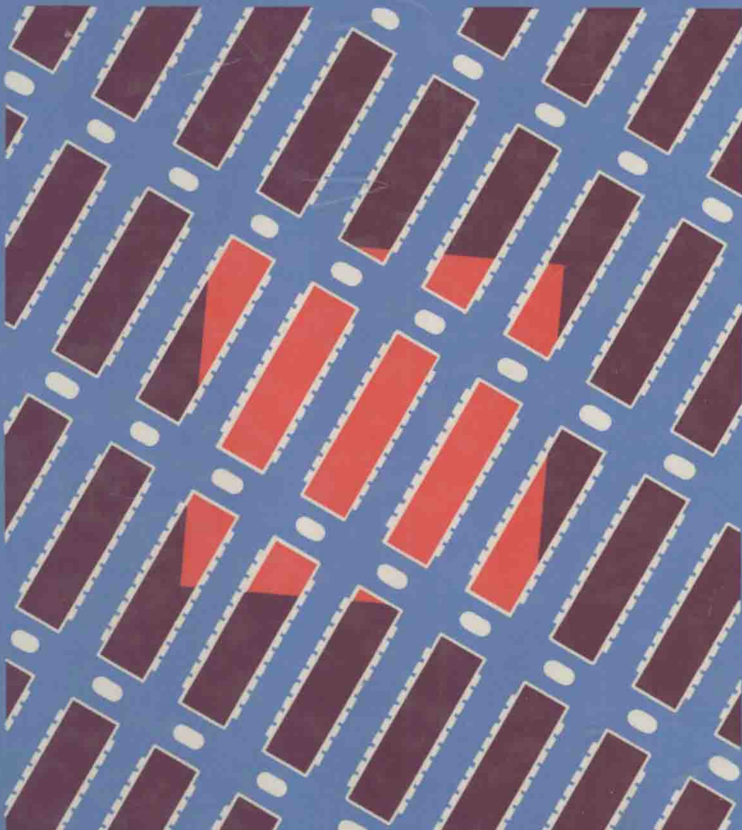
Microprocessor Interfacing and the 68000

PERIPHERALS AND SYSTEMS



Alan Clements

Microprocessor Interfacing and the 68000 Peripherals and Systems

Alan Clements
Teesside Polytechnic,
Middlesbrough, UK

John Wiley & Sons
Chichester · New York · Brisbane · Toronto · Singapore

Library of Congress Cataloging-in-Publication Data:

Clements, Alan, 1948–
Microprocessor interfacing and the 68000 : peripherals and systems / Alan Clements.
p. cm.
Bibliography: p.
ISBN 0 471 91575 0
1. Motorola 68000 (Microprocessor) 2. Computer interfaces.
I. Title.
QA76.8.M6895C53 1989
621.398'1—dc19 88-14202
CIP

British Library Cataloguing in Publication Data:

Clements, Alan, 1948–
Microprocessor interfacing and the 68000: peripherals and systems.
1. Motorola 68000 microprocessor systems
I. Title
004.165

ISBN 0 471 91575 0

Typeset by Wyvern Typesetting Ltd, Bristol
Printed and bound in Great Britain by the Anchor Press Ltd, Tiptree, Essex

Advanced Microprocessor Interfacing and the 68000 Peripherals and Systems

For
J. K. Darby

Preface

A future historian might view the growth of microcomputer technology since the mid 1970s as a series of waves of progress. The first four-bit microprocessors, the Intel 4004 and 4040, were rapidly followed by the first eight-bit microprocessors. A single microprocessor was able to replace tens of SSI and MSI TTL packages in an intelligent controller. These microprocessors also made possible the low-cost, single-board, general-purpose microcomputers.

The second wave of innovation brought special-purpose peripherals (strictly speaking we should use the term *interfaces* rather than *peripherals*), allowing these microprocessors to communicate with the outside world. Included here are the 6850 serial interface and the 6820 parallel interface. Just as the microprocessor made it possible to design low-cost microcomputers, the peripheral made it possible to connect the microcomputers to external systems cheaply and efficiently. Clearly, it does not make economic sense to build low-cost microcomputers with a handful of chips if microcomputers require large numbers of devices to interface with, say, CRT displays and printers.

In the late 1970s, microcomputers had tiny memories and 1024-bit memory chips were the state of the art. With the third wave came the high-density, low-cost memory component, and just after the mid 1980s 1 Mbyte memory chips had become commonplace. These 1 Mbyte chips represented a 1024-fold increase in density in half a decade. The availability of low-cost memory made the personal computer (as opposed to the single-board computer) really possible.

The fourth wave of progress has seen the introduction of more sophisticated microprocessor architectures, from the 8086 to the 68030, the 32000 and the RISC machines. Modern microprocessors are distinguished by their powerful instruction sets, their wealth of addressing modes and their ability to access very large memory spaces. Taken together, the developments in microprocessor and memory technology have led to powerful minicomputers that can be bought with large random access memories, a disk drive, CRT display and a printer for less than the cost of a two-week holiday in the sun or a translatlantic return flight.

Advances in microprocessor architecture and memory technology have not, of course, taken place strictly one after another – there has been a considerable degree of overlap with each advance forcing new developments in related areas. Increasing processor power leads to demands for larger memories to exploit this new power. In turn, larger memories call for yet more processing power to take advantage of the new generation of large programs.

Changes in microcomputer technology have initiated a corresponding development in the books written about microprocessor systems. Early books described basic microprocessor technology and were written for a relatively naïve audience. Later books were able to start from a higher level of assumed knowledge, as computer science gradually filtered down through the education system.

Microprocessor Interfacing and the 68000 has been written to fill a gap in the spectrum of books currently available. It does not attempt to teach microprocessor technology or programming and assumes that the reader is already well versed in these topics, or at least has access to information about them. This book is about the components that are needed to interface a microprocessor system to the outside world. These components range from parallel and serial interfaces to disk controllers and real-time clocks. Many books on microprocessors have devoted so much space to the microprocessor itself that they have had relatively little to say about interface chips. Some of the peripheral chips now available are very complex and a few are considerably more complex than the microprocessor itself. Consequently, there is a danger that students may leave university with a thorough grounding in certain aspects of microcomputer design while having little more than a vague and unsatisfactory knowledge of, for example, microprocessor support chips.

One of the most difficult tasks facing the newly qualified electronic engineer or computer technologist is bridging the gap between the theory learned in university and the practical experience required by an employer. Nowhere is this more evident than in interfacing. Students are taught the basic principles of microprocessor systems in their digital design courses, but are later confronted by highly complex data sheets written by manufacturers using their own in-house jargon. *Microprocessor Interfacing and the 68000* provides designers with a stepping stone between the general course on microprocessor systems design and the requirements of the real world.

The treatment of microprocessor interfacing

The author of a book on microprocessor interfacing must steer a course between the extremes of a vague and 'woolly' text that glosses over all the practical details of peripheral chips and a book which merely repeats the manufacturer's data sheet in all its gory detail. I have selected a middle course and have attempted to show how peripherals are employed and the difficulties encountered in using them. Once readers understand how a generic interface works, they are then in a position to deal with the data sheets of the new generation of interface chips.

In choosing suitable interfaces to describe, I have avoided compiling a list of available chips and have not attempted to compare and contrast one manufacturer's device with another's. *Microprocessor Interfacing and the 68000* introduces a range of interface chips varying from the mundane parallel port to the more esoteric multiprocessor interface.

I have written this book for students taking second-level courses in computer technology or electronic engineering with a computer science bias, and for engineers in industry who have to design microcomputers using these new components.

Contents

Chapter 1 describes the design of a simple microcomputer based on the 68000. Although I do not wish to write yet another microprocessor textbook, it is important to consider the operation of a microprocessor from the point of view of the peripherals that are connected to it. The first part of this chapter concentrates on the 68000's buses and shows how they are used to transfer information between the microprocessor and a peripheral. The second part of Chapter 1 presents the design of a typical 68000 system which can be used in conjunction with the interface components described in this book.

Chapter 2 introduces the peripheral interface chip that allows a microprocessor to be connected to an external system with a minimum of hardware overhead. The chapter begins with an overview of *the generic interface* and discusses some of the ways in which they are connected to a host microprocessor. Topics introduced in Chapter 2 include the way in which registers in a memory-mapped peripheral are accessed by the CPU and the timing requirements of peripherals.

Chapter 3 is devoted to the DUART, dual universal asynchronous receiver/transmitter. The DUART, or a similar peripheral interface component, is found in almost every general-purpose digital computer with some form of external interface and permits a microprocessor to be interfaced easily, flexibly and efficiently to an asynchronous serial data link.

Chapter 4 looks at the real-time clock (RTC). A real-time clock performs two functions: it generates the stream of interrupt requests required by a multitasking processor and maintains a copy of the time of day. The RTC is an unusual peripheral, because it does not link a microprocessor to an external device. In at least one sense, the RTC is *an end in itself*.

Chapter 5 shows how a microcomputer communicates with the analog world. Techniques of converting a voltage into a digital value are described. The inverse process of converting the digital output of a computer into a voltage whose value is a function of the binary value is also considered. Practical examples of analog to digital and digital to analog converters are given.

Chapter 6 looks at one of the most important interfaces found in the general-purpose digital computer, the CRT controller that performs almost all the functions required to map digitally encoded data stored in memory onto the screen of a CRT terminal (i.e. a television). Before the advent of the CRT controller, between 20 and 100 MSI and SSI chips were required to perform this function. An example of a display system suitable for use in a microcomputer is described.

Chapter 7 is concerned with the interface between a microprocessor system and a floppy disk drive. The principles of the operation of a floppy disk drive, together with the way in which data is encoded before recording, are discussed, and an example of a typical floppy disk controller (FDC) chip is given. This section is completed by showing how an FDC is interfaced to a 68000, and a brief introduction to the software needed to drive the FDC is provided.

Chapter 8 applies the maxim 'You can't have too much of a good thing' to microprocessor systems and shows how several microprocessors can be coupled to increase the power or computational throughput of a system. We examine two aspects of multiprocessor systems here: their topology and the way in which they communicate with each other. Chapter 8 provides us with an opportunity to examine the interface between a 68000 system and the VMEbus which is used in many 68000-based multiprocessors.

Acknowledgements

I would like to thank all those who helped me to write this book. In particular, I would like to thank the semiconductor manufacturers who provided me with the source information that enabled me to write about the peripherals I have described. My special thanks goes to Motorola, Philips/Signetics, GE Solid State, Analog Devices, Hitachi, and Western Digital.

In attempting to interpret manufacturers' data sheets and application reports, it is inevitable that I have made both errors of comprehension and mechanical errors. All such errors are my own responsibility.

I would also like to thank those at John Wiley engaged in the commissioning and production of my book.

Finally I would like to thank Paul Lambert in the Computer Centre at Teesside Polytechnic who checked the pseudocode programs in Chapters 6 and 7.

Any readers who wish to 'close the loop' and comment on my book or who find errors that should be removed in later editions can contact me via my publisher, or direct (School of Information Engineering, Teesside Polytechnic, Middlesbrough TS1 3BA, UK), or via E-mail (JANET address ACT012@UK.AC.TP.PA).

Contents

3 *Serial Input/Output* 113

4 *The Real-Time Clock* 167

5 *Analog Input/Output* 204

Chapter 1

The Microcomputer

Before we can even begin to look at the characteristics of microprocessor peripherals and their interface to microprocessors, it is necessary to examine the nature of the microprocessor to which they are interfaced. In this book we are going to concentrate on the interface between peripherals and the popular 16/32-bit 68000 microprocessor. Chapter 1 introduces the 68000 microprocessor from the point of view of its external interface through its 64 pins. We show how the 68000 is connected to memory components and to peripherals, and provide an example of a microcomputer built around a 68000.

The old Roman god Janus is always depicted as facing in two opposite directions at the same time. Interfaces share the same ability to look in two directions at once. An interface looks one way to the computer that it is interfacing and it looks in the opposite direction to the external system that it is controlling. Microcomputer designers must consider both these *directions* separately. Engineers generally refer to the *computer side* and the *peripheral side* of an interface.

As most interfaces are connected to a computer, often called *the host computer*, it is important to examine the relationship between the interface and the computer. A practical difficulty arises here because not all computers have the same characteristics or behave in the same way as far as the interface is concerned. We are therefore forced to adopt either a general approach to interfacing or a more specific approach centred on one particular computer. I have followed the latter path and have decided to adopt the 68000 microprocessor as a vehicle to teach microcomputer interfacing. Fortunately, once the way in which an interface is connected to a 68000 is understood, it is easy to design a connection between the interface and almost any other microprocessor.

1.1 The 68000

If asked why I chose the 68000 to illustrate microcomputer interfacing, I could give the mountaineer's answer – 'Because it's there'. Such an answer would be a reaction to the spate of articles I have seen in the technical press in recent years. Each article is written by a 'high priest' of one of the major faiths: the 68000, the 8086 and the 32000.

Such articles really do belong to the realms of theology. They take a minor attribute of one microprocessor and show that the microprocessor is much better than its rivals. In this way, any microprocessor can always be proven better than any other. Articles like these are most unhelpful to students or to newly qualified engineers who have not built up a personal fund of knowledge and expertise to see through the articles and to realize that one microprocessor may represent an optimum choice for a word processor, while an entirely different device may be optimum for a number cruncher.

I have chosen the 68000 for this book because it is an excellent device for teaching purposes. The reasons for selecting a 68000 can be summarized as follows:

1. The 68000 has a powerful instruction set which is relatively easy to learn. Students new to the 68000 are able to absorb its mnemonics quickly, while more advanced students can make good use of its variable-size operations and powerful addressing modes.
2. The 68000 has much to offer from an interfacing point of view. The three most important features of the 68000's interface are its non-multiplexed asynchronous address and data buses, its prioritized vectored interrupt structure and its support for multiprocessing and DMA through its bus arbitration control pins.
3. The 68000 supports real-time programming and multiprocessing environments. At any instant, the 68000 is in one of two states: user mode or supervisor (or system) mode. As far as the execution of the majority of instructions is concerned, there is no difference between these modes. However, certain instructions can be executed only when the 68000 is in the supervisor mode. By dedicating the system mode to the operating system, it is possible to preserve system integrity and prevent user tasks from playing havoc with the system. For example, one user can be prevented from accessing the memory space assigned to another user. Similarly, it is possible to deny user programs direct access to I/O devices (i.e. the interfaces). Consequently, a user program must perform I/O operations by means of requests to the operating system which runs under the supervisor mode. Note that external logic to the 68000 is necessary to implement these functions.

   The 68000's sophisticated interrupt handling (exception processing) facilities also contribute to this end, because they can be used to take all input/output transactions out of the hands of user tasks and make certain that only the operating system can control input/output operations.

As I indicated in the introduction, this book is not primarily about microprocessors themselves. Therefore, fine details concerning the 68000 are omitted here. Readers not already familiar with the 68000 will find suitable reading matter listed in the Bibliography. What does interest us is the way in which 68000 communicates with peripherals.

1.2 The 68000 data and address bus

Broadly speaking, the physical interface of any microprocessor (i.e. its pins) can be divided into two parts: a communication path between the microprocessor and its external memory, and a control structure that determines the sequence of operations carried out by the microprocessor. The *communication path* is composed of the microprocessor's address bus, data bus and data-flow control bus. Similarly, the *control structure* includes the microprocessor's reset, interrupt and bus arbitration control facilities. We discuss first the 68000's data and address bus which we call the DAT (data and address transfer).

The structure of the 68000's address and data bus is illustrated in Figure 1.1, and Figure 1.2 shows how it is used to effect two-way communication between the microprocessor and its memory or peripherals.

The 68000 DAT is almost entirely conventional and presents only two tiny surprises to the system designer. First, unlike some other 16-bit microprocessors, the DAT is non-multiplexed and therefore has entirely separate address and data paths. Separate address and data buses simplifies microprocessor to memory interfacing and, some would say, enhances processor throughput while increasing system cost due to the large number of signal-carrying paths. The DAT is made up of three elements: a 23-bit address bus, A_{01}–A_{23}, which allows $2^{23} = 8$ Mwords to be uniquely addressed, a 16-bit data bus, D_{00}–D_{15}, and five data-flow control lines: R/W*, AS*, UDS*, LDS* and DTACK*. R/W* (read/write*) is high during a read cycle or when the 68000 is inactive, and is low during a write cycle. AS*, LDS*, UDS* are the address, lower data and upper data strobes respectively. It is often convenient to write DS* rather than UDS* or LDS* when we wish to describe one or both the 68000's data strobes and are not particularly interested in whether the strobe is UDS* or LDS*.

Although the 68000 has 32-bit address registers and a 32-bit program counter, only 24 bits of an address are available off-chip (i.e. A_{01}–A_{23} and UDS*/LDS* which serves as A_{00} for byte accesses). Consequently, it is common practice to express 68000 addresses by *six* hexadecimal characters rather than