xD4	0x14000	110000	Third bank
0xD3	0x13000	110000	Third bank
0xD2	0x12000	110010	Third bank
0xD1	0x11000	110000	Third bank
0xD0	0x10000	110000	Third bank
0xCF	0xF000	111000	Third bank (shadow RAM)
0xCE	0xE000	000000	XT bus
0xCD	0xD000	000000	EMS (bank switched)
0xCC	0xC000	000000	PCMCIA (bank switched)
0xCB	0xB000	000000	XT bus
0xCA	0xA000	110000	Third bank
0xC9	0x9000	100000	Second bank
0xC8	0x8000	100000	Second bank
0xC7	0x7000	100000	Second bank
0xC6	0x6000	100000	Second bank
0xC5	0x5000	100000	Second bank
0xC4	0x4000	100000	Second bank (byte access)
0xC3	0x3000	010000	First bank
0xC2	0x2000	010000	First bank
0xC1	0x1000	010000	First bank
0xC0	0x0000	010000	First bank (word access)

As in the previous example, the graphics RAM cycles are fixed at addresses 0xB8000 to 0xC0000 and override any configuration information to the contrary.

Inversion of bit 18 at segment 0x12000 is required to force that segment into the only open 64kB block of the PSRAM.

The remaining Memory Mapping Registers should be set as shown in Table 7.

Table 7. *Memory Mapping Register Settings*

CREG	Segment	Register Values	Access Location
0xD6-0xDF	0x16000-0x1F000	000000	XT bus
0xE0	0x20000-0x2F000	000000	XT bus
0xF0	0x30000-0x3F000	000000	XT bus

External Glue Logic

Because this example describes a system that contains both single-byte and double-byte accesses in the same physical RAM, some external logic will be required to implement the system.

The need for external logic arises from the fact that when single bytes are accessed in the area between 0x40000 and 0x9FFFF, A0 is substituted for A19 as the high-order address bit. As a result, there is a possibility that two different addresses generated from the F8680 single-chip PC will access the same location in memory.

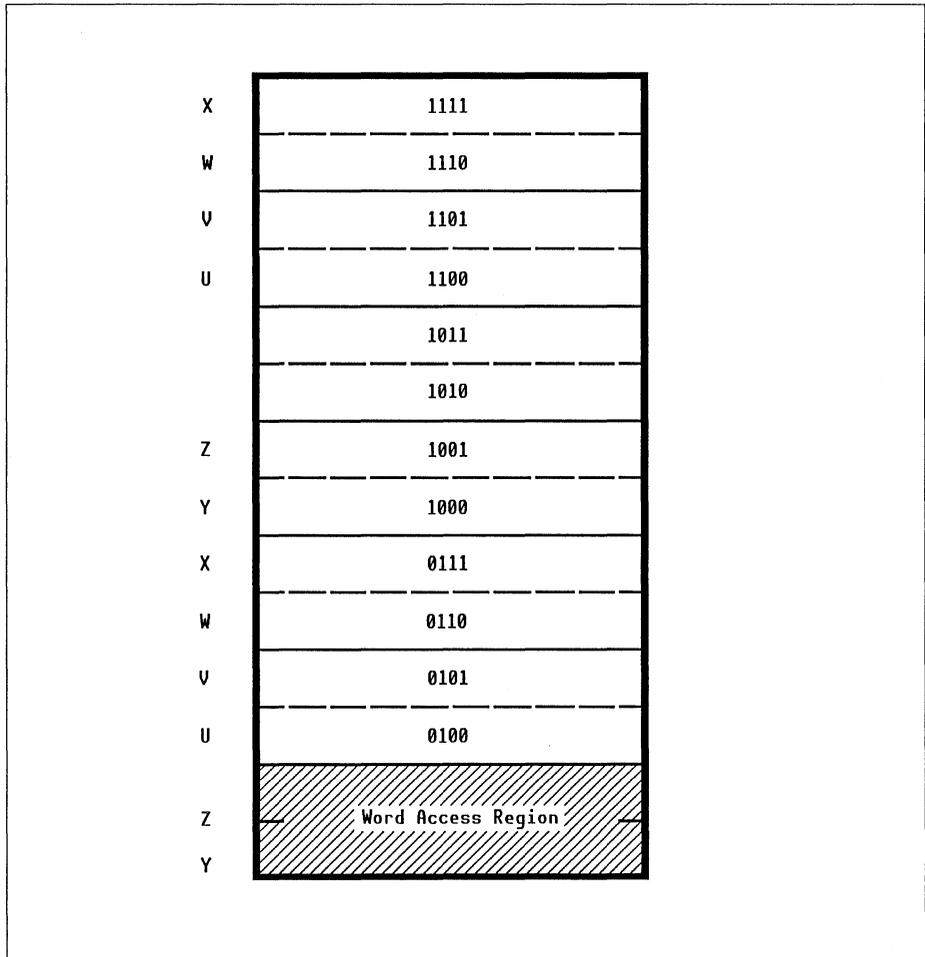
The first 128kB of the 512k x 8 PSRAM is used for word accesses. The other 128kB of the 256kB reserved for word accesses comes from the 128k x 8 PSRAM. Also, the first address generated by the F8680 microchip that will result in a single-byte access is 0x40000. Therefore, it must be ensured that all addresses from 0x40000 to 0x9FFFF do not access the first 128kB of the 512k x 8. Table 8 and its associated Figure 6 illustrate this point.

Table 8. *512k x 8 RAM Usage*

Segment	Address from F8680				Address Seen by RAM				Ref.
	A19	A18	A17	A16	A0	A18	A17	A16	
0x4000	0	1	0	0	0/1	1	0	0	u
0x5000	0	1	0	1	0/1	1	0	1	v
0x6000	0	1	1	0	0/1	1	1	0	w
0x7000	0	1	1	1	0/1	1	1	1	x
0x8000	1	0	0	0	0/1	0	0	0	y
0x9000	1	0	0	1	0/1	0	0	1	z

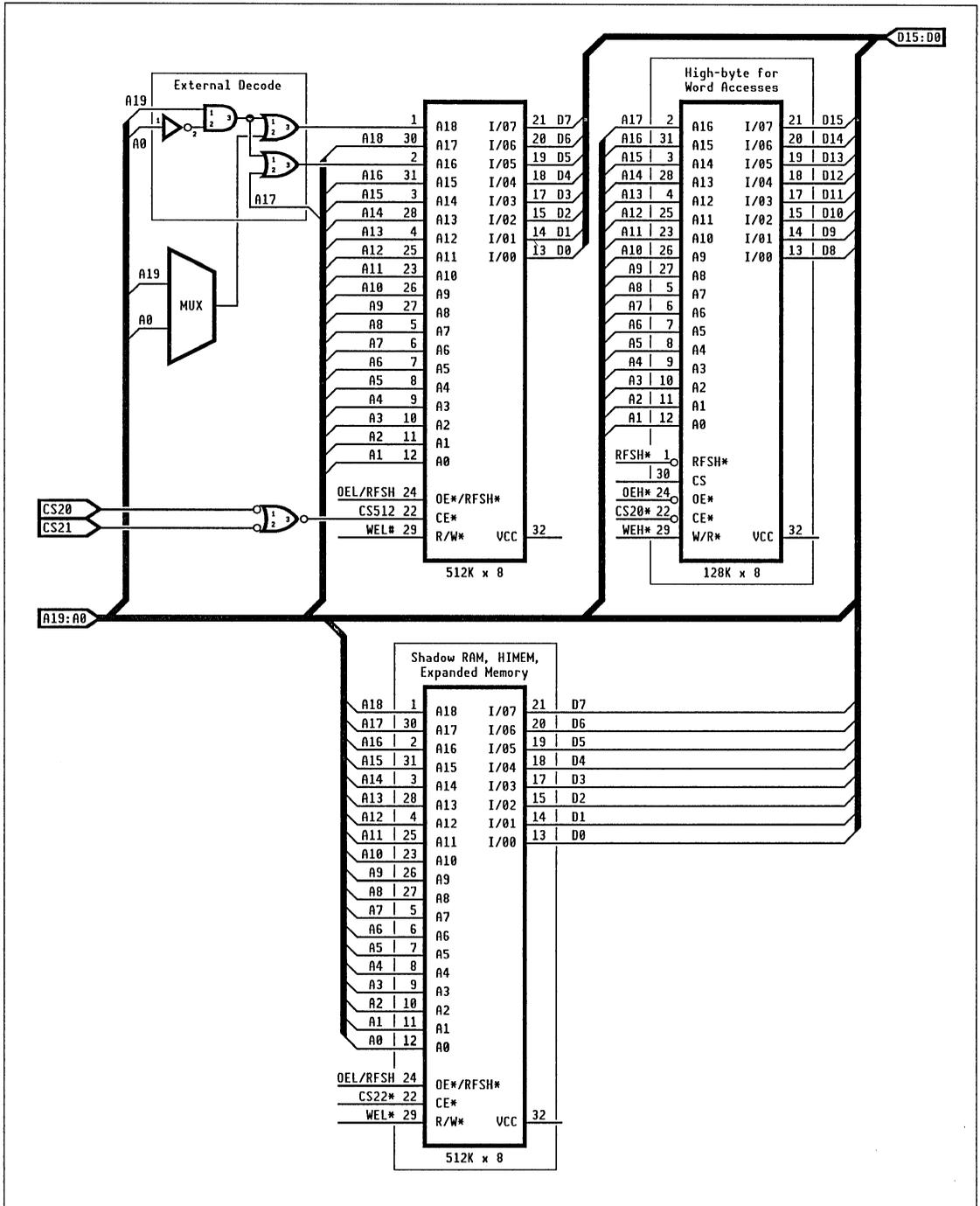
The references in the far right column of the above table are used in Figure 6 to show where each segment is mapped.

Figure 6. 512k x 8 RAM Usage



If this scheme alone were used, when $A0 = 0$ segments 0x8000 (reference x) and 0x9000 (reference y) would access certain locations in the RAM that have already been used because of word accesses. Also, note from Figure 6 that a 64kB block in RAM would remain unused. Therefore, the solution requires use of an external decode, forcing any even-byte access to the 0x8000 or 0x9000 segment to the open block. The schematic shown in Figure 7 implements such a decode.

Figure 7. Memory Interface Schematic



Regional Sales Offices

United States

California, Irvine
Chips and Technologies, Inc.
Phone: 714-852-8721

California, San Jose
Chips and Technologies, Inc.
Phone: 408-437-3300

Georgia, Norcross
Chips and Technologies, Inc.
Phone: 404-662-5098

Illinois, Schaumburg
Chips and Technologies, Inc.
Phone: 708-397-4300

Massachusetts, Andover
Chips and Technologies, Inc.
Phone: 508-688-4600

Texas, Dallas
Chips and Technologies, Inc.
Phone: 214-702-9855

International

Germany
Munich
Chips and Technologies, Inc.
011-49-89-46-3074

Japan
Tokyo
Chips and Technologies, Inc.
011-813-3797-4311

Korea
Seoul
Chips and Technologies, Inc.
011-822-558-5559

Switzerland
Marin
Chip and Technologies, Inc.
011-41-38-33-8490 or
011-41-38-33-8206

Taiwan
Taipei
Chips and Technologies, Inc.
011-886-2-717-5595

Sales Representatives

United States

Arizona, Scottsdale
AzTECH Component Sales
Phone: 602-991-6300

California, Santa Clara
Magna Sales
Phone: 408-727-8753

California, San Diego
S.C. Cubed
Phone: 619-458-5808

California, Thousand Oaks
S.C. Cubed
Phone: 805-496-7307

California, Tustin
S.C. Cubed
Phone: 714-731-9206

Colorado, Wheat Ridge
Wescom Marketing, Inc.
Phone: 303-422-8957

Conneticut, Guilford
DataMark Inc.
Phone: 203-453-0575

Florida, Casselberry
Dyne-A-Mark Corp.
Phone: 407-831-2811

Florida, Ft. Lauderdale
Dyne-A-Mark Corp.
Phone: 305-771-6501

Georgia, Norcross
Bits, Inc.
Phone: 404-446-1155

Illinois, Hoffman Estates
Micro-Tex, Inc.
Phone: 312-382-3001

Indiana, Carmel
Giesting & Associates
Phone: 317-844-5222

Kentucky, Versailles
Giesting & Associates
Phone: 606-873-2330

Maryland, Annapolis
EES
Phone: 301-269-4234

Massachusetts, Woburn
Mill-Bern Assoc.
Phone: 617-932-3311

Michigan, Coloma
Giesting & Associates
Phone: 616-468-3308

Michigan, Comstock Park
Giesting & Associates
Phone: 616-784-9437

Michigan, Livonia
Giesting & Associates
Phone: 313-478-8106

Minnesota, Eden Prairie
High Tech Sales Assoc.
Phone: 612-944-7274

Missouri, Bridgeton
Centech
Phone: 314-291-4230

Missouri, Raytown
Centech
Phone: 816-358-8100

New Jersey, Moorestown
T.A.I.
Phone: 609-778-5353

New York, Commack
ERA, Inc.
Phone: 516-543-0510

New York, Syracuse
Pitronics
Phone: 315-455-7346

New York, Williamsville
Pitronics
Phone: 716-689-2378

Ohio, Cincinnati
Giesting & Associates
Phone: 513-385-1105

Ohio, Cleveland
Giesting & Associates
Phone: 216-261-9705

Oregon, Beaverton
L-Squared Limited
Phone: 503-629-8555

Pennsylvania, Pittsburg
Giesting & Associates
Phone: 412-828-3553

Texas, Austin
OM-Austin
Phone: 512-794-9971

Texas, Houston
OM Assoc., Inc.
Phone: 713-789-4426

Texas, Richardson
OM Assoc., Inc.
Phone: 214-690-6746

Utah, Salt Lake City
Wescom Marketing, Inc.
Phone: 801-269-0419

Washington, Kirkland
L-Squared Limited
Phone: 206-827-8555

Wisconsin, Waukesha
Micro-Tex, Inc.
Phone: 414-542-5352

Canada

Ontario, Kanata
Electro Source, Inc.
Phone: 613-592-3214

Ontario, Rexdale
Electro Source, Inc.
Phone: 416-675-4490

Quebec, Pointe Claire
Electro Source, Inc.
Phone: 514-630-7846

British Columbia, Vancouver
Electro-Source
Phone: 604-948-2314

Distributors

North America

United States

Alabama, Huntsville
Reptron Electronics, Inc.
Phone: 205-722-9500

Arizona, Tempe
Anthem
Phone: 602-966-6600

California, Chatsworth
Anthem
Phone: 818-700-1000

California, Fountain Valley
Bell Microproducts
Phone: 714-963-0667

California, Irvine
Anthem
Phone: 714-768-4444

California, Milpitas
Bell Microproducts
Phone: 408-434-1150

California, Rocklin
Anthem
Phone: 916-624-9744

California, San Diego
Anthem
Phone: 619-453-9005

California, San Jose
Anthem
Phone: 408-453-1200

Colorado, Englewood
Anthem
Phone: 303-790-4500

Colorado, Wheat Ridge
Added Value
Phone: 303-422-1701

Connecticut, Waterbury
Anthem
Phone: 203-575-1575

Florida, Ft. Lauderdale
Reptron Electronics, Inc.
Phone: 305-735-1112

Florida, Tampa
Reptron Electronics
Phone: 813-854-2351

Georgia, Norcross
Reptron Electronics, Inc.
Phone: 404-446-1300

JACO/QC
Phone: 404-449-9508

Illinois, Schaumburg
Anthem
Phone: 708-884-0200

Reptron Electronics, Inc.
Phone: 708-882-1700

Massachusetts, Wilmington
Anthem
Phone: 508-657-5170

Bell Microproducts
Phone: 508-658-0222

Maryland, Columbia
Anthem
Phone: 301-995-6640

JACO/QC
Phone: 301-995-6620

Michigan, Livonia
Reptron Electronics, Inc.
Phone: 313-525-2700

Minnesota, Eden Prairie
Anthem
Phone: 612-944-5454

Minnesota, Minnetonka
Reptron Electronics, Inc.
Phone: 612-938-0000

New Jersey, Pinebrook
Anthem
Phone: 201-227-7960

New York, Commack
Anthem
Phone: 516-864-6600

New York, Long Island
JACO/QC
Phone: 516-273-5500

North Carolina, Raleigh
Reptron Electronics, Inc.
Phone: 919-870-5189

JACO/QC
Phone: 919-876-7767

Ohio, Columbus
EMC
Phone: 614-299-4161

Ohio, Solon
Reptron Electronics, Inc.
Phone: 216-349-1415

Ohio, Worthington
Reptron Electronics, Inc.
Phone: 614-436-6675

Oregon, Beaverton
Anthem
Phone: 503-643-1114

Pennsylvania, Horsham
Anthem
Phone: 215-443-5150

Texas, Addison
JACO/QC
Phone: 214-733-4300

Texas, Austin
JACO/QC
Phone: 512-835-0220

Texas, Richardson
Anthem
Phone: 214-238-7100

Texas, Sugarland
JACO/QC
Phone: 713-240-2255

Utah, Salt Lake City
Anthem
Phone: 801-973-8555

Utah, West Valley
Added Value
Phone: 801-975-9500

Washington, Bothell
Anthem
Phone: 206-483-1700

Canada

Ontario, Rexdale
Valtrie Marketing
Phone: 416-798-2555

Distributors

International

Asia/Pacific

Australia

Melbourne
Zatek Components, Ltd.
3-763-8899

Sidney

Zatek Components, Ltd.
2-895-5534

Hong Kong

Kwung Tong
Wong's KK Ltd.
85-2-345-0121

India

Bombay
Silicon Electronics
91-22-243-460

New Dehli

Ajay Jain
91-11-686-3044

Israel

Tel-Aviv
CVS
97-23-544-7475

Japan

Kawasaki
CTC Components Systems Co., Ltd.
81-44-852-1825

Tokyo

ASCII Corporation
81-3-486-9188

Korea

Seoul
Nae Wae Semiconductor
82-2-704-0103

Malaysia

Penang
Dynamar
60-4-637-292

Selangor

03-776-7140/199

Singapore

Singapore
Technology Distribution
PTE Ltd.
65-368-6065

Taiwan

Taipei
Ally, Inc.
886-2-562-0828

World Peace

Industrial Co., Ltd.
88-62-786-5311

Thailand

Bangkok
Grawinner
66-2-215-8742

Europe

Belgium

Zaventem
ACAL Auriema Belguim
32-2-720-5983

Denmark

Herlev
Nordisk, Elektronik A/S
454-284-2000

Finland

Helsinki
OY Fintronic AB
35-806-926022

France

Le Chesnay
A2M
33-1-395-49113

Germany

Neteltal
Rein Elektronik GmbH
49-215-373300

Italy

Milano
Moxel S.r.L.
39-2-612-90521

Netherlands

Eindhoven
ACAL Auriema Nederland B.V.
31-40-816-565

Norway

Hvalstad
Nordisk Elektronik A/S
47-284-6210

Spain

Madrid
Compania Electronica de Tecnicas
Aplicadas, S.A.
34-1-754-3001

Barcelona

Compania Electronica de Tecnicas
Aplicadas, S.A.
34-3-300-7712

Sweden

Kista
Nordisk Elektronik A.B.
46-8-703-4630

Switzerland

Dietikon
DataComp AG
41-1-740-5140

United Kingdom

Berkshire, Maidenhead
Magna Technology
44-734-880211

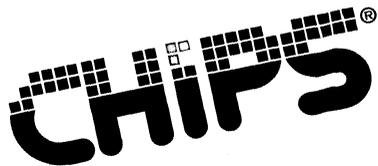
Oxfordshire

Thames Components
8-44-261188

Americas

Mexico

Gaudalajara
Sonika
52-364-74250



Chips and Technologies

3050 Zanker Road

San Jose, California 95134

Phone: 408/434-0600

Telex: 272929 CHIPS UR

FAX: 408/434-6452

Publication No.: AN57

Stock No.: 02057-001

Revision No.: 1.0