Hardware & Construction Manual

Revision C
Version 1.095
August, 1993

Applix 1616 microcomputer project
Applix pty ltd

1616 Hardware
& Construction Manual

The original version of this manual was written by Andrew Morton of Applix
Additional introductory and tutorial material by Eric Lindsay
Editorial and design consultant: Jean Hollis Weber

Welcome to the growing family of 1616 users. Don't forget to keep in touch with Applix, as we are very interested in getting to know you, and what you are doing with your 1616. Apart from our NSW and Victorian User Groups, the main way to get to know other 1616 users (and for them to know you) is through µPeripheral - the 1616 newsletter. So please join in and enjoy!

Comments about this manual or the hardware and software it describes should be sent to:

Applix Pty Limited
Lot 1, Kent Street,
Yerrinbool, 2575
NSW Australia
(048) 839 372

Private Applix BBS include Colin McCormack (02) 543 8213, SSM (02) 554 3114, Trantor (02) 718 6996 and PPT (02) 544 1060 (2400 baud or less, 8 bit, no parity) (some numbers may be ringback - let phone ring twice, then phone back within 30 seconds).

ISBN 0 947341 03 X

1
Introduction

There are an enormous number of cheap business and home computers available.

If you just want to play games with a computer, buy a Commodore 64 or one of its numerous relatives. They are great fun. But don't pretend you are doing it to 'learn how computers work'. You can learn to drive a car by getting a learner's permit and going to a driving school; but to become a motor mechanic you need to get under the hood and get dirt under your fingernails!

You don't learn much about computers by playing games, or by using other peoples' programs; the only way you really learn is by writing your own.

If you have a specific business problem to solve, and it can be done by a word processor, database, or spreadsheet, buy an IBM PC clone. If you read the right manuals, and apply yourself, they can do almost any small (and most large!) business jobs. Better yet, find someone else who has solved your problem, and buy whatever they bought!

However, an increasing number of people realise that, like it or not, there will be more computers in our future. They buy a computer to **learn about computing**. Either for themselves, or to help their children. In most cases, within a year, the computer is languishing untouched in the back of a cupboard, or is being used only to play games.

It is remarkably difficult to come to grips with how computers work, when what you have is a mysterious blue box, and little idea of what is going on within it. Some people in the computer industry appear to like to retain an air of mystery. Perhaps it lets them sell their services at a higher price. Maybe they just want to be the technological equivalents of high priests or shaman.

The Applix 1616 offers you the chance to **know** exactly what is inside it, because **you** can put it together. If a component fails, **you** will **know** exactly what that component does, how to test it, and how to replace it. You don't have to rely upon a computer store, with salespeople who know less than you. Components in the 1616 are standard, and can be obtained at good electronic hobbyist shops. You can have confidence in the quality of your 1616, because you decide on the quality of the construction!

When you have assembled your 1616, you will find your learning will accelerate. The 1616 includes a built in software monitor that helps you learn exactly how computers work. Monitors were formerly included in every home computer, including the famous original Apple which introduced so many people to computer programming. Over the past decade, they have been left out of new computers.

The 1616 also includes numerous devices that allow it to sense and control the outside world. These include analogue to digital, and digital to analogue converters, plus digital control lines, joystick connector, and two watts of stereo sound. It can sense the real world, and control other devices. The 1616 also has all the interfaces of a business computer, with two built in serial ports, for communicating with other computers, a parallel printer port, colour or monochrome video, and four expansion card slots. No other computer known to us has as many, nor as wide a range of inbuilt interfaces.

However the major reason for buying a 1616 is that it is designed to lead you into learning how to program, and program at every level, from the simplest, to the most complex.

Enthusiasts such as myself tend to ignore the dark side of computers, their social effects. As a technophile, I love gadgets of all sorts. My life has been immensely enriched by the fun I've had playing with computers. I owe my present (very enjoyable) job to skills I picked up as a computer hobbyist.

However, the most complex task most average, non-technical people ever learn is driving a car, and a car is also the most complex gadget they normally encounter. A computer system is many times more involved, and more involving. If you are serious about learning about computers, you will be devoting a substantial portion of the next few years of your life to learning a wide variety of esoteric topics. Your social life, your family, your business, your other hobbies, and your finances will all come under greater strain and time restraints. You should be aware that getting involved with computers produces problems outside the strictly technical.

Hobby computing is almost exclusively a male pursuit. Despite the first programmer being Ada Lovelace, COBOL being predominently devised by Grace Hopper, and the many professional programmers who are women, the people who build home computers, and the people who join user groups, are almost all male. This may merely indicate that most women are too sensible to become so involved in a hobby that demands obsessive behaviour for mastery.

Book your start - background reading

If building the Applix 1616 as a kit, we assume that you have had some exposure to building electronic kits, and that you have some knowledge of electronics. If you have no knowledge of electronics, see chapter 5, on Kit Building, for details of other methods of building or buying a 1616.

If you wish to understand something of computer electronics, we recommend a number of cheap books. These are produced by Texas Instruments (inventors of the integrated circuit), and were formerly sold by most Tandy stores at under $10. They are slightly dated, however the essential background is all there. They do not assume a background in either electronics or mathematics, and generally ignore any mathematical treatment of their topics. They probably aren't the very best on their topics, but they are remarkably good, and very cheap, if you can still find copies.

Understanding Solid-State Electronics (Tandy #62-2035) takes you from partly remembered high school physics through basic electronic ideas, semiconductors, diodes, transistors, and digital and linear integrated circuits. This will act as a general introduction to electronics.

Understanding Solid-State Electronics Vol II (Tandy #62-1397) expands the coverage of logic circuits, microprocessors, and a number of more specialised circuits including graphics processors, and interface circuits. This tells you far more about electronics as it applies to the digital circuits that make up a modern computer.

Understanding Digital Electronics (Tandy #62-2010) explains how the digital circuits in a calculator (a form of computer) work. This is more detailed treatment of the internal working than given in the previous books. because the calculator is relatively simple, and familiar, you will learn how digital electronics can be used to make much more complicated circuits.

If you have never had a computer before, and never programmed anything, you have some study to do on computing concepts before really coming to grips with the 1616.

Understanding Microprocessors (Tandy #62-2017) briefly covers digital electronics, computing systems, instruction sets and programming concepts, before providing two detailed examples with complete solutions. This book will be helpful before starting to learn assembly language.

Understanding Computer Science (Tandy #62-1383) briefly covers computer architecture and hardware, then languages, operating systems, managing computing resources, how language translators work, and systems analysis. It provides a helpful introduction to the concepts behind both language programming, and the 1616/OS operating system used by the Applix 1616.

Understanding Computer Science Vol II (Tandy #62-1395) extends its cover to parallel and serial communication, distributed processing and networks, while the programming examples cover using computers to simulate or model the real world. It includes an introduction to computer graphics.

If the list of topics above just sound like buzzwords, drop into your Tandy store, pick up those books, and start reading. By the time you finish, the rest of these manuals will seem a lot more understandable.

If you can find a copy, Don Lancaster's *Microcomputer Cookbook, Vol 1, Fundamentals*, is great fun to read. It is slightly dated, and mostly deals with older designs, but the fundamentals do not alter much from computer to computer.

At about this time, some of you will be muttering "I'm not reading 1500 pages of introduction!" Fine. Various people have managed without. However, unless you are very smart, you will take about two years to pick up the information contained in these manuals, and the books listed above. Your choice.

This manual

This manual is mainly intended for Applix 1616 purchasers who intend to build their computer from either the kit, or by buying a bare board and supplying their own components. It is also the repair manual for the Applix 1616.

It also contains material of interest to electronics enthusiasts who wish to know something more about the components in the Applix 1616, and the general design.

As such, the manual contains two general groups of chapters.

- Design background
- Construction and testing

The first four chapters provide background information on the low level design and electronic circuitry in the Applix 1616.

Chapter 2 briefly introduces the Motorola 68000 processor family. Chapter 3 provides a memory map, and input output map, of the Applix 1616, and explains why these details are less important than in earlier microcomputers. It details the location of all ports, and gives information on the boot sequence, for trouble shooting purposes.

The design of the Applix 1616 is outlined in Chapter 4, often with descriptions of the operation of particular chips. Those intending to make extensive use of this chapter should supplement this manual with data books detailing all the chips used in the 1616. This applies particularly to programmable chips such as the 6522 VIA, 6845 CRTC, and Z8530 SCC. You should also ensure that you have a copy of the circuit diagrams (usually in the back of this manual).

The rest of the manual is devoted to construction and testing of the 1616.

Chapter 5 introduces you to kit building, and suggests alternatives, in case the task appears too difficult. The complete parts list appears in Chapter 6, both as a buying guide, and as part of the construction steps.

The first steps of construction are given in Chapter 7, together with various hints and tips. The more complex integrated circuit components are covered in Chapter 8, together with a description of the full range of built in tests you must conduct to ensure that the board is working correctly.

Various appendices detail the connector pinouts, switches, links and jumpers, and describe how to make appropriate cables.

2
The 68000 Microprocessor

Motorola's M68000 series microprocessor family was first introduced in 1980, at a time when advances in VLSI fabrication and design techniques were permitting manufacturers to put an entire 16 bit microprocessor on a chip.

At this time some manufacturers (most notably Intel) elected to design their 16 bit micros as an upgrade to their 8 bit ones. This eased the problem of converting programs to run on the new designs, at the expense of designs that were not as attractive to programmers. This essentially meant that software for the 8080 (virtually the original microprocessor) was quickly, but often badly, converted to run on Intel's 8086 chips, noteably on the then new IBM Personal Computer. The 8086 chip was designed with 16 bit registers, augmented by 16 bit segment registers to allow a 20 bit address bus. As programmers became more ambitious, this scheme became a problem. Handling large amounts of memory in IBM style computers remains difficult.

Although Intel has now introduced some excellent chips, such as the 80386 and 80486, these still have to retain compatibility with the old 8086. The new features of these chips are rarely used; less than one tenth of one percent of programs treat the new chips as anything other than a faster 8086. This can be very frustrating for low level programmers.

Motorola, on the other hand, produced an architecture which was derivative of none of their previous microprocessors - it was a multi-register device which was designed from scratch to support high-level compiled languages, multi-tasking operating systems, and multiple processors. It was designed from the start to support 32 bit registers, and access to large memories is linear and conceptually easy.

Motorola has excelled in designing microprocessors for the programmer. Their MC6809 was the best of all the 8-bit microprocessors from the programmer's point of view and the MC68000 family instruction set certainly indicates that the software boys made a large contribution to its design. Unlike most earlier 8 bit microprocessors, the 68000 family are microcoded, leading to a relatively orthogonal instruction set, and allowing for bug correction by the manufacturer without breaking working sections of the processor.

About the M68000 family

The microprocessors in the MC68000 family are:

MC68000

A 64-pin IC, with a 16 bit external data bus, 32 bit internal data bus, 16MHz maximum available clock speed. Used in the Apple Macintosh, Atari ST and Commodore Amiga, until 1992, when most manufacturers uprated their systems to the MC68030. Described in detail later, as it is the microprocessor used in the Applix 1616.

MC68008

With 48 pins, an 8 bit external data bus instead of 16 bits, and a 12.5MHz maximum available clock speed, the MC68008 was designed to bridge the gap between existing 8 bit subsystems (such as memory I/O cards) and a 32 bit microprocessor. Used in the defunct Sinclair QL. The 8 bit data bus means most memory operations take twice as long as with a 68000. The address bus uses only 20 bits, for a maximum addressing range of 1 megabyte. The extra four bits available on the 68000 are truncated. Interrupt priority codes 0, 2, 5 and 7 are recognised by the IPL0 and IPL2 lines. It lacks the extra IPL1 line of the 68000. The instruction set is identical.

MC68010

This device is very similar to the MC68000. It has some extra instructions, and some of its instructions have been made more efficient. The main rationale for the introduction of the MC68010 is probably as a 'fix' for a limitation of the MC68000 which prevented it from resuming a program after a memory fault - a capability which is necessary for large operating systems. This MC68010 feature is not really needed in a system of the 1616's size. Despite this, the 1616/OS will detect, and use, an MC68010 chip, if fitted. Experimenters should note this is not the case with the Atari ST and Apple Macintosh.

Significant differences between the 68010 and 68000 include additional instructions MOVEC, MOVES and MOVE from CCR, while MOVE from SR is now privileged. A Return and Deallocate instruction is available, which pops a fixed number of parameters off the stack (but C or assembler programmers are unlikely to use it). These all assist in providing a virtual memory system.

Some instructions are faster. These include the multiply and divide. Loop sequences consisting of a single word instruction, and a DBcc with a displacement of -4, can now execute without an instruction refetch. This has obvious speed advantages for block memory moves (software blitter, anyone?)

Compatibility problems occur due to changes in the exception stack. These are mostly due to the 68010 having three additional registers: the Vector Base register, and the Source and Destination Function Code registers. A short stack (not a pancake) is identical to the 68000 unless the exception is a 'buserr' or an 'address error'. An additional word, including the vector base register displacement, is placed at the bottom of the stack. Since the 68000 places only three words on the stack, this extra fourth word will cause problems in an exception handler written for the 68000. The long form stack is totally different to the 68000, and includes extra address and bus error information. We don't recommend changing over to it.

MC68020

The MC68020 is a 114 pin microprocessor IC with 32 address pins, 32 data pins, on-board 256 byte instruction cache, 25MHz clock speed, dynamic bus sizing, etc, etc. This very powerful microprocessor is often used in multi-user Unix systems, low end graphics workstations, and expensive business commputers such as the Macintosh II. It offers 6 additional address register indirect or PC relative addressing modes, extra control modes, 64 bit divides and 32 bit multiplies, and extra coprocessor support.

MC68030

This device is being used in high end workstations such as the Macintosh IIx, SE/30, NeXT, and Unix mini computers, and in all Macintosh computers from 1993 on. It adds a data cache to the 68020, and has on-board memory management facilities, plus other features to reduce the external support logic normally required. Until recently, the price has been too high for most home systems. Now that it is being used extensively by large manufacturers, the price is expected to drop rapidly.

MC68040

The latest in the Motorola family. Much faster than previous versions, with more features built in.

A hardware description of the Motorola MC68000

The facilities of the 68000 are too extensive to more than summarise. You should obtain a copy of the relevant Motorola manual if you need more than an outline.

Memory transfers

The MC68000 can address 16 Mbytes of memory. This is a 24-bit address range - the microprocessor's internal addresses are in fact 32 bits, but the eight most significant address lines (A24-A31) never see the light of day due to IC packaging constraints.

As with all of Motorola's microprocessors, memory and I/O devices are mapped into the memory space. The MC68000 does not use a separate I/O space with its associated special I/O instructions, as do the Intel and Zilog microprocessors. I/O devices are written to and read from in the same way as memory. You will see the effects of this when you examine the 1616 memory map.

Data alignment

There are 16 data lines connecting the microprocessor to its memory. The MC68000 may read or write data in either 8 bit or 16 bit quantities. The signals which determine the amount of data to be transferred are /UDS (upper data strobe) and /LDS. If /UDS if asserted (set low) during a memory cycle, 8 bits are transferred across D8-D15. If /LDS is asserted, 8 bits are transferred across D0-D7. Asserting /UDS and /LDS together causes a 16-bit transfer.

Although the data bus is 16 bits wide, the MC68000's addresses refer to byte (8 bit) quantities. Bytes which are at even addresses are transferred on data lines D8-D15 (/UDS asserted). Odd address transfers occur across D0-D7. This is why the microprocessor has no A0 (address line zero) pin - the least significant address bit is kept internally and is used to determine which of /UDS and /LDS is to be asserted during a byte transfer. You will see the effects of D0-D7 being at odd addresses if you compare the list of I/O device addresses with the data line connections shown in the circuit schematic.

Reading or writing a 16 bit quantity at an odd address would require two memory accesses to two different 16 bit addresses. The MC68000 does not permit this. 32 bit reads and writes are accomplished by doing two accesses to consecutive 16 bit addresses.

Let me emphasise that point: odd addresses are not used for normal access. When writing code, you must ensure that addresses are on even boundaries.

Data transfer mechanism

The MC68000 uses asynchronous memory transfers. This means that instead of having a fixed memory access timing scheme, the MC68000 requests access to a memory location and waits until external hardware signals that the memory has responded.

The signals which are used to indicate a memory access are /AS (Address Strobe), /UDS and /LDS. When the external memory address decoding hardware detects the assertion of these signals along with a valid address it waits until the memory has had time to respond and then asserts the MC68000's /DTACK (Data Transfer Acknowledge) signal. The microprocessor then negates (raises) /AS and terminates the memory cycle.

Bus errors

The asynchronous memory transfer mechanism essentially involves inserting memory wait states into the processor's bus cycle. Different types of memory have different access (response) times and so the address decoding circuitry must ascertain from the address what type of memory is being accessed and insert an appropriate wait period. For example, access to video memory sometimes adds a single wait state. If running at 15 MHz, video access adds an extra three wait states.

If the processor attempts to access an address which no external circuitry recognises, it will never receive a /DTACK signal and will wait indefinitely. Note that such an address is an invalid one to which no device is mapped, and a reference to it could only arise from a programming fault.

Such an indefinite wait is terminated by yet more external hardware, which watches for a processor memory request (an assertion of /AS) and when one is detected its duration is timed. If it is determined that /AS has been asserted for too long, then it is assumed that the processor is accessing an undecoded (invalid) address and the external circuitry asserts the MC68000's /BERR (Bus Error) signal. This causes the processor to abort the offending memory access and to commence execution of a user-supplied error handling routine.

The mechanism of externally detecting an access to an invalid address by measuring the bus response time is known as a bus error, a bus timeout, or a memory fault.

6800 device interface

The MC68000 signals E (Enable clock), /VMA (Valid Memory Address) and /VPA (Valid Peripheral Address) facilitate a simple interface to devices from Motorola's 8 bit microprocessor family, the MC6800 series. When the address decoding circuitry recognises an access to an MC6800 series device, it drives the MC68000's /VPA signal low, and the processor then uses an MC6800 style memory transfer timing scheme, which is quite different from that described above.

The 1616 uses this mechanism for addressing all of its I/O except the 8 bit output latches. This includes, of course, the 6845 CRTC and the 6522 VIA, both of which have MC6800 compatible bus timing.

Interrupt implementation

The MC68000 uses a prioritised interrupt scheme. It supports eight interrupt levels. The processor has an internal status register which determines its current interrupt priority level. This ranges from zero (lowest) through to seven (highest). External hardware requests an interrupt by encoding a number onto the MC68000's interrupt request pins /IPL0 - /IPL2. If the encoded number exceeds the processor's current priority level, the current priority level is moved up to that which was requested and the MC68000 commences execution of a user-written interrupt handling program. When this program has completed, the processor returns to its previous priority level.

Using this scheme external interrupting devices may be organised according to the urgency with which their interrupt requests must be handled. A high priority interrupt request may temporarily pull the processor out of handling a lower priority interrupt.

An example of this may be seen in the 1616's interrupt organisation. None of the VIA's interrupt sources need be serviced with great urgency, so the VIA is put at interrupt priority level two. For high speed communications we wish to respond quickly to the serial I/O interrupts, so the SCC is put at priority level three. This means that if the processor is dealing with a VIA interrupt when the SCC makes its higher priority interrupt request, the VIA interrupt handler will be temporarily suspended whilst the SCC interrupt is processed. The converse does not apply: if an SCC interrupt is being serviced when the VIA requests an interrupt, the VIA request will be ignored because its level is 2 whilst the processor is currently at priority level 3. The level 2 request will be recognised when the processor's priority returns to level 1 or level 0.

If appropriate 'intelligent' peripheral devices are present, the interrupting device can put an 'interrupt vector' on the data bus, upon receiving an acknowledgement of the interrupt from the 68000. Using this scheme, the vector can select one of 192 interrupt service routines. Due to the nature of the peripheral devices used, this scheme is not available in the Applix 1616.

3
Memory Map

Memory maps are not of all that much use in a system such as the Applix 1616. This is because, with certain exceptions, programs are almost always relocatable, and can be placed anywhere in memory. Some memory areas, such as the system vector table, are common to all 68000 systems, since they are determined by how the 68000 works.

Locations of routines in the eproms are not given. These locations change with every revision. Access to the eprom routines should be via the ***syscall*** mechanism, as described in the *Programmers Manual*.

However, since hackers expect them, here is a general outline of the relevant memory and input output locations.

Address start	Address end	Purpose
000000	07FFFF	On board RAM, including video memory. Moved up by 1 MB boundaries to allow for external RAM
000000	3FFFFF	External RAM memory
400000	47FFFF	On board RAM, on a system fitted with maximum of 4 megabyte of external memory
480000	4FFFFF	Unused
500000	5FFFFF	On board EPROM area
600000	6FFFFF	IO ports and latches
700000	7FFFFF	Peripheral chips and devices
800000	FFC000	1616/OS searches this area on $4000 boundaries for EPROM
FFFFC0	FFFFFF	Disk controller board

There are also a number of cards and other equipment designed for operation with an Applix, or that use the Applix operating system. To avoid memory conflict, addresses for new equipment should only be assigned in consultation with Applix designer Andrew Morton. Here is a list of known uses:

Address start	Address end	Purpose
FFFF00	FFFFFF	SSDCC ports
FFFEC0	FFFEFF	34010 graphics card
FFF000	FFF0FF	Kevin Bertram's transputer board
FFE000	FFEFFF	KCS player activated terminal I/O board
FFDE00	FFDFFF	Memory board / SCSI interface
F40000	F7FFFF	KCS dot addressable display controller
F37000	F37FFF	David Taylor's EPROM programmer
F00000	?	Memory board MMU page translation and history RAM
EF0000	EFFFFF	KCS writer I/O board
EEFF00	EEFFFF	John Telek's ADC card
EEFC00	EEFDFF	Kevin Bertram's I/O space
EE0000	EEFDFF	free?
ED0000	EDFFFF	Michael Milways?
EC0000	ECFFFF	Michael Milways?
E00000	E0FFFF	Kevin Bertram's ROM
DFFC00	DFFCFF	John Taylor's numeric control milling machine

Transient program memory model

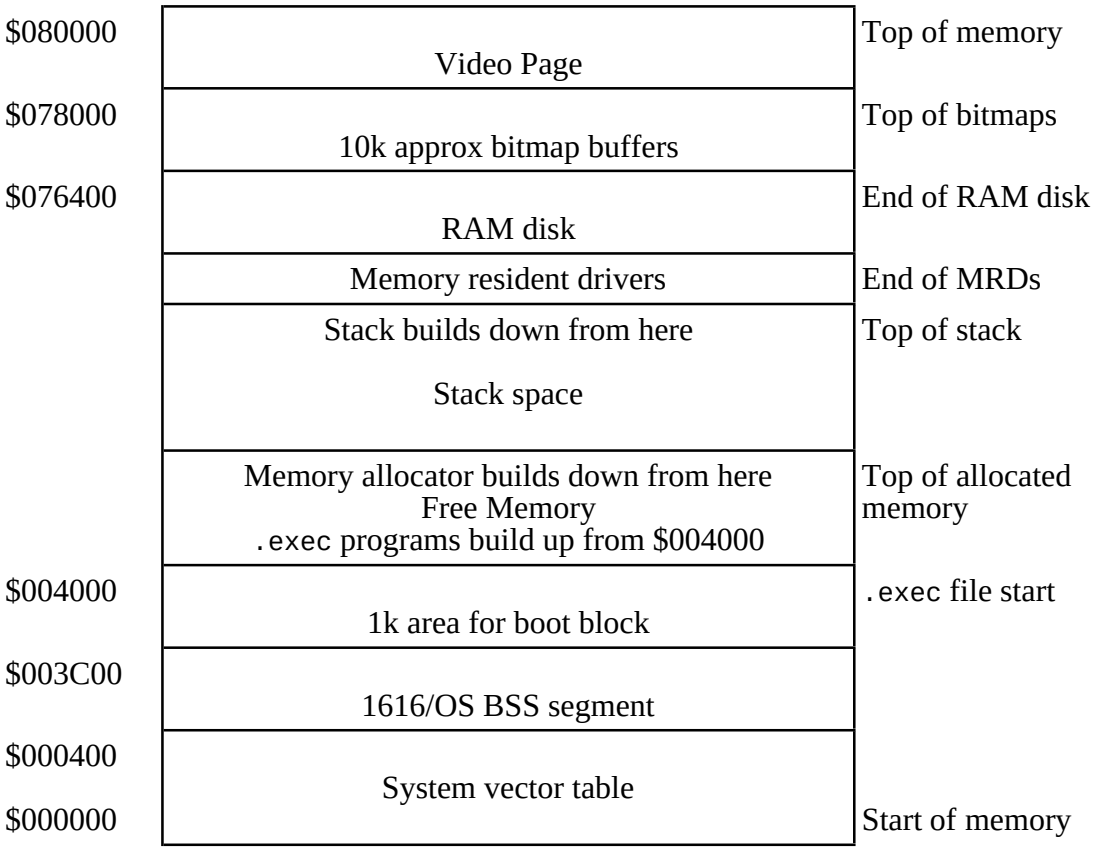
A transient program cannot use any memory without requesting it from the system memory allocator. See the memory manager section of the *Technical Reference Manual* for details.

The memory range $0 to $3FF is reserved for 68000 vectors and some system use.

The memory range $400 to $3BFF is reserved for 1616/OS usage.

The range $3C00 to $3FFF is used for copying in the boot block from a disk device whenever the 1616 is reset.

Normal program memory extends from $4000 up to the end of RAM, which on an unexpanded 1616 is at $7FFFF



The organisation of memory is determined to some extent by the contents of the MRDRIVERS file, or by the setting of the DIP switches on the 1616 motherboard, so precise addresses cannot be given here.

The size of the video page, RAM disk, MR drivers, and the stack space are determined by the contents of the MRDRIVERS file which is loaded at boot time.

The bitmap buffers are reserved for buffering the bitmap blocks of /RD, /F0, /F1, /H0 and /H1. The 1 kbyte typically used for each additional hard disk driver is sufficient for an 8 megabyte volume. The first hard disk volume typically has room reserved for 40 megabyte.

If a memory expansion is fitted, the items above the memory allocator free memory area move up a megabyte at a time, depending on how many megabytes are fitted to the expansion board.

I/O addresses

The 1616's I/O devices are memory mapped. Their addresses are given here mainly for reference purposes; if possible you should use the available system calls for I/O. All devices are one byte wide and are addressed as follows:

Address	Mnemonic	Usage
$600001	CENTLATCH	Centronics (parallel printer) latch
$600081	DACLATCH	D/A converter latch
$600101	VIDLATCH	Video latch
$600181	AMUXLATCH	Analogue multiplexer latch
$600000	PAL0	Video palette entry 0
$600020	PAL1	Video palette entry 1
$600040	PAL2	Video palette entry 2
$600060	PAL3	Video palette entry 3
$700000	SCCBCONT	SCC channel B control register
$700002	SCCBDATA	SCC channel B data register
$700004	SCCACONT	SCC channel A control register
$700006	SCCADATA	SCC channel A data register
$700081	IPORT	The input port
$700100	VIABASE	VIA base address. VIA registers start at this address and appear at every second byte address.
$700180	CRTCADDR	MC6845 CRTC address register
$700182	CRTCDATA	MC6845 CRTC data register

Shadow registers

The four latches and the video pallette are write-only, and we need to know their current contents if we are to alter only some of their bits. For this reason there are eight bytes, called shadow registers, which contain the current contents of the latches and pallette. The shadow registers should be updated when the latches and pallette are changed (if appropriate syscalls are used, updating is done). The shadow registers are listed in the *Programmer's Manual*.

Simulated interrupt vectors

The interrupts are autovectored on the 1616, and the SCC and the VIA do not support multiple interrupt vectors, so 1616/OS simulates multiple interrupt vectors for these devices. When the VIA or the SCC interrupts for any reason, 1616/OS ascertains from the device the reason(s) for the interrupt, and vectors through one or more of the following addresses:

(ISR stands for 'interrupt service routine')

$100 Pointer to VIA timer 1 timeout ISR

$104	Pointer to VIA timer 2 timeout ISR
$108	Pointer to VIA CB1 ISR
$10C	Pointer to VIA CB2 ISR
$110	Pointer to VIA shift register ISR
$114	Pointer to VIA CA1 ISR
$118	Pointer to VIA CA2 ISR
$140	Pointer to SCC channel A character receive ISR
$144	Pointer to SCC channel A character transmit ISR
$148	Pointer to SCC channel A external/status ISR
$14C	Pointer to SCC channel B character receive ISR
$150	Pointer to SCC channel B character transmit ISR
$154	Pointer to SCC channel B external/status ISR

It is the responsibility of each called interrupt service routine to clear the source of its interrupt (and no others) from the interrupting device.

Interrupt priorities

1616/OS requires that the 1616's interrupting devices be set at the following priorities:

Cassette IRQ	Level 4	$70
SCC IRQ	Level 3	$6C
VIA IRQ	Level 2	$68

If you mask off some or all of these interrupts by raising the processor priority, do it for as short a time as possible because the keyboard, vertical sync interrupts, date/time drivers, keyboard, cassette, sound and serial communications are all interrupt driven.

All interrupts on the 1616 are autovectored, so the interrupt vectors are in the range $64 (level 1) to $7C (level 7)

Video colours

The 16 video colors are selected by writing 4-bit nibbles to either the video RAM (in 320 column mode) or to the palette (640 column mode) or to the video latch for the borders (both modes).

The video latch is at $600101, and controls the border colour. Please note that many (most?) multisync monitors explicitly use the border colour output to determine the 'black' level for their operation, and therefore you can not effectively use a border colour other than black on these monitors. Several software packages, including SSEG (EGA) and MGR, use bits in the border colour latch to toggle their operation.

Use of the software shadow register at $300 is described in more detail in the *Programmer's Manual*.

On a monochrome monitor the brightness increases with increasing nibble value, with a value of 0000 corresponding to black.

The colours map as follows:

Number	Colour	Mnemonic

0	Black	PC_BLACK
1	Dark grey	PC_DGREY
2	Dark blue	PC_DBLUE
3	Mid blue	PC_MBLUE
4	Dark green	PC_DGREEN
5	Green	PC_GREEN
6	Blue grey	PC_BGREY
7	Light blue	PC_LBLUE
8	Dark red	PC_DRED
9	Red	PC_RED
10	Dark violet	PC_DVIOLET
11	Violet	PC_VIOLET
12	Brown	PC_BROWN
13	Yellow	PC_YELLOW
14	Light grey	PC_LGREY
15	White	PC_WHITE

Boot Sequence

The following things happen when the system is reset. Those steps marked † only occur at level 0 resets.

After version 3.2, the CRTC controller is initialised prior to any code which uses RAM (this was required for development of the 15 MHz version). CRTC registers 13 and 14 are used to test whether a power on reset was made (RAM contents were formerly used, however some RAM retains its contents too long after a power down for this to be guaranteed).

- The stack pointer is initialised to $10000
- The RAM system call vector table is initialised.
- All I/O devices and major internal program modules are initialised.
- A small model memory manager is installed, using the $4000 - $10000 area as free memory.
- Default values for the RAM disk size, stack space and video RAM space are installed.†
- A search is made on the /F0, /F1, /H0 and /H1 drives (if present) for the MRDRIVERS file. If found, it is loaded in and new values for the RAM disk size, stack space and video RAM space are installed. The memory resident driver code is loaded in and relocated.†
- The stack pointer is moved to point to the area just below the MR drivers.
- Interrupts are enabled.
- A search is made from address $800000 through to $ffc0000 in $4000 byte increments for a ROM wth the values $12, $B5, $06, $A7 at the start. For each external ROM with this pattern at the start, the OS performs a JSR to the start of the ROM + 4. A zero is passed at 4(sp) and the reset level at 8(sp). At this point external ROMS can perform whatever initialisation is necessary for their purposes.
- The previous step is repeated, except a value of 1 is passed at 4(sp) to any called ROM code. It is at this point that an external ROM can take control of the system, with all of the normal system resources available.
- Every memory resident driver is called with command number 0, 1 or 2, depending upon the reset level.

- A search is made of the /RD, /F0, /F1, /H0 and /H1 devices for a bootable device. If one is found (BOOTBLOCK field of the root block non-zero), the boot block is read from the device into memory at $3c00. The system then performs a JSR to address $3c00, passing the reset level at 4(sp), and the number of the boot device driver at 8(sp). The /RD driver is device 0, /F0 is device 1, etc. Note that it is permitted to boot from the ram disk after a reset, if desired (obviously this won't work on power up).
- The system drops into an infinite loop, performing ***iexec*(1)** system calls.

4
The 1616's Hardware

Here we describe the general design of the 1616 mother board, and detail the operation of specific circuits.

EPROMs

The EPROMs are read only memory, and contain the 1616 operating system, boot routines, and a range of utility programs.

The EPROM decoding circuitry is very simple. Any address in the one megabyte range $500000 - $5FFFFF (Hex numbers) enables the EPROMs. This memory select is available at pin 10 of U59 (74LS138) (pins 15 down to 9 select one megabyte areas from 0 to 6 megabyte, while pin 7 selects the 7th megabyte).

The address on power up or reset is more complicated.

The MC68000 requires valid data at addresses $000000 to $000007 immediately after it is reset (to provide the supervisor stack address and program counter contents). This is provided by initially enabling the EPROMs at all addresses throughout the entire memory map. The processor shortly commences execution in the normal EPROM address range at $500000 and the memory map is restored to normal as soon as the processor performs a write operation.

Address selection throughout all memory is provided by a two input AND gate, pins 8, 9 and 10, in U58 (74LS08). Since the ROMSEL is active low, the AND gate output (pin 8) selects the EPROMs whenever either of the input lines are low. One input (pin 10) comes from pin 10 of U59, the 74LS138 main system 3 to 8 decode. Thus the eproms are selected whenever the memory range $500000 to $5FFFFF is addressed.

The other input to the U58 AND gate (pin 9) comes from the RESET circuit. This reset line is normally high, however it goes low on power up, and when the reset button is pressed. The same reset line connects to pin 69 of the expansion slot connectors (to provide an external reset signal). The normally high reset line also connects to the pin 6 (G1) active high select line of U59 (74LS138), the main address decode. This means that the main address decode chip is selected always, except during a reset. Thus during reset, all the U59 decode (active low) outputs are high. This means nothing in the memory map is selected during reset. However, as both the Eprom select line, and the active low Reset line are connected to the AND gate, during reset, the output will be low, selecting the Eprom throughout memory.

RAMs

Decoding

The 1616 has 512 kilobytes of dynamic RAM on board, organised as 256k of 16 bit words. The base address of the on-board RAM may be set at either the 0, 1, 2, 3, or the 4 megabyte boundary using the 'mem' strapping block. For software reasons it is intended that all the RAM in the system be in one block, starting from address $000000. Any expansion RAM boards will have either 1, 2, 3 or 4 megabytes of storage, and when they are installed the on-board RAM's address must be shifted up to the next address beyond the expansion RAM.

The 'mem' strapping blocks are merely connections between the RAMSEL line, and pins 15 to 11 of U59 (74LS138), the main address decode chip. These pins select at 1 megabyte boundaries, from 0 to 4. See the notes on address decode for further details.

Timing

The 1616's memory timing scheme is quite complicated and lies at the very heart of the system.

The complexity of the timing is due to the need to resolve any contention between the MC68000's RAM accesses and those of the video circuitry. The video absolutely must read 16 bits from the RAMS at a rate of 1.875 MHz, or else the display image would be affected. For this reason the processor access is synchronised to that of the video. For most of the time the interleaved video/processor memory accesses are quite transparent and the processor runs without any wait states - the video accesses are squeezed into the first half of the MC68000 bus cycle. Relatively occasionally the MC68000 will lose synchronism with the video and it must be delayed (by retarding /DTACK) in such a way as to cause the processor to resynchronise.

Use of a 10 MHz 68000 is required, to ensure that timing limits are met. Although the 68000 clock is 7.5 MHz, and thus within the timing specifications of an 8 MHz 68000, the need to interleave the video into the bus cycle demands that a 10 MHz 68000 be used. Any other alternative would require wait states. You may note that the 15 MHz conversion requires one wait state when accessing most memory, but four wait states when the video display is enabled.

It should be noted that this synchronisation only applies to the on-board RAM. All other devices on the bus are accessed in a more conventional manner. This is possible because the video data paths are isolated from those of the rest of the system by the 74LS244 tri-state buffers U28 and U30. The video essentially has a private data bus and must only synchronise with the MC68000 when there is to be an exchange of data.

Benchmarks indicate very little speed difference between programs which run in RAM and those in ROM, indicating that the synchronisation delays are not significantly slowing the processor.

Note that the MC68000 and the video circuitry accesses result in the on-board RAM being accessed at a continuous rate of 3.75MHz. There are 7.5 megabytes per second going across the data bus and the RAMs are being cycled at 267 ns. The minimum specified cycle time for 150 ns dynamic RAMS is 250 ns and it is this which limits the 1616's processor clock frequency to 7.5 MHz.

A 16R8 PAL (U51) and a 74F74 flip-flop (U50) are both driven at 30MHz and together they perform the following critical timing functions:

- Generation of the 7.5MHz, 3.75MHz and 1.875MHz signals for system timing control functions.
- Processor/video synchronisation by appropriately delaying /DTACK and the RAM control signals.
- Generation of the RAM /RAS (Row Address Strobe) signal.
- Generation of the /MUX signal which multiplexes the memory address (from either the MC68000 or the video circuitry) onto the RAM chips.
- Generation of the /CASU and /CASL (Column Address Strobe Upper and Lower) signals which enable the RAM chips. The assertion of these signals corresponds to the assertion of the /UDS and /LDS MC68000 signals.
- Generation of the /LPULSE (Load pulse) signal which causes the video shift registers (U29 and U31) to be loaded from the RAM chip outputs.
- Enabling the /RAMWE (RAM Write Enable) signal.
- Enabling the /DBRE (Data Bus Read Enable) signal which turns on buffers U28 and U30, driving the system data bus with the data read from the RAM.

How's that for two chips!

About PALs

A PAL (Programmable Logic Array) is a 20 or 24 pin IC which may be once-only programmed in a manner similar to bipolar fusible-link PROMs. A PAL may replace as many as ten catalogue logic ICs when used to implement complicated logic functions.

The 16R8 PALs used in the 1616 consist of an array of AND gates which drive an array of OR gates, the outputs of which are latched and then passed to the outside world. When the device is in its unprogrammed state the inputs of the AND array are connected to all of the IC's inputs and to all of its outputs. A PAL is programmed by electrically removing some of the internal fuses which connect the AND array to the PAL's inputs and outputs. Designing a PAL fuse map involves selecting those fuses which are required to implement the desired logic function and then blowing all of the others in a suitable programmer.

A useful feature of the 16R8 PAL is its output registers. A group of eight latches on the outputs permit the storage of previous state information for the generation of complex timing sequences. The latches virtually eliminate timing skew between the PAL's output signals; when the PAL is clocked all of the output changes occur within a very short period, which eliminates many design uncertainties concerning device propagation delays.

Processor support

There is an amount of random logic around the processor which does the following things:

Bus fault timer

U33, a 74LS123, is a retriggerable monostable (or one-shot) which monitors /AS and asserts /BERR on a timeout. Pins 10 and 11 are always pulled high by a resistor network, so the output pulse is triggered by a high to low transition by the /AS. The timeout period is determined by C48 (100n) and R40 (3K3), and should be round 1/10,000 of a second. Incidently, pin 12, the /Q output, is not used. With a bit of work you could hang a bus error LED from it.

Interrupt encoding

U56 is a 74LS148 eight input priority encoder which is used to encode the level of the highest priority interrupt which is currently pending onto the processor's interrupt request pins.

/DTACK generation

U57 (74LS11, three input AND) and U58 (74LS08, 2 input AND) are used to generate the processor /DTACK signal when the decoding circuitry detects a valid memory reference. This is basically when you have /ROMSEL (500000 to 5FFFFF), or an I/O select (600000 to 6FFFFF), or an external /DTACK, or /DTACKR (for RAM) from the timing PAL.

/VPA generation

U54 (74LS04 inverter) and U57 (74LS11, three input AND) detect when the MC68000 is addressing an MC6800 type address and inform the processor of this by asserting /VPA. Essentially this is when addresses in the 700000 to 7FFFFF range occur. The /VPA signal is also asserted when the processor is first responding to an interrupt (FC0, FC1 and FC2

all high, as occurs only on interrupt acknowledge in supervisor mode), to inform the processor that the interrupting device does not support the MC68000 vectored interrupt scheme.

Reset circuitry

The MC68000's /RESET and /HALT lines are normally held high via part of resistor network RN5. They can each be pulled low by a Schmitt inverter (74LS05, U61, pins 4 and 6), which in turn are driven by U61 pin 2. The input to this is forced low following the discharge of C11 (10 μF tantalum capacitor) connected across the reset switch SW1. When both /RESET and /HALT are driven low for over 10 clock cycles in this manner, the CPU is reset. An initial power up requires that the /RESET and /HALT lines be driven low for at least 100 milliseconds.

When reset, the 68000 reads the first four bytes of the reset vector table, starting at address 0, and places them in the supervisor stack register. It then reads the next four bytes, starting at address 4, and places them in the program counter. Then it drives its reset pin for 124 clock periods, to force a reset of external devices. After that, it starts executing the instructions pointed to by the contents of the program counter.

The only real complication with this reset scheme is that our instructions are in Eprom, at $500000 to $5FFFFF, not in RAM at 0, where the processor looks. To provide for this, all the chip select lines are disabled on reset, and the eprom select is enabled throughout all the addressing range, as explained in the section on eproms.

The reset line that selects the eproms comes from pin 5, the Q output of U60, a 74LS74 D flip flop with clear. The clock and data inputs (pins 2 and 3) of U60 are permanently connected to +5 volts. The reset or clear (U60, pin 1) is connected to the original reset signal, and when this goes low, produces a low on output pin 5, thus selecting the eproms via AND gate U58.

The original reset line becomes high again after the capacitor C11 discharges. The output on pin 5 of U60 remains low. However the set line, pin 4 of U60, is connected to the 68000 R/W line. When the R/W line goes low during the first /write made by the 68000, this changes output pin 5 of U60 back to high, thus restoring the normal memory map.

Address decoding

The main address decoding is done by U59, a 74LS138 3 to 8 decode. This has three chip select lines. The active low chip selects are connected to A23, and /AS from the 68000. Thus, the chip only selects when /AS is low, and an address on the bus, and only selects the bottom 8 megabyte of memory. The top 8 megabytes of memory are not used. The active high chip select is connected to a normally high reset line, which goes low on reset. Thus, every address is deselected on reset.

Address lines A20, A21 and A22 connect to the input lines of U59 (74LS138), thus providing active low address selects at 1 megabyte boundaries on the 8 output lines, starting at address 0. The first five outputs (pins 15 to 11) go to the 'mem' strap block, to select the base address of the onboard memory. Pin 10 ($500000 to $5FFFFF) is used for /ROMSEL, for the on-board EPROM area. Pin 9 ($600000 to $6FFFFF) is the I/O select (various latches for video palette, Centronics port, DAC, etc.) Pin 7 ($700000 to $7FFFFF) is the /EVPA line, for peripheral selection of the CRTC, VIA, input port, and SCC serial ports.

Extra address decoding for /EVPA is provided by half of U64 (74LS139), a dual 2 to 4 decode. This is selected by ORing (U24 - 74LS32) /EVPA and /VMA. Since the 74LS139 chip select is active low, this occurs only when the 68000 asserts /VMA low, and the address is in the range $700000 to $7FFFFF. The two inputs to this half of U64 are

connected to A7 and A8. This provides four outputs on pins 12, 11, 10 and 9 of U64, each being 128 ($80) from the next. These locations repeat every 512 ($200) locations through the entire $700000 to $7FFFFF area.

The devices decoded in this manner are:

SCC	$700000	Serial ports
Riport	$700081	DIP switches, joystick push buttons, analogue inputs, cassette input
VIA	$700100	6522 VIA
CRTC	$700180	6845 CRT controller

Video

The MC6845 CRTC video controller (U27) generates timing and addressing signals which interact with the timing PAL to load 16 bits of data from the RAMS into the shift registers (U29 and U30, both 74LS166 8 bit parallel in to 1 bit serial out) every 533 ns. This data is shifted two bits at a time into the video PAL (U5) at 15 MHz.

The 8 bit video latch (U16 - 74LS374) holds the video border colour (4 bits) and the video display RAM address offset (4 bits). Using this latch, the video display RAM may be set to start at any 32k boundary within the on-board RAM (normally it is the highest 32k).

This permits the use of multiple display pages to implement animated displays. The 1616 operating system ROMs fully support this feature.

640 column mode

In this mode, video data is shifted out at 15MHz. The two bits coming from the shift registers are used as an index into the 4x4 bit register file, U4 (74LS 670). The four x four bit words in the register file are written by the processor, and allow the programmer to map each of the four colour combinations into one of the sixteen available colours.

320 column mode

In this mode the RGB and I signals are generated by the video PAL (U5), rather than being read from the register file (U4). The PAL simply remembers the previous two bits from the shift registers and presents them alongside the current two. This is essentially a serial to parallel conversion and it produces a 7.5 MHz stream of 4 bit words.

The border colour may be any of 16 colours, independent of the video mode.

The data selector U3 (a PAL which in this mode emulates a 74LS298, quad 2 mux with storage) selects the video data from either the PAL or the register file, latches it internally and presents it to the video mixing circuitry.

The video mixing circuitry converts the video data into both a standard IBM RGBI output and into a 16 grey scale composite sync signal.

MGR mode

On the Revision C board, provision is made for an additional black and white (single pixel) video mode, programable up to 960 by 512 pixels. This uses a special PAL at U3 (replacing the 74LS298 multiplexor used in Revision B boards). MGR software, ported from Stephen Uhler’s version for Sun workstations, is used to provide a multitasking window display, with mouse control. Disk drives (and preferably memory expansion) are required for effective use of this display mode.

CRT controller

The 6845 CRT controller is located at $700180, but has its data register select pin RS (pin 24) connected to A1, or address $700182. Registers in the CRTC are accessed by writing the register number required to $700180, and then reading from or writing to $700182.

The Applix 1616 uses only the first 14 registers, numbered 0 to 13. Understanding their contents is slightly complicated because the 6845 is intended for 'character' oriented displays, however the Applix uses only a graphics display. Thus the normal contents of the 6845 indicate it is displaying 50 rows of 4 scan line high characters, to give 200 lines on the screen.

Reg No	Content	Register	Meaning
0	119	Hor total	120 characters including retrace
1	80	Hor displayed	Show 80 characters
2	96	Hsync position	Start sync
3	10	Hsync width	
4	77	Vertical total	77 characters vertical
5	0	Vtotal adjust	
6	50	Vertical displayed	Show 50 characters vertical
7	63	Vsync position	Start sync
8	0	Interlace	No interlace
9	3	Scan lines/row	Show 4 scan lines per row
10	32	Cursor	Disable cursor, done in software
11	0	Cursor end	
12	0	Start address MSB	Video address, not used
13	0	Start address LSB	

You will notice a 1.875 MHz clock (listed incorrectly in some schematics as 1.825 MHz) at pin 21 of the CRTC. This is divided by the (contents of register 0) plus 1 (or 120), and the result is 15625, or the horizontal sync frequency required by a standard TV style monitor. (Contents of register 9) + 1 equal 4, the number of scan lines per row, and this is multiplied by the (contents of register 6), or 50 displayed characters, to give 200 lines displayed. The vertical refresh rate is determined by ((contents of register 4) plus 1) multiplied by ((contents of register 9) plus 1) plus (contents of register 5) plus 1. This makes 312 vertical lines, or a vertical frame refresh frequency of 50.0801 etc. The difference arises because we really need 625/2 lines to exactly match a TV, and we are half a line out.

Note that the video is adjustable, and other video outputs can be programmed. See the shareware video drivers by Conal Walsh, for an example of how to produce EGA (640 by 350) video output.

Screen memory mapping

The mapping of screen memory addresses to screen coordinate positions is fairly straightforward as software scrolling has been implemented. The exact mapping is explained in detail in the Programmer's Manual. There is extensive software support for video display manipulation in the 1616's Operating System ROMs.

The 1616 video, unlike most computers, has no text only mode. Due to the speed of the MC68000, it is possible to draw characters onto the screen from a table of character shapes stored in memory. This table is stored in a compressed format in the 1616's ROM and is transfered into main memory at power on. There are two main advantages to this method. Firstly, it allows an infinite number of display formats, including different font styles and sizes. Secondly, it simplifies the design of the video circuitry. Check the video terminal emulation escape modes, in the *Programmer's Manual*, for examples of bold, underline, italic, supscript and superscript characters.

The use of an MC6845 allows the programmer to select a wide range of video formats, including interlaced mode and variable horizontal and vertical formats. The 1616's video drivers initialise the CRTC for 640 (or 320) pixel columns, and 200 scan lines. This requires 32k of video RAM. The number of pixel columns and lines may be increased or decreased, but the programmer will have to write his own drivers to support non-standard configurations.

Versatile interface adaptor (VIA - 6522)

The VIA is a complex and powerful interface chip providing two 8-bit bi-directional I/O ports, whose lines can be individually set for either input or output. It includes interrupt facilities, two 16 bit programmable counter timers, and a rudimentary serial port. Several of the I/O lines can be controlled from the timers to produce programmable square wave outputs, or for counting externally generated pulses.

The clock input is the 68000 generated E clock, designed for the 02 clock input on 8 bit peripherals such as the 6522 VIA and 6845 CRTC. When generated by the VIA, the interrupt line output, /VIAIRQ, produces a Level 2 interrupt on the 68000 CPU.

Port A is entirely used by internal devices, as follows.

PA7	/CLRKBL	Clears the keyboard shift register
PA6	/KBRES	Keyboard reset
PA5	/CLRCASIRQ	Clear the cassette interrupt, cassette output line
PA4	/ECASRIRQ	Enable cassette read IRQ
PA3	EN640	640 mode input to timing PAL
PA2	/ECASWIRQ	Enable cassette write IRQ
PA1	/CENTSTB	Strobe the printer port line 1
PA0	/CENTBUSY	Check for printer busy signal on line 21

Access to port B is available at the User Port, pins 13 to 20, however PB7 is used as an output by the system to generate /CASIRQ. This port line was used because it can produce a regular signal under control of one of the VIA timers.

The CB1 and CB2 interrupt inputs are used for keyboard clock and data input. Under control of the VIA shift register, they can act as a serial to parallel converter, perfect for keyboard input.

CA1 and CA2 act as interrupt inputs for the /CENTACK printer acknowledge line, and for the vertical sync pulse generated every 20mS by the CRTC chip. The vertical sync interrupt is available for use by memory resident drivers and similar timing uses.

VIA register addresses

The VIA is nominally mapped to the 16 even addresss (data lines D8 to D15) between $700100 and $70011E. As addresses are only partly decoded, the locations are actually for the 128 bytes from $700100 to $71017F, and is repeated every 512 bytes throughout the entire megabyte from $700000 to $7FFFFF.

700100	B port I/O register, r/w
700102	A port I/O register, r/w with handshake
700104	Data direction register for B, r/w
700106	Data direction register for A, r/w
700108	Read timer 1 counter low order byte
	Write sets timer 1 latch low order byte
70010A	Read timer 1 counter high order byte
	Write sets timer 1 latch high order byte, and initiates count
70010C	Read timer 1 latch low order byte
	Write sets timer 1 latch low order byte
70010E	Read timer 1 latch high order byte
	Write sets timer 1 latch high order byte
700110	Read timer 2 low byte, and reset interrupt
	Write sets timer 2 low byte, interrupt not cleared
700112	Read timer 2 high order byte, interrupt not cleared
	Write sets timer 2 high byte, and resets interrupt
700114	Serial I/O shift register
700116	Auxiliary control register
700118	Peripheral control register
70011A	Interrupt flag register
70011C	Interrupt enable register
70011E	A port I/O (700102), without handshake

VIA register bit patterns

700104	Data direction B	0 sets for input, 1 for output
700106	Data direction A	0 sets for input, 1 for output

700116	Auxiliary control	
Bit No	Bit value and use	Register
7	0 = one-shot mode	Counter 1 control
	1 = free running	Counter 1 control
6	0 = PB7 pin not used	
	1 = PB7 used as output	
5	0 = on clock in one-shot	Decrement counter 2
	1 = on PB6 input	Decrement counter 2
4,3,2	000 = disable shift register	
	001 = shift at counter 2 rate	
	010 = shift in at clock rate	
	011 = shift in at external clock rate	
	100 = free-running output at counter 2 rate	
	101 = shift out at counter 2 rate	
	110 = shift out at clock rate	
	111 = shift out at external clock rate	
1	0 = disable inputs	B port
	1 = enable inputs	B port
0	0 = disable inputs	A port
	1 = enable inputs	A port

70011A	Interrupt flag register
Bit No.	Use
7	1 = any interrupt is active, 0 = no interrupt
6	1 = Timer 1 interrupt
5	1 = Timer 2 interrupt
4	1 = Shift register interrupt
3	1 = CB1 interrupt

2	1 = CB2 interrupt
1	1 = CA1 interrupt
0	1 = CA2 interrupt

70011C	Interrupt enable register
	0 = interrupt disabled 1 = interrupt enabled
Bit No.	Use
7	controls enable/disable function
6	timer 1
5	timer 2
4	shift register
3	CB1
2	CB2
1	CA1
0	CA2

70011B	Peripheral control register
Bit No.	Use
7, 6, 5	000 = CB2 input mode 001 = CB2 independent input mode 010 = CB2 input mode 011 = CB2 independent input mode 100 = CB2 output low on CPU write 101 = CB2 output low pulse on CPU write 110 = output CB2 low 111 = output CB2 high if 000 or 001 then interrupt is high to low on CB2 if 010 0r 011 then interrupt is low to high on CB2 interrupt flag register bit 2 is set in each case.
4	0 = high to low on CB1 causes interrupt 1 = low to high on CB1 causes interrupt interrupt flag register bit 3 is set in each case.
3, 2, 1	000 = CA2 input mode 001 = CA2 independent input mode 010 = CA2 input mode 011 = CA2 independent input mode 100 = CA2 output low on CPU write 101 = CA2 output low pulse on CPU write 110 = output CA2 low 111 = output CA2 high if 000 or 001 then interrupt is high to low on CA2 if 010 0r 011 then interrupt is low to high on CA2 interrupt flag register bit 0 is set in each case.
0	0 = high to low on CA1 causes interrupt 1 = low to high on CA1 causes interrupt interrupt flag register bit 1 is set in each case.

Keyboard Interface

The IBM XT compatible keyboard produces either 10 or 11 bits of clocked serial output depending upon the style of the keyboard. Schmitt triggers in U34 clean the noise from the incoming clock and data signals. The leading 2 or 3 clock pulses from the keyboard are then discarded by the latches U53 and U49 and the remaining 8 pulses are used to clock the data into the VIA's internal shift register. Filling this register causes the VIA to interrupt the 68000.

Software then reads the incoming data from the VIA and lowers the VIA /CLRKBL signal, preparing U53 to receive the next keycode. Note that some IBM keyboards differ slightly in their output. A jumper is provided on the motherboard to adjust for this problem.

The Serial Interface

The 1616's standard serial protocol is RS232C which is obtained by loading the serial drivers and receivers U6-U9 and linking pins on the serial jumpers. These jumpers may be wired to accommodate differing serial port pinouts. Other communication protocols such as RS422, Appletalk and SDLC may be implemented by mounting a peripheral circuit board containing appropriate logic and drivers onto the serial jumper pins.

The 1616's serial interface is implemented using the Zilog Z8530 Serial Communications Controller (SCC). The dual 4 to 1 data selector U63 is used to generate the SCC's control signals /SCCRD and /SCCWR from the MC68000's bus signals. The SCC has no reset pin and is reset by simultaneously lowering both these signals.

The 3.75 MHz signal from the timing PAL is sufficiently close to the desired 3.6864 MHz for it to be used as the baud rate reference for the SCC. This signal is brought to the serial jumpers for this reason.

The SCC handshake signals are RTS and CTS. If the CTS input is false then the SCC will not transmit. If the RTS output is false then external serial devices should not transmit.

The parallel printer port

The parallel printer port is a minimal Centronics interface. Characters are written to the 8 bit latch U1 (74LS374), and the VIA /CENTSTB signal is pulsed low by software to send the data to the printer.

Centronics type printers send an 'acknowledge' pulse to the host computer when they are ready for new data. This signal, /CENTACK, is buffered and passed to a VIA interrupt input pin.

The printer also transmits a 'busy' signal which is high when the printer is not ready to receive data. This signal is buffered and passed to the VIA CENTBUSY input.

No attempt is made to check for paper out, or other printer error conditions, as printers also invariably send a high 'busy' signal when there is a problem.

The cassette interface

The tape recording format is a simple one: a binary zero is represented by a 200 microsecond cycle and a binary one by a 400 microsecond cycle. To save data on tape, the VIA is programmed to generate a stream of interrupts which provide the timing and the VIA /CLRCASIRQ signal is raised and lowered at interrupt time to generate the appropriate signal to be recorded.

Upon playback the recorded square-edged signals are greatly smoothed. The LM301 op-amp U12 is used as a slope detector to cancel the effects of tape recorder bandwidth limitations. The LM319 comparator retrieves the original signal and produces a TTL compatible level.

The monostable U33 has a timing period of approximately 300 microseconds and is used to decode the pulse-width modulation recording format: if two rising edges from the comparator are separated by less than 300 microseconds (a binary zero) then the monostable output will be

low on the second rising edge; if the separation exceeds 300 microseconds then the monostable output will be high. On each rising edge of the output of U11 during tape reading three things happen:

- The U33 monostable is retriggered, starting the 300 microsecond timer.
- The previous output of the monostable is latched in U32 and presented to the MC68000 via the input port.
- The flip-flop U32 is clocked and /CASIRQ is lowered, interrupting the MC68000. The /ECASRIRQ signal enables these cassette read interrupts. The cassette read interrupt software reads the data bit from the input port and resets the interrupt status by pulsing /CLRCASIRQ low, thus presetting U32.

Because the cassette I/O is software driven it is important that both the reading and writing interrupts be serviced by the MC68000 without any delay. For this reason the /CASIRQ signal is put at a higher priority than any other interrupting device.

During cassette writing /CASIRQ is generated by programming one of the VIA's internal timers to produce a continuous square wave on the PB7 pin. The flip-flop U49 is clocked by this waveform and requests the interrupt if the /ECASWIRQ (enable cassette write interrupt) signal is low.

A bit on U18, the analogue multiplexer port, is used to drive transistor Q2 and thus the relay which software uses to stop and start the cassette motor.

Analogue input/output

The analogue input / output is centred upon the U20 / U21 8 bit digital to analogue converter circuit (DAC). This DAC is used for the following things:

- General purpose analogue output and sound generation. The dual 4 to 1 analogue multiplexer U23 switches the buffered output of the DAC onto one of the four holding capacitors C39-C42. Although the voltage held on these capacitors will decay after a period this technique enables us to produce four analogue signals using one DAC.
- Analogue inputs. The 8 to 1 analogue multiplexer U10 selects an input from amongst the joystick potentiometers and the six general purpose inputs, and presents it to the comparator U11. This comparator compares the selected input with the DAC's output and a software successive approximations routine is used to determine the input voltage level.

Sound generation

Sound output is produced by using the DAC to produce audio waveforms on C40 and C41. These are amplified by the dual audio amplifier U13 and drive loudspeakers. Stereo sound is produced by quickly switching the output of the DAC between the two amplifiers.

The potentiometers RV1 and RV2 control the left and right output signal level and hence also serve as a balance control.

Low-level signals 'LEFTSIG' and 'RIGHTSIG' are available at the speaker connector for connection to audio equipment such as recorders and amplifiers.

The user port

The user port is a 34-way connector which makes available the 1616's power supplies, eight I/O pins from the VIA, six analogue inputs, two analogue outputs and an interrupt input. This port permits the user to implement his own control and interfacing projects.

The expansion connectors

The four 80-way expansion connectors make available various clock signals, the power supplies and all of the MC68000's signals. They are designed to accept add-on cards such as I/O expansion, disk controllers, memory expansion and co-processors.

5
Building a Kit Computer

Why do it yourself?

The major justification for a DIY computer is to demonstrate that most people can learn about a complex subject, provided they have the correct tools. The Applix 1616 is a teaching tool for those people who are not satisfied until they have extended their own abilities, and taught themselves new skills.

Who can build this kit

The 1616 computer has been designed from the very beginning to be as simple to build as possible. Applix have tried to make the circuit as 'clean' and reliable as possible. It is expected that you have constructed electronic kits before and/or have experience in digital electronics at either a hobbyist or professional level. If you are a complete beginner it might be best to try a smaller project first, or to make arrangements with an experienced friend to help you just in case you get into trouble. On the other hand, the youngest Applix enthusiast we know built his system at age fourteen, with only minor fault finding required.

Built and tested

If you feel you are not capable of building the 1616, and you are not interested in the construction workshops, you may wish to return the kit in its original condition and purchase a built and tested 1616, or arrange a refund. We can not accept a return once construction has commenced, if any of the component packs have been opened, or if the kit or cartons have been damaged in any way. Please contact Applix concerning pricing of the built and tested 1616, and the returns procedure. Applix are always happy to arrange to sell built and tested computer systems to individuals and organisations.

Kit building

The 1616 computer is a very complex circuit. In its complete form, it is a powerful personal computer with features similar to most commercial computers.

There are basically two ways to build the 1616. The first way is simply to solder in all the components, plug in all the ICs and turn it on. This may seem at first to be the fastest and easiest way. However if the system does not work first time, debugging the complete circuit can be much harder than debugging one section at a time. It also doesn't provide the constructor with any insight as to how the computer works or how everything fits together.

The second method is called Progressive Assembly and Test (PAT) and is the technique that will be described in this construction manual. PAT involves breaking the construction into a number of small, simple steps, essentially breaking the 1616's complex circuit into a number of smaller ones. As each step is completed, a number of simple checks are done ensuring that that step is working. If something is wrong it is corrected at this stage before proceeding to the next step. The steps are arranged in such a way that each will work only if the ones preceding it work, so you must never go on to the next step until the present one works. By constructing the 1616 in this fashion, at the completion of the last step, everything should work. Also, by breaking the circuit up into a number of smaller ones, it gives a better understanding of how the computer functions.

'Fix It' guarantee

If your built-up 1616 fails to operate and you can not trouble shoot the fault(s), you may wish to take advantage of the Applix 'Fix It' guarantee.

Our technicians will check and repair your 1616 as required, for the flat fee of $150. This fee includes replacement of any necessary components that may have been damaged during or after construction.

Your 1616 must be unmodified and must be constructed using IC sockets to use the 'Fix It' service. If a kit is so badly constructed as to make repair impossible, we reserve the right to return the computer (and the service fee) in the condition received.

The 'Fix It' fee covers only the actual computer; it does not cover your power supply, keyboard, expansion boards etc. It is a 'Fix It' service only, not a construction service. We want to see your 1616 working as much as you do!

Getting started

Before getting started there are a few extras you will have to get. Below is a list of what would be considered the minimum required.

( ) A pair of fine cutters. (Jaycar Cat #TH-1892, Rod Irving Electronics #T12071, Dick Smith Cat #T-3254, Tandy #64-1845 or similar)

( ) A fine tipped soldering iron. (Jaycar #TS-14640, Rod Irving #T12640, Dick Smith Cat #T-1350, Tandy #64-9580 or similar)

( ) A roll of fine solder. (Multicore 60% tin/40% lead with no copper, medium active flux is recommended)

( ) Small screwdrivers etc.

( ) Alligator clips, small pieces of wire etc.

( ) A DC power supply capable of supplying:
+5 volts @ 2.5A, +- 3%
+12 volts @1.5A, +- 6%
-5 volts @250mA, +- 10%
-12 volts @250mA, +- 10%

Note: This is the minimum power supply required. A bigger power supply is recommended to allow for expansion boards, disk drives, etc. Applix has available a suitable switching 'Apple type' supply complete with mating power plug, power connectors and cords. Note that this power supply may not be able to run a hard disk - if planning to run a large hard disk, an IBM style switch mode power supply is required.

( ) Keyboard: An 'IBM XT' compatible detachable keyboard. Any that includes an 'XT' switch should work. The recommended one is an 'IBM AT' style 'XT' keyboard. These are available directly from Applix (as a service to customers), however many other companies also sell suitable keyboards.

( ) Cassette player: A standard mono cassette player of reasonable quality. (Most cassette problems have been due to incorrect components in the 1616 cassette circuitry, or a faulty cassette player.) It must have sockets for 'EAR','MIC' and 'REMOTE'. A tape counter, although not necessary, is an advantage. Contact Applix for recommendations and sourcing. If you plan to move straight to a disk system, you need not bother about the cassette.

( ) A monitor. If you plan to expand your colour video display capabilities, a 'multisync' colour monitor (as used on many IBM clone computers) is suggested as the ideal choice. Although recommended, these are unfortunately also the most expensive style of monitor.

Other monitors that can be used include:

A standard (old fashioned) composite video (single RCA connector) monochrome screen (as used by Apple][, Microbee and other home computers). Cheap, but harder to find now, and can never show colour, nor can it use every video mode available.

'Dual scan' monochrome display intended for use with both IBM colour cards and Hercules monochrome cards for IBM. For monochrome, we suggest the 'dual scan' gives more flexibility, if you plan to alter the video circuits later to use EGA equivalent video. Can show EGA mode video, but not MGR mode.

RGBI (red/green/blue/intensity) type colour screen as used by early IBM PCs for CGA graphics. Like the composite monochrome, this can show only CGA (640 by 200) video. It can not handle EGA or MGR mode video.

( ) Cabling. You will be required to make cables to connect the 1616 to your cassette player, loudspeakers, monitor, and possibly printer, modem, mouse, and custom equipment, as discussed in appendix C.

( ) Test Equipment. The minimum required here is logic probe (Jaycar Cat #QT-2210, Rod Irving #Q11272, Dick Smith cat #Q-1272, Tandy #22-303 or similar). Some cheap logic probes can give very strange (and worrying) results, so we suggest it is false economy to get a poorly designed one.

Multimeter. A cheap multimeter will probably be satisfactory however, as you are not doing any really fine measurements. You really only need to test DC voltages up to 12 volts, and resistance.

A small oscilloscope is an advantage if you encounter problems (a big oscilloscope is heaven) and a frequency counter can also be useful. Most people don't have these, and manage perfectly well without them.

6
Parts List

Before doing anything, check off the parts list against the parts supplied with your kit. If you have purchased a Mini Kit, use this as your shopping list. Do the same for any expansion kits (I/O Kit etc) that you may also have purchased. If you find anything incorrect, or any parts missing, you should contact Applix immediately for the part(s) to be sent or replaced.

Basic kit

The Basic Kit includes all the electronic components (except the dynamic RAM) needed to build up an Applix 1616 motherboard, as a single board computer. It does not however include a case, power supply, keyboard or monitor, as many users prefer to obtain these elsewhere (they are, of course, also available from Applix). The Basic Kit also excludes I/O components (the connectors for serial, parallel and user ports, and their associated components), since some builders will not require all of these.

The kit does not include IC sockets. Applix recommend that sockets be used when constructing the kit, and have a set of suitable quality sockets available. Many builders will however prefer to use machine pin or similar high quality sockets instead. As preferences in socket styles are often strongly held, it was thought better to make the sockets an optional item, even though they are strongly recommended.

Potential builders with well stocked parts boxes, or those who wish to buy their own components, can purchase a Mini Kit. The Mini Kit includes PCB, PALS, EPROMS, 68000, 30 MHz clock oscillator, manuals and schematics, but does not include other components.

Resistors

1	@	22	(red-red-blk)
13	@	68	(blu-gry-blk)
1	@	100	(brn-blk-brn)
2	@	150	(brn-grn-brn)
2	@	330	(org-org-brn)
1	@	470	(yel-pur-brn)
5	@	1k	(brn-blk-red)
11	@	2k2	(red-red-red)
1	@	3k3	(org-org-red)
3	@	4k7	(yel-pur-red)
1	@	8k2	1% (gry-red-blk-brn) (grey-red-red if 5%)
1	@	10k	(brn-blk-org)
2	@	15k	(brn-grn-org)
2	@	18k	(brn-gry-org)
3	@	33k	(org-org-org)
5	@	56k	(grn-blu-org)
2	@	120k	(brn-red-yel)
1	@	220k	(red-red-yel)

Resistor networks

( )	RN2	1k	(102)
( )	RN3	3k3	(332)
( )	RN4	3k3	(332)
( )	RN5	3k3	(332)

( )	RN6	3k3	(332)
( )	RN7	3k3	(332)

Notes: These resistor networks are 10 pin, that is, 9 resistors and one common. RN1 is used in the Centronics printer port and is supplied in the optional I/O Kit. Resistor networks are polarised, with the dot on the package indicating the orientation. Match the dot to the marking on your 1616 printed circuit board.

Capacitors

3	@	56pF	ceramic (56)
1	@	120pF	ceramic (121)
1	@	390pF	ceramic (391)
1	@	.01uF	monolithic (103)
1	@	.1uF	greencap 5% (104)
68	@	.1uF	monolithic (104)
15	@	10uF	tantalum 16 volt (106)
3	@	220uF	electrolytic 16v

The kit may be supplied with either low leakage electrolytic or tantalums capacitors. C3 and C4 (both 10uf tantalum) are not supplied. They are only needed if you wish to use the auxiliary power connector (located close to C3).

Electrolytic capacitors should be PCB mount types, not axial mount. Monolithic capacitors should have a lead spacing of 0.2 inch, to suit the board layout, and ease construction.

Integrated Circuits

( )	U2	74LS244	Tri-state buffer
( )	U3	MGR PAL	replaces Quad 2-Mux with storage
( )	U4	74LS670	4 x 4 register file
( )	U5	16R8	Video PAL, marked VPAL1 U5
( )	U10	4051	8 channel analogue multiplexer
( )	U11	LM319	Dual high speed comparator
( )	U12	LM301	Operational amplifier
( )	U13	LM1877†	Dual 2 watt audio amplifier
( )	U14	MC68000-P10	Motorola 68000 CPU 10MHz
( )	U15	6522 VIA	Versatile interface adaptor
( )	U16	74LS374	Tri-state octal latch
( )	U18	74LS374	Tri-state octal latch
( )	U19	74LS244	Tri-state buffer
( )	U20	74LS374	Tri-state octal latch
( )	U21	DAC0800	8 bit digital to analogue converter
( )	U22	LM324	Quad operational amplifers
( )	U23	4052	Dual 4 channel analogue multiplexer
( )	U24	74LS32	Quad 2 input OR gates
( )	U25	27512/010	1616/OS ROM (high byte)
( )	U26	27512/010	1616/OS ROM (low byte)
( )	U27	MC6845	CRT controller
( )	U28	74LS244	Tri-state buffer
( )	U29	74LS166	8 bit parallel in/serial out shift reg.
( )	U30	74LS244	Tri-state buffer
( )	U31	74LS166	8 bit parallel in/serial out shift reg.
( )	U32	74LS74	Dual D type flip flops with clear
( )	U33	74LS123	Dual retriggerable one shots (Do not use TI brand in this position.)
( )	U34	74LS14	Hex schmitt triggers
( )	U35	74F153	Dual 4 to 1 multiplexers

( )	U36	74F153	Dual 4 to 1 multiplexers
( )	U37	74F153	Dual 4 to 1 multiplexers
( )	U38	74F153	Dual 4 to 1 multiplexers
( )	U39	41256-15	256k x 1 bit dynamic RAM chip
( )	U40	41256-15	256k x 1 bit dynamic RAM chip
( )	U41	41256-15	256k x 1 bit dynamic RAM chip
( )	U42	41256-15	256k x 1 bit dynamic RAM chip
( )	U43	41256-15	256k x 1 bit dynamic RAM chip
( )	U44	41256-15	256k x 1 bit dynamic RAM chip
( )	U45	41256-15	256k x 1 bit dynamic RAM chip
( )	U46	41256-15	256k x 1 bit dynamic RAM chip
( )	U49	74LS74	Dual D type flip flops with clear
( )	U50	74F74	Dual D type flip flops with clear
( )	U51	16R8	Timing PAL, marked TPAL1 U51
( )	U53	74LS174	Hex D flip flops with clear
( )	U54	74LS04	Hex inverters
( )	U55	74LS32	Quad 2 input OR gates
( )	U56	74LS148	Priority encoder
( )	U57	74LS11	Triple 3 input AND gates
( )	U58	74LS08	Quad 2 input AND gates
( )	U59	74LS138	3 to 8 line decoder
( )	U60	74LS74	Dual D type flip flops with clear
( )	U61	74LS05	Open collector hex inverters
( )	U62	74F153	Dual 4 to 1 multiplexers
( )	U64	74LS139	Dual 2 to 4 line decoders
( )	U66	41256-15	256k x 1 bit dynamic RAM chip
( )	U67	41256-15	256k x 1 bit dynamic RAM chip
( )	U68	41256-15	256k x 1 bit dynamic RAM chip
( )	U69	41256-15	256k x 1 bit dynamic RAM chip
( )	U70	41256-15	256k x 1 bit dynamic RAM chip
( )	U71	41256-15	256k x 1 bit dynamic RAM chip
( )	U72	41256-15	256k x 1 bit dynamic RAM chip
( )	U73	41256-15	256k x 1 bit dynamic RAM chip

Notes: U47,48,52,74 are all spare and are not used. U1,6,7,8,9,17 and 63 are used in the serial and Centronics ports and are supplied in the optional I/O Kit. † U13 may be a LM377 or LM378, or LM1877.

U65, the Vsync invert option, accepts a 74LS86 Xor gate (in this case only, cut the track between U65/5 and U65/6 on the underside of the board). This option is provided to assist the stability of some monitors in EGA mode. Setting U27/17 (MA13) high will then invert Vsync. If you don't have a stability problem with your monitor, don't bother making the change, as the board works fine without the 74LS86. Since this option is highly monitor dependent, you should contact Applix for the latest information before installing it.

Miscellaneous

( )	D1	1N914 diode or similar
( )	D2	1N914 diode or similar
( )	Q1	BC337 transistor
( )	Q2	BC327 transistor
( )	RV1	200k trimpot horizontal mount
( )	RV2	200k trimpot horizontal mount
( )	LED1	5mm LED (test indicator, yellow looks nice)
( )	LED2	5mm LED (power on, green looks nice)
( )	SW1	Reset switch C&K 8168A(BE) or similar (RA PCB mount)
( )	SW2	4 way DIL switch

( ) RL1 Relay 211-CD005M or similar (David Reid)
( ) OSC 30.000MHz TTL oscillator
( ) 1616PCB 1616 printed circuit board.

Connectors

( ) Keyboard PCB mount right angle 5 pin female DIN
( ) Speaker PCB mount right angle 5 pin female DIN
( ) Cassette PCB mount right angle 5 pin female DIN
( ) Joystick PCB mount right angle female DB9
( ) Video PCB mount right angle female DB9
( ) Power 5 way keyed DC power connector (Apple clone)
( ) 14 way IDC strip (2x7 pins)
( ) 10 way IDC strip (2x5 pins)
( ) 15 way pin strip (1x15 pins)
( ) Shunts to suit strips (total 7)

Note: The exact IDC strips provided may vary slightly.

Mini kit

This low cost option is suited to experienced builders with a well stocked parts box. Those with experience in sourcing a wide range of components may also prefer to work from the Mini Kit. If you are not experienced in locating a range of components, the Basic Kit is the easier (albeit possibly more expensive) option.

Integrated Circuits

( ) U3 16R8 MGR PAL
( ) U5 16R8 Video PAL, marked VPAL1 U5
( ) U14 MC68000-P10 Motorola 68000 CPU 10MHz
( ) U25 27256/512 1616 ROM (high byte)
( ) U26 27256/512 1616 ROM (low byte)
( ) U51 16R8 Timing PAL, marked TPAL1 U51

Miscellaneous

( ) OSC 30.000MHz TTL oscillator
( ) 1616PCB 1616 printed circuit board.

IC Socket Kit

Sufficient sockets for the entire board, of a quality that has been established as satisfactory for a project of this nature. Those interested in intensive modification may prefer to obtain more expensive and reliable machine pin or similar sockets. I prefer using coloured (blue) sockets, to make the board more interesting visually.

( ) 1 x 8 pin IC socket
( ) 19 x 14 pin IC socket
( ) 33 x 16 pin IC socket
( ) 11 x 20 pin IC socket
( ) 2 x 32 pin IC socket
( ) 3 x 40 pin IC socket
( ) 1 x 64 pin IC socket

I/O Kit

These are the components and connectors for the Centronics parallel printer port, and the two RS232C serial ports. As not every builder requires these immediately, they have been made available separately, rather than increasing the cost of the Basic Kit.

Resistor Networks

( )	RN1	3k3	(332)

Integrated Circuits

( )	U1	74LS374	Tri-state octal latch
( )	U6	1489	RS232C line receiver
( )	U7	1488	RS232C line driver
( )	U8	1489	RS232C line receiver
( )	U9	1488	RS232C line driver
( )	U17	Z8530	Dual serial communications controller
( )	U63	74F153	Dual 4 to 1 multiplexers

Connectors

( )	Centronics	26 way right angle IDC header connector
( )	User I/O	34 way right angle IDC header connector
( )	Serial A	PCB mount right angle male DB9
( )	Serial B	PCB mount right angle male DB9
( )	SJA-0	18 way IDC strip (2x9 pins)
( )	SJA-1	18 way IDC strip (2x9 pins)
( )	SJB-0	18 way IDC strip (2x9 pins)
( )	SJB-1	18 way IDC strip (2x9 pins)
( )	Shunts to suit strips (total 26)	

7
Construction - First Steps

Hints, Tips and Notes

You must always turn the power off and wait about 30 seconds before inserting and/or soldering in any components. It is a good idea to also unplug your power supply from the main board.

Many of the components are polarised. This means that they must go in a particular way. These include the diodes, LEDs, oscillator, transistors, electrolytic capacitors (caps), tantalum caps, resistor networks and the integrated circuits (ICs). Incorrect insertion will probably damage the component, and possibly others connected to it. The PCB's component overlay indicates the orientation of polarised devices.

Although resistors, ceramic caps and monolithic caps are not polarised, it is advisable to orient them all the same way. This makes reading the values easier later on, and also makes the board much neater. Orienting them so you can read them right to left, or from back of board to front, is suggested, since we read values that way.

Integrated Circuits and Sockets

It is highly recommended that the 1616 be constructed using IC sockets. This makes construction and replacement of faulty ICs easier, and it also allows you to take advantage of the Applix 'Fix It' service if worst comes to worst. It is important to use high quality sockets, as intermittent problems caused by using low quality sockets are hard to find, and makes the operation of your computer unreliable.

Orientation of the IC sockets is also important. Most sockets have a notch at one end to indicate pin 1. Match this with the orientation markings on the PCB.

Great care must be taken when handling ICs. The MC68000, EPROMS, VIA, SCC and RAMS are particularly sensitive to static electricity. ICs must be inserted the correct way and in the correct place. Before inserting an IC always check the following:

- You have turned the power off.
- You have the correct IC.
- You are plugging it into the correct socket.
- You are plugging it in the correct way round.
- You haven't bent any pins under or out.
- All of the above again!

It is a good idea to mark pin 1 of each IC with a small drop of liquid paper or to make a little stick-on.

Soldering

The most expensive part of the whole 1616 kit is the printed circuit board. It is also the hardest thing to replace if damaged. The board has been especially designed for manual construction. The solder pads are larger than normal and the board is manufactured with two ounce copper rather than the normal one ounce. It has a solder resist mask on both the component side and the wiring side for easier soldering and a comprehensive component overlay to aid construction.

A fine tipped soldering iron is a must. Any tip larger than about 4mm may damage the board and damage the tracks or pads. An iron temperature of about 350 degrees is recommended. It is advisable to fit a new tip to your iron before starting this project and, if your soldering is a bit rusty, spend a bit of time practicing on a junk board. A few minutes practice could be the difference between success or failure.

Use only high quality fine solder and clean the tip of your iron on a damp sponge regularly. Always apply solder to the hot joint, don't try to take it to the board on the tip of your iron. The solder should flow easily through the plated-through holes; the solder should not flow through on to the component side of the board. Just one more point- keep your soldering neat! Do not use excessive amounts of solder. This only results in an ugly board and increases the risk of solder splashes and bridging. Check your soldering regularly. Try and keep everything neat and don't rush!

Construction - passive components

As mentioned in the introduction, we will construct the 1616 in small steps using the PAT (progressive assembly and test) method. Remember, it is pointless to jump steps, or to proceed to the next step, until it is verified that the current one is working correctly.

Step 1: PCB Checkout

Just before commencing construction, spend a few minutes examining your board. Hold it up to the light and look closely at the back and front for anything odd. This inspection is best done in daylight. In particular, check the holes through the board for correct drilling, and to ensure they are all properly plated through. If you find something wrong, contact Applix before going any further. It is pointless to start soldering a board that is faulty.

With your multimeter switched to resistance, measure at the main power connector (the large one closest to the rear corner of the board) for open circuits between the following points:

( ) COM and +12	( ) COM and -5
( ) COM and -12	( ) COM and +5
( ) +12 and -5	( ) +12 and -12
( ) +12 and +5	( ) -5 and -12
( ) -5 and +5	( ) -12 and +5

PROBLEM: It is highly unlikely that you should find a short between any of these points. If you do, contact Applix for a free board replacement.

Finally, just before we begin construction, gently wipe the board with a clean soft cloth.

Step 2: Resistors, Diodes, Power Connector

Insert and solder into place all the resistors, and 1N914 diodes. Note that the diodes must be oriented with the black band corresponding with the bar shown on the PCB overlay. It is best to do about ten resistors at a time. Just before soldering them in, double check that the resistors are the right values, and that the diodes are oriented the correct way. After soldering, turn over the board and neatly trim off the leads.

Note that R41 must be 8k2 and 1% tolerance. Remember that R1 and R49 do not exist.

Remember that your board may work despite having incorrect resistor values in many areas. However these areas will give incorrect results, when used. The cassette port, joystick port, and analogue inputs and outputs will all work to some extent despite incorrect values. In a project of this complexity, you can expect to solder at least two or three resistors in the wrong positions, if you fail to double check each value. More cautious constructors check the value of resistors with a multimeter prior to inserting them, in case some are faulty, marked incorrectly, and simply to avoid assembly errors.

Resistors

( )	R2	18k	(brn-gry-org)
( )	R3	2k2	(red-red-red)
( )	R4	4k7	(yel-pur-red)
( )	R5	10k	(brn-blk-org)
( )	R6	2k2	(red-red-red)
( )	R7	2k2	(red-red-red)
( )	R8	330	(org-org-brn)
( )	R9	68	(blu-gry-blk)
( )	R10	56k	(grn-blu-org)
( )	R11	56k	(grn-blu-org)
( )	R12	15k	(brn-grn-org)
( )	R13	15k	(brn-grn-org)
( )	R14	150	(brn-grn-brn)
( )	R15	33k	(org-org-org)
( )	R16	1k	(brn-blk-red)
( )	R17	2k2	(red-red-red)
( )	R18	2k2	(red-red-red)
( )	R19	2k2	(red-red-red)
( )	R20	56k	(grn-blu-org)
( )	R21	33k	(org-org-org) (18k may be better here)
( )	R22	2k2	(red-red-red)
( )	R23	470	(yel-pur-brn)
( )	R24	220k	(red-red-yel)
( )	R25	56k	(grn-blu-org)
( )	R26	56k	(grn-blu-org)
( )	R27	1k	(brn-blk-red)
( )	R28	2k2	(red-red-red)
( )	R29	120k	(brn-red-yel)
( )	R30	120k	(brn-red-yel)
( )	R31	2k2	(red-red-red)
( )	R32	1k	(brn-blk-red)
( )	R33	150	(brn-grn-brn)
( )	R34	100	(brn-blk-brn)
( )	R35	1k	(brn-blk-red)
( )	R36	2k2	(red-red-red)
( )	R37	2k2	(red-red-red)
( )	R38	1k	(brn-blk-red)
( )	R39	33k	(org-org-org)
( )	R40	3k3	(org-org-red)
( )	R41	8k2	1% (gry-red-blk-brn) (grey-red-red if 5%)
( )	R42	68	(blu-gry-blk)
( )	R43	68	(blu-gry-blk)
( )	R44	68	(blu-gry-blk)
( )	R45	68	(blu-gry-blk)
( )	R46	68	(blu-gry-blk)
( )	R47	68	(blu-gry-blk)

( )	R48	330	(org-org-brn)
	R49	Does not exist	
( )	R50	68	(blu-gry-blk)
( )	R51	68	(blu-gry-blk)
( )	R52	68	(blu-gry-blk)
( )	R53	68	(blu-gry-blk)
( )	R54	68	(blu-gry-blk)
( )	R55	68	(blu-gry-blk)
( )	R56	4k7	(yel-pur-red)
( )	R57	4k7	(yel-pur-red)
( )	R58	22	(red-red-blk)

Notes: There is no R1 on the board. R49 (2k2) is not required when using 41256 RAM chips, therefore it is no longer on PCB. All resistors are 5% tolerance (the last band is gold) except R41 which should be 1% (the last band is brown).

Alterations: Better cassette loading may be obtained by changing R21 from 33k to 18k (both are supplied).

Solder into place the main power conector.

With your multimeter, check for about 2k resistance between +5v and COM. The rest of the supplies should still be open.

PROBLEM. If there is a problem double check your soldering, that the values of the resistors are correct, that no resistor leads are shorting, and the PCB.

Step 3: IC Sockets, Resistor Networks

Solder in all the IC sockets. It is best to do about 10 at a time. Make sure that you solder the correct sized socket into each position, and that the notch on the IC socket matches the small dot identifying pin 1 on the PCB. It is a good idea to solder in the top left hand pin and the bottom right hand pin of the socket first. Then holding the PCB upright, gently push the socket from the top side of the board whilst touching the soldered pins with your iron to make sure that the socket is flush with the board before soldering it in completely.

I prefer to insert the larger sockets first, since these will not accidently fit into a smaller socket position.

Solder in all the resistor networks (including RN1 if you have the I/O kit). Match the small dot identifying pin 1 of the resistor network with the dot on the PCB. Also, note that RN2 is 1k. Remember that, unlike resistors, resistor networks are polarised, and must be oriented correctly.

Resistor networks

( )	RN2	1k	(102)
( )	RN3	3k3	(332)
( )	RN4	3k3	(332)
( )	RN5	3k3	(332)
( )	RN6	3k3	(332)
( )	RN7	3k3	(332)

Notes: These resistor networks are 10 pin, that is, 9 resistors and one common. RN1 is used in the Centronics printer port and is supplied in the optional I/O Kit. Resistor networks are polarised, with the dot on the package indicating the orientation. Match the dot to the marking on your 1616 printed circuit board.

With your multimeter check for about 2k resistance between +5v and COM. The rest of the supplies should still be open.

There is a modification to the 6522 VIA area required for about 10% of VIAs (Macintoshes have the same modification!) This involves adding a 3K3 resistor to pins 35, 36, 37 and 38 of the VIA, and taking the other end to +5 volts. A resistor pack, RN7, corrects this problem, but may not be shown on the schematic.

PROBLEM. If there is a problem double check the orientation of resistor networks, your soldering and the PCB.

Step 4: LEDs, Power Supply

Solder in LED1 and LED2, noting orientation. LEDs have their anode connected to the positive supply via a current limiting resistor of a few hundred ohms. The anode is the longer leg on the LED. The cathode, indicated by a bar in the schematic, is connected to ground, and is the shorter leg. Although red LEDs are normally supplied, some builders prefer to use green for LED2 (indicates power on), and yellow or orange for LED1, the test indicator.

Connect your power supply and switch it on. The power LED (LED2) should come on. Measure all the supplies for the correct voltages.

PROBLEM. If there is any smoke, explosions etc, turn the power off immediately. Give the PCB a good visual inspection and double check for shorts, solder-bridges, reverse polarisation etc. Replace any damaged parts. If LED2 did not come on check that it is the right way around. Test your power supply. If your power supply needs a load to work (this would be the case if your supply is an Apple type that makes ticking noises) connect a 10 ohm, 5 Watt 'Power On' resistor between +5v and COM (near KBD link). This resistor will be removed later, when sufficient components are plugged in to provide an proper load for the power supply.

Step 5: Monolithic, Ceramic Caps

Solder in all the monolithic and ceramic caps, about ten at a time. Double check the values prior to soldering. It is a good idea to check the power supply resistances (+5v to COM should remain about 2k or at 10 ohms if you have installed a 'Power On' resistor) and power supply voltages every 10 or so caps.

Note that, if a switch mode or Apple style power supply is used, the resistance checks on the +12 volt, -12 volt and -5 volt lines may be wrong *unless* the power supply is disconnected.

Capacitors

( )	C1	10uF	tantalum (106)
( )	C2	10uF	tantalum (106)
( )	C5	.1uF	monolithic (104)
( )	C6	.1uF	monolithic (104)
( )	C7	.1uF	monolithic (104)
( )	C8	.1uF	monolithic (104)
( )	C9	.1uF	monolithic (104)
( )	C10	.1uF	monolithic (104)
( )	C11	10uF	tantalum (106)
( )	C12	.1uF	monolithic (104)
( )	C13	.1uF	monolithic (104)
( )	C14	390pF	ceramic (391)
( )	C15	120pF	ceramic (121)
( )	C16	56pF	ceramic (56)
( )	C17	.1uF	monolithic (104)
( )	C18	.1uF	monolithic (104)
( )	C19	.1uF	monolithic (104)
( )	C20	10uF	tantalum (106)
( )	C21	220uF	electrolytic 16v
( )	C22	220uF	electrolytic 16v

( )	C23	.1uF	monolithic (104)
( )	C24	10uF	tantalum (106)
( )	C25	10uF	tantalum (106)
( )	C26	.1uF	monolithic (104)
( )	C27	220uF	electrolytic 16v
( )	C28	.1uF	monolithic (104)
( )	C29	.1uF	monolithic (104)
( )	C30	.1uF	monolithic (104)
( )	C31	.1uF	monolithic (104)
( )	C32	.1uF	monolithic (104)
( )	C33	.01uF	monolithic (103)
( )	C34	.1uF	monolithic (104)
( )	C35	.1uF	monolithic (104)
( )	C36	.1uF	monolithic (104)
( )	C37	.1uF	monolithic (104)
( )	C38	.1uF	monolithic (104)
( )	C39	.1uF	monolithic (104)
( )	C40	.1uF	monolithic (104)
( )	C41	.1uF	monolithic (104)
( )	C42	.1uF	monolithic (104)
( )	C43	10uF	tantalum (106)
( )	C44	.1uF	monolithic (104)
( )	C45	.1uF	monolithic (104)
( )	C46	.1uF	monolithic (104)
( )	C47	.1uF	monolithic (104)
( )	C48	.1uF	monolithic (104)
( )	C49	.1uF	greencap 5% (104)
( )	C50	10uF	tantalum (106)
( )	C51	10uF	tantalum (106)
( )	C52	.1uF	monolithic (104)
( )	C53	10uF	tantalum (106)
( )	C54	.1uF	monolithic (104)
( )	C55	.1uF	monolithic (104)
( )	C56	.1uF	monolithic (104)
( )	C57	.1uF	monolithic (104)
( )	C58	.1uF	monolithic (104)
( )	C59	.1uF	monolithic (104)
( )	C60	.1uF	monolithic (104)
( )	C61	.1uF	monolithic (104)
( )	C62	.1uF	monolithic (104)
( )	C63	.1uF	monolithic (104)
( )	C64	.1uF	monolithic (104)
( )	C65	.1uF	monolithic (104)
( )	C66	.1uF	monolithic (104)
( )	C67	.1uF	monolithic (104)
( )	C68	10uF	tantalum (106)
( )	C69	10uF	tantalum (106)
( )	C70	.1uF	monolithic (104)
( )	C71	.1uF	monolithic (104)
( )	C72	.1uF	monolithic (104)
( )	C73	.1uF	monolithic (104)
( )	C74	.1uF	monolithic (104)
( )	C75	.1uF	monolithic (104)
( )	C76	.1uF	monolithic (104)
( )	C77	.1uF	monolithic (104)
( )	C78	.1uF	monolithic (104)

( )	C79	.1uF	monolithic (104)
( )	C80	.1uF	monolithic (104)
( )	C81	.1uF	monolithic (104)
( )	C82	.1uF	monolithic (104)
( )	C83	.1uF	monolithic (104)
( )	C84	10uF	tantalum (106)
( )	C85	10uF	tantalum (106)
( )	C86	.1uF	monolithic (104)
( )	C87	.1uF	monolithic (104)
( )	C88	.1uF	monolithic (104)
( )	C89	.1uF	monolithic (104)
( )	C90	.1uF	monolithic (104)
( )	C91	.1uF	monolithic (104)
( )	C92	10uF	tantalum (106)
( )	C93	56pF	ceramic (56)*
( )	C94	56pF	ceramic (56)
( )	C95	.1uF	monolithic (104)
( )	C96	10uF	tantalum (106)* (marked C93 on PCB)

Notes: On revision ‘B’ and ‘C’ boards there are two C93’s. The one located near C94 is the 56pF ceramic. The other one, which is located near C30, is a 10uF tantalum (this will be C96 on later boards). The kit may be supplied with either low leakage electrolytic or tantalums capacitors. C3 and C4 (both 10uf tantalum) are not supplied. They are only needed if you wish to use the auxiliary power connector (located close to C3).

Electrolytic capacitors should be PCB mount types, not axial mount. Monolithic capacitors should have a lead spacing of 0.2 inch, to suit the board layout, and ease construction.

Tantalum and electrolytic capacitors are polarised, and care must be taken to mount them with their leads oriented correctly, as shown on the PCB silk screen overlay.

Note. C49 is a greencap (5%).

PROBLEM. Double check soldering and PCB. If the fault cannot be found cut one leg of each cap (on the top side of board) until the faulty cap is located (this is why we recommended only putting in 10 at a time).

Step 6: Tantalum Caps

Solder in all the tantalum caps, being very careful to orient tantalum or electrolytic capacitors so the positive lead is in the hole marked + on the PC board. It is a good idea to check the power supply resistances (+5v to COM should remain at about 2k or at 10 ohms if you have installed a ‘Power On’ resistor) and power supply voltages every 10 or so caps.

Note. The resistance checks will take a while to settle once the capacitance gets large.

PROBLEM. Double check orientation of caps, your soldering and PCB. If the fault cannot be found cut one leg of each cap (on the top side of board) until the faulty cap is located.

Step 7: Odds and Ends

Solder in the relay (RL1), 200k trimpots RV1, RV2, transistor Q1 (BC337) and Q2 (BC327), being careful to orient them correctly.

Solder in the electrolytic capacitors C21, C22, C27. Note that electrolytic capacitors usually have the - line marked, while tantalum capacitors usually have the + line marked.

Solder in the Reset switch (SW1). It is a good idea to check the power supply resistances (+5v to COM should remain at about 2k or at 10 ohms if you have installed a ‘Power On’ resistor) and power supply voltages.

PROBLEM. Double check orientation of caps and transistors, your soldering and the PCB.

Step 8: Connectors, Links

Next solder in the DIL switch (SW2) and the keyboard, speaker, cassette, joystick and video connectors (also Serial , Centronics and User I/O connectors if you have them). Check the power supply resistances and power supply voltages again.

Note. SW2 should be soldered in so that the 'ON' position is closest U11. The PC board overlay shows the four switches as '0' to '3', while many switches themselves are marked '1' to '4'. You will just have to get used to translating between them in this case.

Also note that the joystick and video connectors are female DB9s (sockets) and the two serial connectors are male DB9s (pins).

Next solder in the strapping blocks as follows:

'MEM' strapping block	2 x 5 pins
'INT LEVEL' strapping block	2 x 7 pins
EPROM strapping block LK3	1 x 3 pins
Keyboard (KB) block	1 x 2 pins
If required,	
Speedup kit LK1 and LK2	1 x 3 pins
	1 x 3 pins
Video strapping block	1 x 1 pins (next to Q1)
	1 x 2 pins

Also, if you have the I/O kit, solder in SJA-0, SJA-1, SJB-0 and SJB-1 (all 2x9 pins).

Note: Do not install any straps on the blocks as yet.

Check the power supply resistances (+5v to COM should remain about 2k or at 10 ohms if you have installed a 'Power On' resistor) and power supply voltages.

PROBLEM. Give the PCB a good visual inspection and double check for shorts, solder-bridges etc.

8
Construction - ICs and Tests

Guide to inserting Integrated Circuits and testing

From here on the format of construction changes. The following six instructions detail how to proceed through each step and should be used with reference to the circuit diagram and PCB component overlay.

POWER OFF — You must always turn the power off and wait about 30 seconds before the insertion or removal of any components. It is a good idea to also unplug your power supply from the main board. 'Power off' may be followed by a few additional instructions to be completed after the board has been turned off.

INSERT — This is always followed by a list of ICs to be inserted. Great care must be taken when handling ICs. The MC68000, EPROMS, VIA, SCC and RAMS are particularly sensitive to static electricity. ICs must be inserted the correct way and in the correct place. Check the following every time you are about to insert an IC:

- You have turned the power off.
- You have the correct IC.
- You are plugging it into the right socket.
- You are plugging it in the right way round.
- You haven't bent any pins under or out.
- All of the above again!

It is a good idea to mark next to pin 1 on top of each IC with a small drop of liquid paper or to a make a little stick-on.

VERIFY — This is a list of tests to complete and tick off or a list of instructions to complete and verify. It is assumed that the constructor has a multimeter, logic probe and/or an oscilloscope for such tests. It is safe to assume that you must turn the power on at the start of 'verify'.

CRO — This instruction can only be successfully completed if you have an oscilloscope (CRO). If you are not using an oscilloscope this instruction can be by-passed.

PROBLEM — This instruction always follows 'Verify'. If you can not verify a test 'Problem' suggests possible causes and/or fixes.

SWITCH — Here you should dial up the correct 'test' on the DIL switch (SW2) and press the reset switch (SW1). The switch settings and their functions are outlined in detail in appendix B. Set all the switches to ON for test 0. Set the first switch off for test 1, and press reset. Set the second switch off, and the first on for test 2, and press reset. Set both the first and second switch off for test 3, and so on. An assembly source listing for the test code is given in appendix D.

Step 9: CPU timing

POWER OFF — Solder in the 30 MHz oscillator (OSC) noting orientation. Match the small dot identifying pin 1 of the oscillator with the dot on the PCB.

INSERT

	Part	Function
U50	74F74	Dual D flip flop with clear (F type)
U51	TPAL1	Timing PAL

VERIFY

( )	U51/1	30MHz
( )	U50/5	15MHz
( )	U51/19	/7.5MHz
( )	U51/18	/3.75MHz
( )	U51/17	/1.875MHz
( )	U51/12	/LPULSE Oscillation
( )	U51/15	/RAS Oscillation
( )	U51/14	/CASU Oscillation
( )	U51/13	/CASL Oscillation
( )	U51/16	/DTACKR High

If your logic probe doesn't give good indications on these tests, it is probably time to buy a decent logic probe, not a single chip DIY monster.

CRO — Force /RAMSEL low (by grounding pins 1 and 2 of 'MEM' strap) and /AS low (by using a thin wire from U14/16 to U14/6).
Grounding /UDS (U14/7) and /LDS (U14/8) should change the waveform of /CASU (U51/14) and /CASL (U51/13) respectively.
Grounding /RAMSEL and /AS should change the shape of /RAS (U51/15) and make /DTACKR (U51/16) pulse low (may be pulsing prior to grounding lines).
Lower speed CROs will not be capable of showing the 30MHz and 15MHz clocks, so you may not be able to check these. The existence of the other clocks proves this portion of the circuit however.
Remove the wires upon completion of this step.

PROBLEM — If something is not working suspect U50, U51 or OSC.

Step 10: Reset circuitry

POWER OFF

INSERT

	Part	Function
U2	74LS244	Tri-state buffer
U60	74LS74	Dual D flip flops
U61	74LS05	Hex inverter, open collector

VERIFY — ( ) /RESET (U14/18) and
( ) /HALT (U14/17) go high a fraction of a second after power on (as soon as capacitor C11 charges up).

PROBLEM — If they don't go high, trace back through U61 and U2.

VERIFY — Close the reset switch (SW1) and check that
( ) /RESET (U14/18) and
( ) /HALT (U14/17) go low.
They should cleanly rise high a fraction of a second after the reset switch (SW1) is released.

PROBLEM — If they don't go high, trace back through U61 and U2.

VERIFY The reset ROM enable circuitry as follows:
( ) Power on, check that /STARTUP (U60/5) is low.
( ) Briefly touch R/W' (U14/9) to ground and /STARTUP (U60/5) should go high, and stay high until you reset.
( ) Press reset (SW1) and check that /STARTUP (U60/5) goes low again.

PROBLEM Suspect U60, U61 or U2.

Step 11: Processor support

POWER OFF

INSERT

	Part	Function
U54	74LS04	Hex inverter
U56	74LS148	Priority encoder
U57	74LS11	Triple 3 input AND
U58	74LS08	Quad 2 input AND

VERIFY

( )	U14/23	/IPL2	High
( )	U14/24	/IPL1	High
( )	U14/25	/IPL0	High

PROBLEM Verify all U56 inputs are high, suspect U56. Pin 10 is not connected, as we only need seven inputs.

VERIFY ( ) U14/10 /DTACK Low
Briefly touch R/W' (U14/9) to ground and check that /DTACK (U14/10) goes high.

PROBLEM Suspect U57, U58.

VERIFY ( ) U57/6 High

PROBLEM Suspect U57/3, U57/4, U57/5

VERIFY

( )	U54/8		Low
( )	U57/8	/VPA	Low

PROBLEM Suspect U54, U57

Step 12: Bus error circuitry

POWER OFF

INSERT

	Part	Function
U24	74LS32	Quad 2 input OR
U33	74LS123	Dual retriggerable one shots

VERIFY ( ) U33/5 Low

PROBLEM Suspect U33, R40 (3K3), C48 (0.1µF mono)

VERIFY ( ) U14/22 /BERR High

PROBLEM Suspect U24

VERIFY ( ) Ground /AS (U14/6) and check that /BERR (U14/22) goes low.

PROBLEM Suspect U33, U24

Step 13: Address decoding

POWER OFF

INSERT

	Part	Function
U55	74LS32	Quad 2 input OR
U59	74LS138	3 to 8 line decode
U64	74LS139	Dual 2 to 4 decode

Strap on 'MEM' joining pins 1 and 2

VERIFY

( ) Touch U14/9 (R/W') to ground or COM (to clear reset ROM enable circuitry).
( ) Connect /AS (U14/6) and A23 (U14/52) to ground or COM and check that /EVPA (U59/7) is low (these two steps must be done in the sequence given, or you may get strange results - thinking about the 68000 bus actions should show you why - you can't just install the wires before power up, so be careful!).

PROBLEM

Check A20 (U59/1), A21 (U59/2) and A22 (U59/3) are all floating (1- 2 volts, usually closer to 2).
/STARTUP (U59/6) is high
U59/4 and U59/5 are low
Suspect U59

Step 14: CPU

POWER OFF

Install the appropriate 'EPROM' links for the size of the EPROMs being inserted, as tabled in appendix B. If you are using 27512, leave pins 1, 2, 31 and 32 of eprom sockets empty. Line up pin 1 of 28 pin eprom to pin 3 or 32 pin socket. Link LK3 to LK3/A (this joins the +5 volt line to pin 30 of 27512).

You could remove the 'power on' load resistor (5 watt), if you fitted it, at this stage, as the components added now provide a sufficient load for a switch mode power supply.

INSERT

	Part	Function
U14	68000	CPU
U25	27256	Eprom (high)
U26	27256	Eprom (low)
U18	74LS374	Octal latch, tri-state
U19	74LS244	Tri-state buffer

Note: Be careful to insert the High ROM (U25) and Low ROM (U26) in the correct locations.
Note: The 68000 is a large chip. You may find that one side of it will 'pop' out of its socket every few months, especially if you plug and unplug other cards frequently. If you have a reliability problem after a few months, check the chip is still seated correctly.

SWITCH

Select test '0' on SW2, by setting all switches ON.

VERIFY

( ) LED1 should be flashing, and the relay (RL1) toggling, about every half scond or so. You may have to press the reset switch (SW1) to start up.

PROBLEM

Hold the reset switch (SW1) closed (difficult, isn't it) and check that pin 22 of U25 and U26 goes low, all address and data lines float.
Suspect address and data lines that are not around 1 or 2 volts when reset is held on.
Check for opens, shorts, interconnections, etc on data and address buses, /AS (U14/6), /UDS (U14/7), /LDS(U14/8), R/W' (U14/9), /DTACK (U14/10), RESET (U14/18), /HALT (U14/17) etc.
Suspect U25, U26, U14, U59
Note that on many chips, it is a bad idea to hold reset on for long periods ('long' is sort of relative in chips). Some Intel chips get hot and break, so don't get into the habit of doing this.

VERIFY

With test '0' still running, check for:

( )	U64/4	/WCENT	high, with negative-going pulses
( )	U64/5	/WDAC	high, with negative-going pulses
( )	U64/6	/WVLATCH	high, with negative-going pulses
( )	U64/7	/WAMUX	high, with negative-going pulses

(You may need a reasonably fancy logic probe to catch all of these - el cheapo ones may not show all.)

POWER OFF

INSERT

	Part	Function
U16	74LS374	Octal latch, tri-state
U20	74LS374	Octal latch, tri-state

(also U1 if I/O kit purchased)

VERIFY

( ) Pins 2, 5, 6, 9, 12, 15, 16 and 19 of U16, U18 and U20 (also U1 if I/O kit is purchased) are all pulsing.

PROBLEM

Suspect data bus, clock signal of latch (pin 11), U16, U18 (74LS374), U20 (also U1 if I/O kit is purchased). Trace back through U64 (74LS139) and U55 (74LS32).

SWITCH

Select test '1' on SW2, by turning the first switch off, and pressing reset. The LED should pulse more slowly.

VERIFY

( ) U64/12 /SCCCS Pulsing low

PROBLEM

Suspect U64 (74LS139). Trace back through U24 (74LS32) and U59 (74LS138).

SWITCH

Select test '2' on SW2, by turning the second switch off, and the first switch on. Press reset.

VERIFY

( ) U64/10 /VIACS Pulsing low

PROBLEM

Suspect U64. Trace back through U24 and U59.

SWITCH

Select test '3' on SW2, by turning switch one and two off. Press reset.

VERIFY

( ) U64/9 /CRTCCS Pulsing low

PROBLEM

Suspect U64. Trace back through U24 and U59.

SWITCH

Select test '4' on SW2, by turning switch three off, and two and one on. Press reset.

VERIFY

( ) U64/11 /RIPORT Pulsing low

PROBLEM

Suspect U64. Trace back through U24 and U59.

Step 15: RAM

POWER OFF

INSERT — U35, U36, U37, U38, U62
All 74F153 Dual 4 to 1 multiplexers (must use F type).

SWITCH — Select test ‘0’ on SW2.

VERIFY — ( ) LED1 should flash and relay (RL1) should be toggling.

PROBLEM — Suspect 35, U36, U37, U38, U62

POWER OFF

INSERT

	Part	Function
U28	74LS244	Tri-state buffer
U30	74LS244	Tri-state buffer

SWITCH — Select test ‘0’ on SW2.

VERIFY — ( ) LED1 should flash and relay (RL1) should be toggling.

PROBLEM — Check pins 1 and 19 of U28 and U30 (negative going enable pulse from pin 3 of U55). Suspect U28, U30. Trace back through U55 and U54.

POWER OFF

INSERT — U39, U40, U41, U42, U43, U44, U45, U46, U66, U67, U68, U69, U70, U71, U72, U73
All 41256, 256k by 1 dynamic RAM

SWITCH — Select test ‘0’ on SW2.

VERIFY — ( ) LED1 should flash and relay (RL1) should be toggling.

PROBLEM — Suspect RAM (U39-U46, U66-U73)

SWITCH — Select test ‘5’ on SW2. Don’t forget to press reset.

VERIFY — ( ) LED1 should be flashing. This test writes a value to the byte at memory location $55AA, then reads it back and echoes it to the DACLATCH (U20). It then increments the value and repeats (about a minute is sufficient testing).

CRO — Check for incrementing byte on DACLATCH (U20) outputs.

PROBLEM — Check for switching signals on A0 - A8, Din, Dout, /RAS, /CAS and /WE on all RAM chips (41256).
For reference purposes, address lines A0 to A8 are pins 5, 6, 7, 9, 10, 11, 12, 13 on RAM chips. Din is pin 2, Dout is pin 14. /RAS is pin 4, /CAS is pin 15, /WE is pin 3.
Check values of R42-R47 and R50-R55 (all 68 ohms termination resistors, used to ease voltage overshoot, and ringing, on rapidly pulsing lines).
Check for switching signal on U62, U35, U36, U37, U38 pins 14.

SWITCH — Select test ‘6’ on SW2.

VERIFY — ( ) This test checks the RAM and slowly toggles LED1 and the relay (RL1). If a failure is detected it flashes LED1 quickly and stops toggling the relay (RL1). It then sits in a loop reading from the bad address. The address of the error is echoed to the AMUXLATCH (U18) (bits 16-19), the VIDLATCH (U16) (bits 8-15) and the DACLATCH (U20) (bits 0-7) (CRO recommended here). About a minute is sufficient testing.

PROBLEM

Carefully inspect the board for solder faults, solder splashes, pins not soldered, chips not correctly socketted.
Check for switching signals on A0 - A8, Din, Dout, /RAS, /CAS and /WE on all RAM chips (41256).
Check values of R42-R47 and R50-R55 (all 68 ohm termination resistors on multiplexed address lines to RAM, and RAS and CAS lines to RAM).
Check for switching signal on U62, U35, U36, U37, U38 pins 14 (/WE and pin 14 signals may not switch after test fails).
Suspect RAM chips, power supply levels and power supply noise.
Suspect U35, U36, U37, U38, U62, especially if using old versions. Note these must be 74F type.
Suspect U51 (TPAL).

Step 16: Video output

POWER OFF

INSERT

	Part	Function
U27	6845	CRT controller

SWITCH

Select test '7' on SW2.

VERIFY

( )	U27/39	HSYNC	15625 Hz horizontal sync pulses
( )	U27/40	VSYNC	50 Hz vertical sync pulses

PROBLEM

Select test '3'. Check that /CRTCCS (U27/25) is pulsing. Check for /1.875 MHz on U27/21. Check R/W' is pulsing on U27/22. Suspect U27

VERIFY

( ) Check that CSYNC (U24/11) is the logical OR of HSYNC (U27/39) and VSYNC (U27/40). ie: High whenever one of HSYNC and VSYNC is high.

PROBLEM

Suspect U24

POWER OFF

Set the VIDEO strap as outlined in appendix B (for IBM style RGBI colour monitor, no strap is required). Connect EN640 (U5/2) to +5v.

INSERT

	Part	Function
U3	MGR PAL	replaces Quad 2 MUX with storage
U4	74LS670	4 by 4 register file
U5	VPAL	Video PAL
U29	74LS166	Shift register 8 bit in, serial out
U31	74LS166	Shift register

SWITCH

Select test '7' on SW2.

VERIFY

( ) Connect your monitor to the video connector. The screen should display different colours, character set, lines and the Applix 1616 logo (the actual display varies from ROM version to version). You may have to adjust the controls on your monitor to obtain a steady image.

PROBLEM

Check that the CRTC (U27) is incrementing the video address bus (pins 4-15, 36-38). Suspect U27.
Check that U29/13 and U31/13 are pulsing. Check U5/1, U29/7 and U31/7 for 15MHz signal.
If using composite video, check for composite video signal at base of Q1 (U61/12).
Suspect U27, U3, U4, U5, U29, U31, video cable, monitor.

VERIFY

( ) Disconnect the wire (EN640 to +5v) above and touch EN640 (U5/2) to COM. The screen display should change from 640 mode to 320 mode. Remove the wire after completion of this step.

PROBLEM Suspect U5, U3

15 MHz ONLY There is a wiring problem on Revision C boards running 15 MHz clocks. On these only, pull Pin 21 of the 6845 (U27) out of its socket, and run it directly to the 1.875 MHz signal from the timing PAL. This line got accidently gated with the DISPEN signal, so as soon as the 6845 drops the DISPEN signal, it turns off its own clock (opps!)

Step 17: VIA

POWER OFF

INSERT

	Part	Function
U15	6522	Versatile Interface Adaptor (VIA)

SWITCH Select test '2' on SW2.

VERIFY ( ) LED1 should be toggling.
This test initialises both ports of the VIA for output and writes incrementing values to them. The CPU reads back the value from the VIA B port and echoes it to the DACLATCH.

( ) All VIA port signals (U15, pins 3 through to17), except PA0 (U15/2), should be switching.

PROBLEM Suspect U15 (6522)

Step 18: VIA interrupts

POWER OFF Insert the VIA interrupt jumper strap on 'INT LEVEL' between pins 11 and 12 (second from the bottom).

INSERT

	Part	Function
U34	74LS14	Hex Schmitt trigger
U49	74LS74	Dual D flip flop with clear
U53	74LS174	Hex D flip flop with clear

SWITCH Select 'run' mode (8 or above) on SW2.
Set the switch to 10 for most 'run' tests. Note that from 1616/OS Version 3.3, run modes '12' to '15' expect their control to come from serial port SA:, not from the keyboard (there will be a message on the display about this). If using Versions prior to this, you can use 'run' modes above 12 (9 or 13 is a good choice, but any above 8 will give a video output).

VERIFY ( ) The Applix 1616 logo, sign on prompts and a flashing cursor should appear on the screen.

PROBLEM If the prompt shows, but the cursor does not flash, check VSYNC (U15/39) for 50Hz. If /VIAIRQ (U15/21) is not pulsing low at 50Hz, suspect VIA (U15).
Trace back through VIA addressing.
If /VIAIRQ (U56/12) is low, check that U56/6, U56/7 and U56/9 are not all permanently high. If they are suspect U56 (74LS148).

Step 19: Keyboard

POWER OFF Connect your keyboard to the keyboard connector.

SWITCH Select 'run' mode (10) (9 or 13 for Version 2 users) on SW2.

VERIFY ( ) The Applix 1616 logo, sign on prompts and a flashing cursor should appear on the screen. Try typing some letters. They should appear on the screen as you type. If nothing happens or you get funny characters, try putting a jumper strap on the 'KBD' link. Reset the board and try again.

PROBLEM Whilst holding down a key, observe the keyboard CLK and DATA signals. They should be pulsing as keycodes come in. Check U53/9 and U49/12 for an inverted keyboard CLK signal. Suspect U34 (74LS14), U49 (74LS74), U53 (74LS174), U54 (74LS04) and your keyboard.

Step 20: D/A conversion

POWER OFF

INSERT

	Part	Function
U21	DAC0800	8 bit digital to analogue converter
U22	LM324	Quad operational amplifiers

SWITCH Select test '0' on SW2, if you have a CRO available.

CRO Check for a 'ramping' waveform on U22/1 and U22/14. Note that video may be 'strange' while in test mode.

SWITCH Select 'run' mode (10) on SW2.

VERIFY ( ) If you do not have an oscilloscope, use the 1616/OS command `syscall .74 .n` (where 'n' is a number between 0 and 255) to write different values to the DAC whilst observing U22/1 with a multimeter. (0 should produce round -2.2 volts, .64 or 40 about -1.1 volts, .128 or 80 about 0 volts, .192 or C0 about +1.1 volts, and 255 or FF about +2.2 volts.)

Step 21: Analogue outputs

SWITCH OFF Connect your speakers to the speaker connector.

INSERT

	Part	Function
U23	4052	Dual 4 channel analogue multiplexer
U13	LM1877	Dual 2 watt audio amplifier

Note: Some PC board overlays incorrectly lists U13 as LM337.

SWITCH Select test '0' on SW2. Note that video may be 'strange' while in test mode.

VERIFY ( ) The speakers should be making a (very) loud 'toot'. If using an analog meter, check for a 5 volt square wave at the speaker connector. Note that if you get too much 'hash' on the analogue sound outputs, additional filtering across the supply rails helps.

CRO U23/5, U23/4, U23/2 and U23/1 should all be switching.

PROBLEM Suspect U23. Trace back though U22 to U21.

Step 22: Analogue inputs

POWER OFF

SWITCH Select run mode (10) on SW2

INSERT

	Part	Function
U10	4051	8 channel analogue multiplexer
U11	LM319	Dual high speed comparator

VERIFY — Select analogue input 7 by using the `syscall .70 7` 1616/OS command. Check that it flips U10 lines from low to high:

( ) U10/9 High
( ) U10/10 High
( ) U10/11 High

( ) Check U10 by varying the input voltage at U10/4 and checking the output voltage on U10/3. Do this by putting various resistor values (in the 10-100K range) between pins 2 and 5 of the joystick port. The same effect could also be obtained by plugging in and varying a joystick. You should be able to obtain voltages between about +2.5 volts and -2.5 volts at the input, U10/4, and therefore similar voltages at the output U10/3.
( ) With a resistor in, measure the voltage in at U10/4 and verify that the same voltage appears out at U10/3 and into U11/5.

PROBLEM — If input voltages at U10/4 are wrong, suspect joystick, cables, and resistors R11 and R13 (better check R10 and R12 also, for Port 6).
If output voltage at U10/3 does not follow U10/4, suspect U10 (especially if you bought cheap parts at a junk sale).

VERIFY — ( ) Use the 1616/OS command `syscall .70 7` to select the joystick port, then `syscall .73` to read and display a value from the joystick port. Note the result. When different resistor values are put in, the displayed result should go up and down with the voltage at U10/3. You should be able to obtain a range of 00 to FF.

PROBLEM — If the A to D conversion is failing, select test '0' to 'ramp' the DAC output. Check that the output of the comparator (U11/12) changes state when the DAC 'ramp' at U22/1 passes through the voltage at U11/5 (as set by the resistor connected to the joystick port). Suspect U11, U10.

Step 23: Cassette/Sound interrupts

POWER OFF — Insert the CASIRQ interrupt jumper strap on 'INT LEVEL' between pins 7 and 8.

INSERT

	Part	Function
U32	74LS74	Dual D flip flops with clear

SWITCH — Select 'run' mode on SW2.

VERIFY — ( ) Test interrupts by observing the 'right' channel at U13/13 with a CRO, and hitting reset (SW1). A digital meter won't show much here. If possible, connect a speaker up to the 'right' channel (across pins 2 and 3 of the speaker connector). You should hear a 'beep' when you reset.

PROBLEM — Check that U49/6 is low and that U56/1 is high. Suspect U49 (74LS74), U56 (74LS148).

Step 24: Cassette I/O

POWER OFF — Connect your cassette to the cassette connector as outlined in Appendix C.

INSERT

	Part	Function
U12	LM301	Operational amplifier

SWITCH — Select 'run' mode on SW2.

VERIFY

( ) Use the 1616/OS command `mdl >testfile`. This creates a test file to save on tape. Insert and position a blank tape to the beginning (off the leader) using the Alt T (obtained by pressing T whilst holding down Alt key) command to toggle the cassette relay.
Type `tsave testfile`. The file should get written to tape.
Rewind the tape using the Alt T command, and reverse switch on cassette, and set it up ready to load the tape back in.
Adjust your volume control to around half and type `tload`.
The previously saved file `testfile` should now load in.

PROBLEM

Fiddle with the playback volume control. Set any tone controls flat. Listen to the recorded signal, it should be a buzzing noise, crisp with no wow or flutter in the headers (the headers are the tone before the raucous noise).

PROBLEM

If you can not play back an apparently correct cassette, carefully check **all** resistor and capacitor values in the cassette area. C14 is 390pF, C15 is 120pF, C16 is 56pF, C49 is 0.1µF 5% greencap. R18 and R22 are 2k2, R21 is 33k (however the cassette loading may be improved by using 18k here), R23 is 470 ohms, R24 is 220k, R25 and R26 are 56k. R41 is 8.2k 1%.

CRO

If you are still having cassette problems and have a oscilloscope do the following:

Record a test waveform onto a blank tape as follows:
Type `mfb 4000 c000 ff`. This fills RAM with 32K of $ff's.
Position your tape ready to save.
Type `syscall .21 4000 8000 .400`. This writes the 32K in one hunk to the tape.
Disconnect the tape player's remote cable, then rewind and play the new tape.
Observe U33/4. It should be low for about 310 microseconds and high for about 103 microseconds. Use any Hitachi, Motorola or Mitsubishi brand for U33 (74LS123), but don't use TI (Texas Instruments).
If it is a long way out, check the values of R41 and C49 and consider replacing U33.
If U33/4 is low for more than 310 microseconds, decrease the value of R41 by putting a resistor in parallel with it (probably in the 22-47k range).
If U33/4 is low for less than 310 microseconds, increase the value of R41.

9
Construction - Serial I/O

The I/O kit

The I/O kit contains all the components to implement the Centronics Port, the Serial Ports and the User I/O port.

Step 25: Centronics Port

POWER OFF

INSERT

	Part	Function
U1	74LS374	Tri-state octal latch

VERIFY ( ) After connecting your printer cable (see Appendix C) from the 1616 to your printer, type `mdl >cent:`. Your printer should display a memory dump.

PROBLEM Check U1, U2 (74LS244), U64 (74LS139). Double check your cable. Make sure your printer is working, plugged in, online etc.

Step 26: Serial Ports

POWER OFF

INSERT

	Part	Function
U6	1489	RS232C line receiver
U7	1488	RS232C line driver
U8	1489	RS232C line receiver
U9	1488	RS232C line driver
U17	Z8530	Serial communications (SCC)
U63	74F153	Dual 4 to 1 multiplexer

Add shunts between the following pins:

SJA-1 pins 1 and 2	SJB-1 pins 1 and 2
SJA-1 pins 3 and 4	SJB-1 pins 3 and 4
SJA-1 pins 5 and 6	SJB-1 pins 5 and 6
SJA-1 pins 7 and 8	SJB-1 pins 7 and 8
SJA-1 pins 17 and 18	SJB-1 pins 17 and 18

SJA-0 pins 1 and 3	SJB-0 pins 1 and 3
SJA-0 pins 5 and 7	SJB-0 pins 5 and 7

This lets the outputs of some lines become the inputs of others, thus allowing full testing of all components except the actual 9 pin D connector. Remember these shunt settings must be changed for normal operation when testing is done.

'INT LEVEL' pins 9 and 10

VERIFY ( ) Type `date >sa:`, then `cio <sa:`
The time and date should appear on the screen. (type Alt C to regain keyboard control).

PROBLEM — Check that SJA-1 pins 5,6,7 and 8 are all low.
Check that SJA-0 pins 5 and 7 are +12v.
Suspect U6, U7 and U17.

VERIFY — ( ) Type `date >sb:`, then `cio <sb:`
The time and date should appear on the screen. (type Alt C to regain keyboard control).

PROBLEM — Check that SJB-1 pins 5,6,7 and 8 are all low.
Check that SJB-0 pins 5 and 7 are +12v.
Suspect U8, U9 and U17.

POWER OFF — Adjust the shunts on SJA-0 and SJB-0 for normal use as outlined in Appendix B. This is important; if you don't set the shunts up as listed, the serial ports will not work with external devices (since they won't actually be connected).

10
Connectors - Appendix A

Connector pinouts

Although most of the connector pinouts are provided in the schematics, a summary, together with their functions, is provided for convenient reference.

Keyboard connector

This is a 180^0 five pin female DIN PCB mount connector, intended for a standard IBM PC keyboard. As there are several different styles of IBM PC keyboard, the keyboard link in the Applix 1616 adapts it to keyboards producing either 2 or 3 start bits.

Pin#	Name	Description
1	CLK	Bidirectional keyboard clock signal.
2	DATA	Unidirectional keyboard data signal.
3	N/C	Not Connected.
4	COM	System common ground.
5	+5v	+5 volt power supply.

Speaker Connector

A 180^0 five pin female DIN PCB mount connector. The left and right signal outputs (pins 4 and 5) are direct outputs from the 4052 (U23) analogue multiplexer. The speaker outputs are the same signal via an LM377 or LM378 (U13) two or four watt stereo amplifier. A potentiometer output level control for each channel is provided on the Applix 1616 board.

Pin#	Name	Description
1	LEFT	Left speaker output.
2	COM	System common ground.
3	RIGHT	Right speaker output.
4	LEFT SIG	Left signal output.
5	RIGHT SIG	Right signal output.

Cassette Connector

A 180^0 five pin female DIN PCB mount connector. The relay switch can also be used to control low voltage, low current external devices.

Pin#	Name	Description
1,3	RELAY	Relay switch. Used for motor control.
2	COM	System common ground.
4	Cassette in	Cassette input signal (to 1616).
5	Cassette out	Cassette output signal (from 1616).

Joystick Connector

This 9 pin female D connector accepts an Apple II style joystick with up to two pushbuttons, or two Apple II paddles.

Pin#	Name	Description
2	+5v	+5 volt power supply.
3	COM	System common ground.
4,6,9	N/C	Not Connected.
5,8	PDL0,PDL1	Hand control inputs. Each of these should be connected to a 150K variable resistor connected to +5 volts.
7,1	JPB0,JPB1	Joystick push buttons. Switches to Ground, normally open.

Serial 'A' Connector

A nine pin male D connector, provides RS232C connection, at standard RS232C voltages, to serial devices (default pinouts only, assumes standard links). Link changes make full access to the SCC available for custom interfaces, such as Appletalk networks.

Pin#	Name	Description
1	COM	System common ground.
2	CTS	Clear To Send (input).
3	RTS	Request To Send (output).
4	RxD	Receive Data (input).
5	TxD	Transmit Data (output).
6	+12v	+12 volt power supply.
7	-12v	-12 volt power supply.
8	DCD	Data Carrier Detect (input).
9	DTR	Data Terminal Ready (output).

Serial 'B' Connector

A 9 pin male D connector, as above (default pinouts only, assumes standard links)

Pin#	Name	Description
1	COM	System common ground.
2	CTS	Clear To Send (input).
3	RTS	Request To Send (output).
4	RxD	Receive Data (input).
5	TxD	Transmit Data (output).
6	+12v	+12 volt power supply.
7	-12v	-12 volt power supply.
8	DCD	Data Carrier Detect (input).
9	DTR	Data Terminal Ready (output).

User I/O Connector

This 34 way 0.1 inch dual in line male connection provides access to six analogue inputs, two analogue outputs, eight digital input-output lines, the Centronics port acknowledge interrupt, and a full range of power lines, for the experimenter.

Pin#	Name	Description
1,2	-5v	-5 volt power supply.
3,4,5,6	+5v	+5 volt power supply.
7,8	+12v	+12 volt power supply.
9,10	-12v	-12 volt power supply.
11,12	COM	System common ground.
13-20	PB7-PB0	VIA input/output signals.
21,22	COM	System common ground.
23	/CENTACK	Centronics Acknowledge. Used on the Centronics Port. Useful for devices connected to the User I/O Port to generate interrupts.
24	N/C	Not Connected.
25,26	COM	System common ground.
27-32	AI0-AI5	analogue Input 0 through 5.
33,34	AO0,AO1	analogue Out 0 and 1.

Centronics connector

Dual in line 0.1 inch 26 way male connector. This should work most Centronics style printers. The interface does not however detect paper out, printer error signals, and does not provide an output signal for selecting a printer (this means genuine IBM printers won't work, but most others will).

Pin#	Name	Description
1	/CENTSTB	Centronics Strobe (output).

3	D0	Data Bit 0
5	D1	Data Bit 1
7	D2	Data Bit 2
9	D3	Data Bit 3
11	D4	Data Bit 4
13	D5	Data Bit 5
15	D6	Data Bit 6
17	D7	Data Bit 7
19	/CENTACK	Centronics Acknowledge (input).
21	CENTBUSY	Centronics Busy (input).
2,4	COM	System common ground.
6,8	COM	System common ground.
10,12	COM	System common ground.
14,16	COM	System common ground.
18,20	COM	System common ground.
22,24	COM	System common ground.
23,25	N/C	Not Connected.
26	N/C	Not Connected.

Expansion Connector Pinout

This provides access to all the 68000 lines, plus power supply, and has provision for external interrupts.

Pin#	Name	Description
1,2	COM	System common ground.
3,4	+5v	+5 volt power supply.
5,6	+12v	+12 volt power supply.
7	-12v	-12 volt power supply.
8	-5v	-5 volt power supply.
9,24	D0-D15	Data Bus (D0-D15). Bidirectional, three state data bus.
25	/AS	Address strobe. This signal indicates that there is valid data on the address bus.
26	/UDS	Upper Data Strobe. This signal indicates that valid data is available on data bus lines D8-D15.
27	/LDS	Lower Data Strobe. This signal indicates that valid data is available on data bus lines D0-D7.
28	R/W’	Read/Write. This signal defines the data bus transfer as a read cycle or as a write cycle. It also works in conjunction with /UDS and /LDS.
29	/DTACK	Data Transfer Acknowledge. This pin is used by expansion boards to determine when another board or the main board has responded.
30	/EXTDTACK	External Data Transfer Acknowledge. This open collector signal is used by expansion boards and indicates that the data transfer is completed. When the processor recognises /EXTDTACK during a read cycle data is latched and the bus cycle terminated. When /EXTDTACK is recognised during a write cycle, the bus cycle is terminated.
31	/BG	Bus Grant. This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.
32	/BGACK	Bus Grant Acknowledge. This input indicates that some other device has become the bus master.
33	/BR	Bus Request. This input is wire ORed with all other devices that could become bus masters. It indicates to the processor that some other device desires to become the bus master.

34	/HALT	Halt. When this bidirectional line is driven by an external device, it will cause the processor to stop at the end of the current bus cycle. When the processor has been halted using this input, all control signals are inactive and all three-state lines are high impedance. When the processor has stopped executing instructions, such as in a double bus fault condition, the /HALT line is driven by the processor to indicate to external devices that the processor has stopped.
35	/RESET	Reset. This bidirectional line acts to reset the processor in response to an external reset signal. An internally generated reset (result of a RESET instruction) causes all external devices to be reset with the internal state of the processor unaffected. A total system reset results from external /HALT and /RESET signals being applied at the same time.
36	/VMA	Valid Memory Address. This output is used to indicate to M6800 devices that there is a valid address on the address bus and the processor is synchronised to enable (E) signal.
37	E	Enable. This signal is the standard enable signal common to all M6800 devices. The period for this output is ten MC68000 clock periods (six low; four high).
38	/VPA	Valid Peripheral Address. This signal indicates to expansion boards that the device or region currently being addressed is a M6800 family device.
39	/BERR	Bus Error. This input informs the processor that there is a problem with the cycle currently being executed.
40	/IPL2	Interrupt Control (/IPL2,/IPL1,/IPL0). These input pins indicate
41	/IPL1	the encoded priority level of the device requesting an interrupt
42	/IPL0	
43	/FC2	These function code outputs indicate the state (user or supervisor)
44	/FC1	and cycle type (program, data, or interrupt acknowledge) cur-
45	/FC0	rently being executed, and are valid whenever /AS is active.
46-68	A23-A1	Address Bus (A23-A1). This 23 bit, unidirectional, three state bus is capable of addressing 8 megawords of data. It provides the address for bus operation during all cycles except interrupts.
69	/STARTUP	Startup. Power on jump signal for use with expansion cards.
70	/EXTVPA	External Valid Peripheral Address. This input indicates that the device or region addressed is a M6800 family device and that the data transfer should be synchronised with the enable (E) signal.
71	/EIRQ0	External Interrupt Request. These decoded inputs are used by
72	/EIRQ1	expansion boards to generate interrupts. The level of the interrupt
73	/EIRQ2	is determined by the setting of the 'INT LEVEL' straps on the
74	/EIRQ3	main board. Only autovectored interrupts are supported on Rev B or C 1616 PCBs.
75,76	+5v	+5 volt power supply.
77,78	COM	System common ground.
79	CLK	Clock. 7.5 MHz system clock (some expansion boards may expect 15 MHz here on speedup systems).
80	30M	30 MHz clock.

11
Switches and Links - Appendix B

SW2 settings

The 4 pole DIP switch to the rear of the board has two functions. It allows testing of the Applix 1616 circuit during construction (SW2, pole 4 **on**), and in the event of service problems. When set for normal operation (SW2, pole 4 off), it sets the RAM disk size, serial operation, and boot sequence. The poles are marked 0 to 3 on the PCB, but are 1 to 4 on most actual switches. Confusing, isn't it?

Test mode

#	4 (3)	3 (2)	2 (1)	1 (0)		Results of test
0	on	on	on	on	Test 0	Latch test: writes incrementing byte to CENTLATCH, DACLATCH, VIDLATCH, AMUXLATCH (except bits 3 and 7), all 4 register file addresses. Toggles LED1 and relay.
1	on	on	on	off	Test 1	Sccrwtest: reads and writes SCC register and toggles LED1.
2	on	on	off	on	Test 2	Viarwtest: writes incrementing byte to VIA B port and A port (excepting bit 0 of A port). Echoes the value read from B port to the DACLATCH, toggles LED1.
3	on	on	off	off	Test 3	Crtcrwtest: reads and writes CRTC register and toggles LED1.
4	on	off	on	on	Test 4	Iporttest: Reads from input port, writes to DACLATCH and toggles LED1.
5	on	off	on	off	Test 5	Memrwtest: writes incrementing byte to address $55aa and echoes read value to DACLATCH. Toggles LED1.
6	on	off	off	on	Test 6	Memtest: writes incrementing word patterns to the entire 512k of memory, checks, and toggles relay and LED1. If failure echo the bad address to DACLATCH (bits 0-7), VIDLATCH (bits 8-15) and AMUXLATCH (bits 16-18). It then loops,toggling LED1, reading the bad address.
7	on	off	off	off	Test 7	Cvidtest: C coded video test. Increments border and screen colours in 320 mode. Displays character set in 640 mode. Draws coloured line test pattern in 320 mode.

Most of these tests can be done without keyboard, video or disk drives. In many computers, major diagnostic tests can only be done if the keyboard, video and disk drives are functioning. Since obviously this is unlikely to be the case during major failures, the diagnostics on most systems are of little use in troubleshooting the circuit. In the Applix 1616, the test functions run from EPROM, and test the CPU, major address decoding, access to most large chips, provide a test pattern written to memory, and also a video display. Results are indicated by flashing an LED, and actuating a relay.

Run mode

#	4 (3)	3 (2)	2 (1)	1 (0)		Video and RAM Disk
8	off	on	on	on	Run 0	Serial, 304k RAM disk.
9	off	on	on	off	Run 1	Serial, 200k RAM disk.
10	off	on	off	on	Run 2	Serial, 104k RAM disk.
11	off	on	off	off	Run 3	Serial, 24k RAM disk.
12	off	off	on	on	Run 4	Colour, 304k RAM disk.
13	off	off	on	off	Run 5	Colour, 200k RAM disk.
14	off	off	off	on	Run 6	Colour, 104k RAM disk.
15	off	off	off	off	Run 7	Colour, 24k RAM disk.

In run mode, SW2 pole 1 and 2 set the default RAM disk size (this can be modified by MRD software under 1616/OS Version 3 and later).

DIP SW2 pole 3 sets the boot order. If pole 3 is open, the system attempts to boot from the hard disk, in this order. /h0, /f0, /f1, /h0, /h1. This was added so hard disk owners didn't have to wait while the (empty) floppies were searched. If SW2 pole 3 is closed, the boot order is the normal /f0, /f1, /h0, /h1.

DIP SW2 pole 3 also sets up the 1616 to be operated from an external serial port via serial port A (but only if no disk controller is present). It formerly (prior to 1616/OS Version 3.2) set the video output to suit either colour or monochrome displays (output is now generally suited to either).

Link settings

As with many computers, a number of links are provided on the Applix 1616 motherboard, to allow changes to the operation of the computer, and for future expansion. These are detailed here.

EPROM links

The Applix can use 512k bit (64k by 8), 1 megabit (128k by 8), or 2 megabit (256k by 8) eproms. The middle position of Link LK3 can be connected to either side A or side B.

Link LK3 to LK3/A for 27512 or 27010 eproms (this connects +5 volts to pin 30, which is +5 volts on 27512 or Vpp on 271024).

If using a 27020 eprom, link Lk3 to Lk3/B, to connect 68000/A18 to eprom A17 line.

Int level links

The internal functions of the Applix 1616 use three of the available interrupt levels. Four levels of interrupt are available for plug in (external) boards.

Pins 1	and	2	/EIRO3	Not normally linked.
Pins 3	and	4	/EIRO2	Not normally linked.
Pins 5	and	6	/EIRO1	Not normally linked.
Pins 7	and	8	/CASIRQ	Linked.
Pins 9	and	10	/SCCIRQ	Linked.
Pins 11	and	12	/VIAIRQ	Linked.
Pins 13	and	14	/EIRO0	Not normally linked.

Keyboard link

The Applix 1616 uses an IBM PC keyboard. The keyboard contains a microcomputer that produces a *keydown* and a *keyup* code for all valid key combinations. This code is transmitted to the Applix 1616 as a serial bit pattern. Various PC keyboards produce a different number of start bits, so provision is made to adapt the Applix 1616 to the pattern required by means of this link.

Linked	=	Strip 3 start bits off keyboard data.
Unlinked	=	Strip 2 start bits off keyboard data.

Memory location links

Additional memory can be added to the Applix 1616 by means of plug in boards. The design calls for a minimum of one megabyte of additional memory, and a maximum of four megabyte, in one megabyte increments. In all cases, the on-board memory is used for video output, and is expected to be at the highest address location occupied by RAM. As on-board memory is shared between the CPU and video, there is insufficient addressing time to run it faster than the present 7.5 MHz clock speed. The 15 MHz internal speed-up kit alters the timing PAL to provide 4 wait states on memory whenever the actual video memory is being accessed, and one wait state at other times. This is about equivalent to 12 to 13 MHz access. External memory could (possibly) run at up to 15 MHz.

Pins 1	and	2	On board RAM starts at $00000000 (normal).
Pins 3	and	4	On board RAM starts at $00100000.
Pins 5	and	6	On board RAM starts at $00200000.
Pins 7	and	8	On board RAM starts at $00300000.
Pins 9	and	10	On board RAM starts at $00400000.

Video links

Monochrome output is provided by mixing the inverted composite sync output from U61, pin 12 (74S05 open collector inverter), and the R, G, B and intensity outputs from U3 (74LS298 quad multiplexer) via weighing resistors R3, R4, R5 and R2. This signal is fed into the base of transistor Q1 (BC337), which is biased by R6, R7 and R8, to produce a composite video signal. The video link between pins 2 and 3 merely connects this signal to pin 7 of the video output connector. If you never intend to use the composite output, Q1, R6, R7 and R8 may be omitted during construction.

The video links are a set of three links located directly above the video PAL, U5. They are numbered as follows:

```
        1 o   o 2
  o           o 3
  DE
```

Monochrome mode
Pins 2 and 3 linked. All others unlinked.

Colour mode
No links connected.

Speedup kit links

Only if you have the optional 15 MHz speedup kit (15 MHz PAL, 16 MHz 68000 P16).

Cut LK1/A, and link LK1 to LK1/B
cut LK2/A and link LK2 to LK2/B.

Serial links

U17, the 8530 serial communications controller (SCC) includes two identical serial ports capable of a variety of different protocols. The SJA1 and SJB1 links normally connect the serial lines to 1488 and 1489 TTL to RS232 voltage level converters. However, you can use the SJA1 and SJB1 connectors to connect the SCC chip to different input output converters. Possible variations would include Appletalk networks, or external clocks. The other set of connectors, SJA0 and SJB0 simply connect the 1488 and 1489 level converters to the external 9 pin serial connectors. You should note that provision is also made for supplying +12v and -12v power to these connectors.

SJA-0 links: (normal)

Pins 1	and	2	Linked.
Pins 3	and	4	Linked.
Pins 5	and	6	Linked.
Pins 7	and	8	Linked.
Pins 9	and	10	Linked.
Pins 11	and	12	Linked.
Pin	13	Not normally connected.	
Pin	14	Not normally connected.	
Pin	15	Not normally connected.	
Pin	16	Not normally connected.	
Pin	17	Not normally connected.	
Pin	18	Not normally connected.	

Note: Pins 16, 17 or 18 (all power supplies) should never be linked together.

SJA-1 links: (normal)

Pins 1	and	2	Linked.
Pins 3	and	4	Linked.
Pins 5	and	6	Linked.
Pins 7	and	8	Linked.
Pins 9	and	10	Linked.
Pins 11	and	12	Linked.
Pin	13	Not normally connected.	
Pin	14	Not normally connected.	
Pin	15	Not normally connected.	
Pin	16	Not normally connected.	
Pins 17	and	18	Linked.

SJB-0 links: (normal)

Pins 1	and	2	Linked.
Pins 3	and	4	Linked.
Pins 5	and	6	Linked.
Pins 7	and	8	Linked.
Pins 9	and	10	Linked.
Pins 11	and	12	Linked.
Pin	13	Not normally connected.	
Pin	14	Not normally connected.	
Pin	15	Not normally connected.	
Pin	16	Not normally connected.	
Pin	17	Not normally connected.	
Pin	18	Not normally connected.	

Note: Pins 16, 17 or 18 (all power supplies) should never be linked together.

SJB-1 links: (normal)

Pins 1	and	2	Linked.
Pins 3	and	4	Linked.
Pins 5	and	6	Linked.
Pins 7	and	8	Linked.
Pins 9	and	10	Linked.
Pins 11	and	12	Linked.
Pin	13	Not normally connected.	
Pin	14	Not normally connected.	
Pin	15	Not normally connected.	
Pin	16	Not normally connected.	
Pins 17	and	18	Linked.

12
Cabling - Appendix C

Where to get cables

Applix can arrange to make up cables for you, and are usually able to sell the connectors required. Connectors can also be obtained from most hobby electronics shops, if you would prefer to make your own. Making cables is generally fairly tedious, so full advantage should be taken of any shortcuts available. In particular, use insulation displacement connectors (IDC) on the printer cable. Ensure that your monitor comes complete with cable. You can often find DIN audio cables that can easily be cut and adapted for Applix 1616 cassette use.

Cassette connection

The Applix 1616 can be connected to a standard cassette recorder, and is designed to operate the remote control of such a recorder. Components required for the cable are a 5 pin 180^0 male DIN connector for the Applix end. The cassette end depends upon the brand of cassette, but usually requires two 3.5mm plugs for *microphone* and *earphone*, and a 2.5mm plug for the remote control socket.

Cassette cable

1616 connector:		Cassette player connector:
Pin# 1	------	First side of cassette remote jack.
Pin# 3	------	Second side of cassette remote jack.
Pin# 4	------	Signal. Inner of cassette EAR jack.
Pin# 2	------	Ground. Outer of cassette EAR jack.
Pin# 5	------	Signal. Inner of cassette MIC jack.
Pin# 2	------	Ground. Outer of cassette MIC jack.

Printer connection

The IBM style Centronics parallel printer cable uses 36 wire round cable. When the Applix was designed, such cables were rather expensive to buy. If making your own cable, a tedious amount of soldering is needed to make a round style cable. The Applix is designed to use 26 way flat cable, and insulation displacement connectors (IDC). You need a 26 way female IDC at the Applix end, and a 36 way Centronics male IDC connector at the printer end. The components for this style of cable can be obtained at most hobby electronic shops. The cable length should be approximately 2 metres, however up to 4 metres can be usually be used. Cables longer than 4 metres should be avoided, as a parallel interface is not suited to such lengths (a serial printer and cable should be used instead).

Centronics cable

1616 connector:		Printer connector:	
Pin# 1	------	Pin# 1	/STROBE
Pin# 2	------	Pin# 19	Ground
Pin# 3	------	Pin# 2	data 0
Pin# 4	------	Pin# 20	Ground
Pin# 5	------	Pin# 3	data 1
Pin# 6	------	Pin# 21	Ground
Pin# 7	------	Pin# 4	data 2
Pin# 8	------	Pin# 22	Ground

Pin# 9	------	Pin# 5	data 3
Pin# 10	------	Pin# 23	Ground
Pin# 11	------	Pin# 6	data 4
Pin# 12	------	Pin# 24	Ground
Pin# 13	------	Pin# 7	data 5
Pin# 14	------	Pin# 25	Ground
Pin# 15	------	Pin# 8	data 6
Pin# 16	------	Pin# 26	Ground
Pin# 17	------	Pin# 9	data 7
Pin# 18	------	Pin# 27	Ground
Pin# 19	------	Pin# 10	/ACKNLG
Pin# 20	------	Pin# 28	Ground
Pin# 21	------	Pin# 11	BUSY
Pin# 22	------	Pin# 29	Ground
Pin# 23	N/C	Pin# 12	High on printer Paper Empty
Pin# 24	------	Pin# 30	Ground
Pin# 25	N/C	Pin# 13	+5v on some printers
Pin# 26	N/C	Pin# 31	Pull low to reset some printers

Serial connection

Serial connections are often a problem, due to the large number of possible permutations of cabling and serial parameters. There is, for example, no mandatory RS232 connector, nor any standard pin connection (although the 25 pin D connection used by IBM is widely imitated). It is impossible to cover these in any reasonable length text (see the various books on RS232 and serial communications by Joe Campbell for background and detailed information).

You will need a 9 pin female D connector for the Applix end of the cable, and preferably a 6 (or more) wire cable. If the other end is to be connected to an IBM style computer you will need a 25 pin female D plug for PC, XT and PS/2 models, or a 9 pin female D plug for AT models. IBM 25 pin connections are 1 frame ground, 2 TxD, 3 RxD, 4 RTS, 5 CTS, 6 DSR, 7 signal ground, 8 DCD, 20 DTR, 22 RI. IBM AT 9 pin connections are 1 DCD, 2 RxD, 3 TxD, 4 DTR, 5 signal ground, 6 DSR, 7 RTS, 8 CTS and 9 RI. Please note that the Applix 1616 is not identical to the IBM 9 pin connector.

Applix to IBM PC-XT 25 pin cable

Pin# 1	Ground	------	Pin# 7	Signal ground
Pin# 2	CTS	------	Pin# 8	DCD
Pin# 3	RTS Jump to #2			
Pin# 4	RxD	------	Pin# 2	TxD
Pin# 5	TxD	------	Pin# 3	RxD
Pin# 6	+12v	N/C		
Pin# 7	-12v	N/C		
			Pin# 4	RTS Jump to #5
Pin# 8	DCD	------	Pin# 5	CTS
Pin# 9	DTR	------	Pin# 6	DSR
		N/C	Pin# 20	DTR

Applix to IBM 9 pin cable

Pin# 1	Ground	------	Pin# 5	Signal ground
Pin# 2	CTS	------	Pin# 1	DCD
Pin# 3	RTS Jump to #2			
Pin# 4	RxD	------	Pin# 3	TxD
Pin# 5	TxD	------	Pin# 2	RxD
Pin# 6	+12v	N/C		
Pin# 7	-12v	N/C		

			Pin# 7	RTS Jump to #8
Pin# 8	DCD	------	Pin# 8	CTS
Pin# 9	DTR	------	Pin# 6	DSR
		N/C	Pin# 4	DTR

Applix to typical serial printer 25 pin connector

Pin# 1	Ground	------	Pin# 7
Pin# 2	CTS	------	Pin# 5
Pin# 3	RTS	------	Pin# 4
Pin# 4	RxD	------	Pin# 3
Pin# 5	TxD	------	Pin# 2
Pin# 6	+12v	N/C	
Pin# 7	-12v	N/C	
Pin# 8	DCD	------	Pin# 8
Pin# 9	DTR	------	Pin# 20

Video Cable:

The video connector is identical to that used in the IBM PC XT range for connection of an IBM CGA video card to an IBM RGBI colour monitor. Monitors intended for use with original IBM systems should plug straight in.

Monochrome: (video link set, see appendix B)

Pin# 7	------	Inner connector of video plug (RCA normally)
Pin# 1	------	Outer connector of video plug (RCA normally)

Colour: (video link set, see appendix B)

Pin# 1	------	Ground
Pin# 2	------	Ground
Pin# 3	------	Red
Pin# 4	------	Green
Pin# 5	------	Blue
Pin# 6	------	Intensity
Pin# 7	------	N/C
Pin# 8	------	Horizontal Sync
Pin# 9	------	Vertical Sync

13
Eprom Startup Code - Appendix D

```
|
| Applix 1616/OS ROM assembler startup code
|
| Copyright (C) 1987 by Applix pty limited
|
| 28 Jan 1987.  Programmer: Andrew K.P. Morton
|
BERR_VEC    =           8           | Bus error vector
ROMSTART    =           0x500000    | Start of ROMs
CENTLATCH   =           0x600001    | Centronics latch
DACLATCH    =           0x600081    | DAC latch
VIDLATCH    =           0x600101    | Video latch
AMUXLATCH   =           0x600181    | Analog multiplexer latch
PAL0        =           0x600000    | Pallette entries
PAL1        =           0x600020
PAL2        =           0x600040
PAL3        =           0x600060
SCCBASE     =           0x700000    | SCC start
IPORT       =           0x700081    | Input port
VIABASE     =           0x700100    | VIA start
CRTCBASE    =           0x700180    | CRTC start
SCCBC       =           SCCBASE+0   | SCC B control
SCCBD       =           SCCBASE+2   | SCC B data
SCCAC       =           SCCBASE+4   | SCC A control
SCCAD       =           SCCBASE+6   | SCC A data
CRTCADDR    =           CRTCBASE    | CRTC address register
CRTCDATA    =           CRTCBASE+2  | CRTC data register
V_BREG      =           VIABASE     | Port B I/O register */
V_AREG      =           VIABASE+2   | Port A I/O register */
V_DDRB      =           VIABASE+4   | Port B DDR   */
V_DDRA      =           VIABASE+6   | Port A DDR   */

|
| Start of code
|
            .text
            .globl      _main,__coldbo,__warmbo,_obramst
initsp:     .long       0x300       | Initial SP
            .long       __coldbo    | Entry point
__coldbo:
__warmbo:
            movb        d0,ROMSTART | Bank out ROMs
            movl        initsp,a7   | Set up Stack pointer
                                    |(non-reset entry)
            reset
dofn:       movb        IPORT,d0    | If switch 3 open
            bmi         toc         | set things up and branch to C
            movb        d0,DACLATCH | Echo it
            lsrb        #4,d0       | Ignore bottom 4 bits
            andl        #7,d0       | Only use 3 bits
            asll        #2,d0
            movl        #jtab,a0
            movl        0(a0,d0:w),a0 | Get code address
            jmp         (a0)        | Go to it
```

```
jtab:       .long         latchtest     | 0: Test analog, video, dac,
cent, pallette
            .long         sccrwtest     | 1: SCC
            .long         viarwtest     | 2: VIA
            .long         crtcrwtest    | 3: CRTC
            .long         iporttest     | 4: IPORT
            .long         memrwtest     | 5: Read / write memory
            .long         memtest       | 6: Memory test
            .long         vidtest       | 7: Video test
jtabe:

| Write incrementing pattern to latches, pallette

latchtest:  clrb          d3            | LED / relay bits
lt3:        movl          #30000,d1     | Loop counter

lt2:        movb          d0,CENTLATCH
            movb          d0,DACLATCH
            movb          d0,VIDLATCH

            movb          d0,d2
            andb          #0x77,d2      | Preserve LED & relay bits
            orb           d3,d2
            movb          d2,AMUXLATCH

            movb          d0,PAL0
            movb          d0,PAL1
            movb          d0,PAL2
            movb          d0,PAL3
            addqb         #1,d0
            subql         #1,d1
            bne           lt2
            eorb          #0x88,d3
            bra           lt3

| Read/write SCC

sccrwtest:  movl          #SCCBASE,a0
            bra           rwtest

| Read/write VIA

viarwtest:
            movb          #0xfe,V_DDRA  | Port A has one input bit.
            movb          #0xff,V_DDRB
vrwtest:    clrb          d2
vrwtest3:   movl          #100000,d1
vrwtest2:   movb          d0,V_AREG
            movb          d0,V_BREG
            movb          V_BREG,DACLATCH | Echo the read back byte
            addqb         #1,d0
            subql         #1,d1
            bnes          vrwtest2
            eorb          #0x8,d2
            movb          d2,AMUXLATCH
            bras          vrwtest3

| Read/write CRTC

crtcrwtest: movl          #CRTCBASE,a0
            bra           rwtest

| Read from Input Port

iporttest:  movl          #IPORT,a0
            bra           rtest
| Read / write memory

memrwtest:  movl          #0x55aa,a0
            bra           rwtest

| General input port test: read from port and write to Centronics latch.
| Input port address in A0.
```

```
rtest:      clrb          d1
rtest3:     movl          #100000,d0
rtest2:     movb          (a0),DACLATCH
            subql         #1,d0
            bnes          rtest2
            eorb          #0x8,d1
            movb          d1,AMUXLATCH  | Flash LED
            bras          rtest3

| General output port test: increment latch contents.  Latch address in A0

wtest:      clrb          d2
wtest3:     movl          #100000,d1
wtest2:     movb          d0,(a0)
            addqb         #1,d0
            subql         #1,d1
            bnes          wtest2
            eorb          #0x8,d2
            movb          d2,AMUXLATCH
            bras          wtest3

|General read/write test.  Write a value to port,
|read it back, echo to centronics port,
|increment value and loop

rwtest:     clrb          d2
rwtest3:    movl          #100000,d1
rwtest2:    movb          d0,(a0)
            movb          (a0),DACLATCH
            addqb         #1,d0
            subql         #1,d1
            bnes          rwtest2
            eorb          #0x8,d2
            movb          d2,AMUXLATCH
            bras          rwtest3

| Do memory test (this simplistic test will shortly be replaced)

memtest:    clrw          d0
            clrw          d1
            clrw          d2            | Start value

memtest2:   movw          d2,d0
            movl          #0,a0
fillmem:    movw          d0,(a0)+
            addqw#1,d0
            cmpl          #0x80000,a0
            bnes          fillmem

            movw          d2,d0
            movl          #0,a0
checkmem:   cmpw          (a0)+,d0
            bnes          badmem
            addqw#1,d0
            cmpl          #0x80000,a0
            bnes          checkmem

            addw          #0x101,d2     | New pattern
            eorb          #0x88,d1
            movb          d1,AMUXLATCH
            bra           memtest2
```

```
badmem:     subql       #2,a0          | Point to bad word
            movl        a0,d0
            movb        d0,DACLATCH    | Low byte of address
            lsrl        #8,d0
            movb        d0,VIDLATCH    | Next byte of address
            lsrl        #8,d0
readbad2:   movb        d0,AMUXLATCH   | Bits 16-23
            movl        #30000,d1
readbad:    movw        (a0),d2
            subql       #1,d1
            bnes        readbad
            eorb        #8,d0
            bras        readbad2

|
| Do Video test: set up stack and jump to C code
|
            .globl      _cvidtes
vidtest:    movl        #0x40000,a7    | Middle of RAM
            clrl        _obramst
            jmp         _cvidtes
```

Index

Table of Contents