The development of a distributed interfacing system

Gray, Peter

The Development of a Distributed Interfacing System

by

Peter Gray, BSc.

A Thesis submitted in accordance with the regulation for the degree of Master of Science in the Department of Applied Physics and Electronics at the University of Durham.

May 1986

ABSTRACT

The Thesis submitted describes the origins and development of an industrial distributed interfacing system. The component modules of the system are described individually and sample flowcharts and software listings are provided.

ACKNOWLEDGEMENTS

The author would like to acknowledge the assistance and support for this project given by the staff of the University of Durham, especially Professor G.R. Roberts (now at Oxford University) and Professor T. Spracklen (now at Aberdeen University), and also by the management and staff of Glaxo Pharmaceuticals (referred to herein as 'the Company'). Special thanks are due to Messrs W.H. Presley and R.A. Pearson of the Company's Management Services Department for approving the project development and granting permission to submit this Thesis, to my immediate superiors Messrs J. Howe and J. Prytherick, to Messrs H.G. Myatt and L.Rodrigues for checking production drawings, and to Mr R.W. Atkinson, who developed professional working systems from the test software described in this document. My thanks are also due to Messrs J.M. Wood and S.M. Wood of Labcon Ltd. (referred to herein as 'the Contractors') and to Messrs I. Robertson and J. Williams for prototype layout and much stimulating discussion, and also to Mr R.J. Hickson (now with R.J. Designs, Shildon) for producing the CAD artworks for production boards.

CONTENTS

Chapter	1	Introduction	1
"	2	The Pilot Scheme	6
"	3	The Expanded System	23
"	4	Memory Expansion	53
"	5	Communications Interfaces	63
"	6	Serial Port Interfaces	86
"	7	The Analogue Board	104
"	8	Running MCS BASIC-52 On the System	114
"	9	Project Review	127
References			132
Appendix	1	Crystals/Oscillator Components	134
"	2	Pilot Scheme Data	147
"	3	The 8751 Serial Port	158
"	4	HPIB Overview	163
"	5	Features of the 8748	168
"	6	BCD Board (Testing/Firmware)	169
"	7	A Sample HPIB-RS232 Converter	173
"	8	The 8052-BASIC CPU	176
"	9	Serial/UART Board Functions & Options	182
"	10	Power Supplies	187
"	11	Details of RAM Memory Devices	189

LIST OF PLATES AND ILLUSTRATIONS

Figure	2.1	Pilot Scheme - Schematic	7
"	2.2	Power Supplies	10
"	2.3	Schematic Diagram of a Node	11
"	2.4	The NE567 Configured as a PLL	14
"	2.5	" " " " VCO	17
"	2.6	Circuit Diagram of a Node	18
"	3.1	Backplane Connector Pin-out	26
"	3.2	8751 Pin-out	28
"	3.3	Timing Diagram for External Memory Access	30
"	3.4	CPU Board Circuit Diagram	32
"	3.5	DVI Board Circuit Diagram	40
"	3.6	Watchdog Board Circuit Diagram	45
"	4.1	RAM Board Circuit Diagram	58
"	5.1	HPIB Board Circuit Diagram	65
"	5.2	BCD Board Circuit Diagram	72
"	5.3	UART Board Circuit Diagram	76
"	5.4	Schematic Diag. Showing HPIB Board Testing	80
"	6.1	Serial Board Circuit Diagram	90
"	6.2	R.F. Board Circuit Diagram	95
"	6.3	Schematic Diag. Showing R.F. Board Testing	101
"	7.1	Analogue Board Circuit Diagram	109
"	7.2	Measured ADC Response	110
"	7.3	Test Rig for ADC Evaluation	111
"	8.1	Port 1 Monitor Breadboard & Audio Monitor	124

Figure	A1.1	CPU Oscillator Tests	135
"	A1.2	CPU Clock Waveforms for 3 Selected Xtals	136
"	A2.1	VCO Freq. Variation with Mod. Voltage	148
"	A2.2	Serial Data Waveforms	150
"	A2.3	Flowchart of Polling Firmware	154
"	A2.4	" " Monitoring Node Firmware	155
"	A2.5	" " Subroutine "NOUT"	156
"	A2.6	" " Subroutines "INCH" & "OUCH"	157
"	A4.1	HPIB Schematic	164
"	A6.1	Schematic of 8748/8751 Interaction	170
"	A6.2	" " 8748/8243 Firmware	171
"	A7.1	" " HPIB to RS232 Converter	174
"	A8.1	" " a Typical BASIC-52 System	177
"	A9.1	Current Loop Configurations	183
"	A9.2	UART Initialization Sequence	184
Plate	3.1	The Sub-Miniature Rack	24
"	3.2	The DVI Board	24
"	5.1	The HPIB "	70
"	5.2	The BCD "	70
"	6.1	The Serial Board	87
"	6.2	The R.F. Board	87
"	7.1	The Analogue Board	107
"	8.1	Alternative CPU Chips	116
"	8.2	The 'BASIC' System Configuration	116
"	8.3	MCS BASIC-52 Running On the System	119
"	8.4	The Port 1 Monitor Breadboard	119

CHAPTER 1 - INTRODUCTION

As the world's population becomes increasingly dependent on manufactured goods, so the trend towards industrial automation in manufacturing industry becomes more dominant, and the demand for plant monitoring and control equipment increases. The introduction in the 1960's of minicomputers such as the D.E.C. PDP-8 into industrial applications greatly extended the scope for monitoring and control of industrial plant. For the first time a flexible controller, for example, could be realized; the hard-wired dedicated machines, often based on wired core storage technology, rapidly fell into disfavour. Purpose-built electronic devices began to be displaced by general purpose computer based units; traditional analog equipment was often superseded by digital hardware. The introduction of microprocessors in the 1970's, with the corresponding increase in the performance/cost figure, increased this trend towards computer-based line monitoring/control equipment.

The author, who has worked since the 1960's on monitoring/control projects, currently works for a large international company, dealing with problems relating to the various hardware/software aspects of the monitoring and control of industrial plant in the Company's U.K. sites. The Company, which has a heavy financial investment in both "mainframe" and microcomputers, has a logical policy of single source supply from a major American manufacturer. This policy, as well as being financially sensible as regards bulk purchase discounts etc., also has the advantage of avoiding the familiar pitfalls of the multi-vendor situation. Thus, when considering purchasing monitoring/control equipment to link, for example, sensors on a production line to a microcomputer, the natural first choice of supply was the approved computer manufacturer mentioned above, who is also a major manufacturer of laboratory instrumentation equipment.

As an aid to this general discussion on line monitoring, (for brevity we will adopt the term "line monitoring" as a convenient - though rather glib - phrase to reference all areas of the fields of monitoring and control of industrial mechanization) it is perhaps useful at this stage to catalogue the various factors relating to the choice of monitoring equipment. These are as follows:-

* Cost
* Single supply source
* Software/firmware programmable
* Versatile range of input/output options
* Good noise immunity
* Networkable
* Easily interfaced to computers
* Compact, rugged construction
* Ease of maintenance

Cost, of course, is often the overriding factor that governs the choice of any equipment to be purchased. If, as in this case, the equipment may be duplicated many times in different sites, the total capital expenditure over a period of time may be very large. Conversely, for small projects with tight budgets, the price/performance ratio for the equipment to be purchased will frequently determine the scope of the project, and may tip the balance between success or failure. Another factor which determines the successful outcome is the maintainability of the equipment - which can become a major problem in multi-vendor situations, as indicated earlier.

Digital or analogue signals that the monitoring equipment may need to process can be of many different types. For example, analogue signals may take the form of either currents in the range 4-20 milliamps or voltages in the range 0-10 volts. These are two commonly used conventions, although the analogue signals may not conform to either - e.g. thermocouple outputs. A similar situation also arises when digital signals are considered.

A common method of digital signalling (for example, the output of an industrial optical sensor) is to interrupt a closed current loop system; switching devices commonly encountered may be new technology solid-state devices, such as Hall Effect switches, or more traditional electrical hardware - e.g. relays or reed switches. Ideally, as well as supporting digital switching signals conforming to the standard industrial protocols, equipment should also have the ability to communicate with (possibly) remote computers or microprocessors. To achieve this communication, the equipment should be capable of supporting a number of interface protocols, such as RS232C, IEEE488 (HPIB), 20mA current loop and (preferably) a networking protocol - which would allow a number of monitoring "nodes" to be networked with a host computer.

For maximum versatility of application, monitoring systems should be software or "firmware" based (the latter term is generally used in connection with EPROM - as opposed to RAM - storage). This allows, possibly, the internal program to be replaced by another program, allowing the equipment to perform a different task. Another possibility is that the user may wish to vary program parameters, while keeping the basic program task the same. Such an occasion arises when a device is required to scan a different (or extra) channel or sequence of channels.

Perhaps the greatest single source of problems concerning purchased equipment is the possibility of error due to it's lack of resilience to the stresses imposed on it by the working environment - the major sources of stress frequently being mechanical vibration and electrical noise. "Noise" invariably takes the form of large mains born "spikes" (often, say, 40dB in excess of the signal levels inside the equipment), although electromagnetic induction can also cause problems. Manufacturers equipment, often demonstrated by sales staff in an office environment, can sometimes prove

disastrous in performance when sited on an industrial production line, next to heavy electrical machines. Mechanical stress can take the form of prolonged vibration, or sudden, perhaps violent, jolts. Thus equipment must be rugged, compact and generally mechanically sound. Additionally, it should preferably be constructed in a modular easily maintained manner (such as the standard Eurocard system - which also has the advantage of allowing incorporation into other manufacturers racking systems). A modular approach further extends the range of applications, as discussed later in this introduction, and it also means that spare modules can be held on-site, whilst a complete integrated unit may be prohibitively expensive to hold in stock.

Having considered all these factors outlined above, the Company examined the market for potential suppliers, the first choice being the approved manufacturer, mentioned earlier. A limited range of equipment was available, but it was prohibitively expensive, and required an external controller in the form of a desk-top type microprocessor. A networking system was available, but this was again very expensive, and failed to work when demonstrated. Thus the Company began to look elsewhere for sources of supply.

Several companies were able to offer systems that partly fulfilled the requirements outlined above - a considerable number of companies are in the process control/industrial automation market, offering expensive and inflexible "turnkey" systems. The products of a smaller number of firms who manufacture rack based interface systems (typically "Eurocard" based) offering a (limited) range of plug-in options were examined, but the systems offered were either biased towards "low noise" environments or lacked the required networking capability. A possible hybrid system was considered - that is to say that it was envisaged that the total equipment could originate from two independent suppliers, the interface equipment

being supplied from one source and the networking equipment from another. However, apart from defaulting on the single-source requirements, it did not prove to be a viable alternative economically. Those networking systems that possessed good noise immunity were, inevitably, amongst the most expensive available.

Faced with the possibility of having to purchase multi-sourced expensive equipment lacking some of the features required, the Company then considered the possibility of developing it's own; although the Company lacked the facilities to manufacturer in quantity it's own equipment, it was felt that if a prototype design could be developed and proven in a pilot scheme, then the equipment could then be manufactured by an external contractor. The author was approached to develop the prototype equipment for the pilot scheme, which is described in the next chapter. The thesis proceeds, in further chapters, to narrate how the full Distributed Interfacing System was developed from this successful initial scheme.

CHAPTER 2 - THE PILOT SCHEME

The initial scheme chosen to test the viability of a subset of the proposed hardware was a three 'node' system used to monitor an ageing production line in one of the Company's North of England sites, the aim of the scheme being to pin-point troublesome areas within the line by recording fault indications. A distributed system was preferred to a centralized system, in order to minimize cabling to sensors, thus saving installation effort, and reducing the risk of noise pick-up. A schematic diagram is shown in Figure 2.1. The three nodes are represented N1 - N3, a fourth node (Nc) is also shown - this being the 'controller', as explained later; a microprocessor node (Nm) connects an HP-85 (Hewlett-Packard desk-top computer) to the system. Standard M-70 75 Ohm coaxial cable was used as the interconnecting medium, for economy.

2.1. The 8751 Single Chip Microprocessor

To minimize size and to keep printed circuit board (PCB) layouts simple, a decision was taken to base the circuit design of the node hardware around a 'single chip' microprocessor. Such a device contains the central processing unit (CPU), memory - typically erasable programmable read only memory (EPROM) and some random access memory (RAM) - and input/output hardware, in a single package. Several such devices were at that time available, but the chip selected, the Intel 8751, was unique in that it featured an on-board full duplex UART (Universal Asynchronous Receiver Transmitter). Table 2.1. below lists the major features of the device. Reference 1 contains a full description of the MCS-51 family, of which the 8751 is a member.

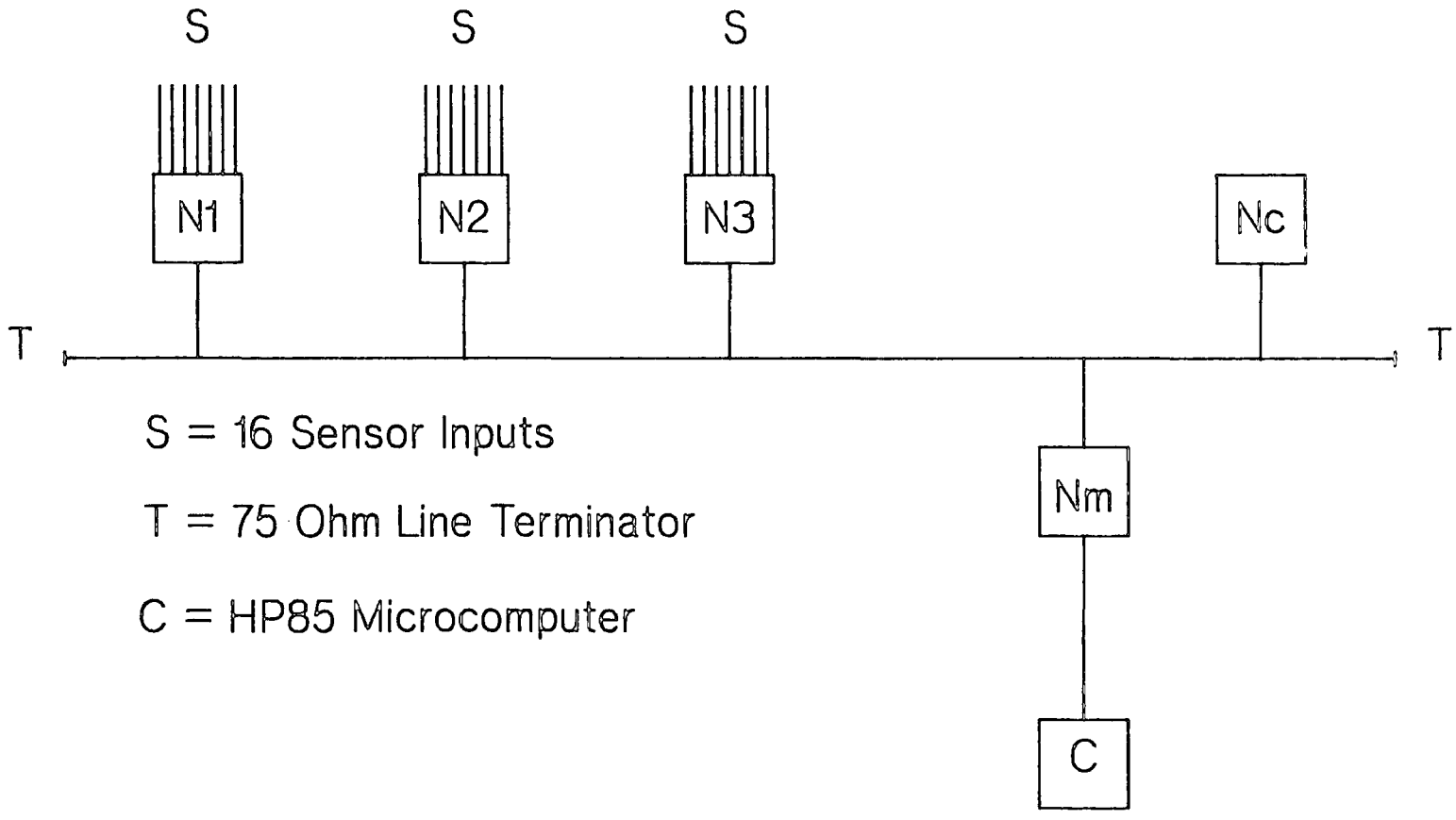


Figure 2.1 Pilot Scheme - Schematic

* Eight-Bit CPU with hardware mult./div
* Four Eight-Bit I/O ports
* 4K Bytes of EPROM
* 128 Bytes of RAM
* Full Duplex UART
* Boolean processor
* Two Sixteen-bit Timers
* Multiplexed Sixteen-bit Address/Data Bus

TABLE 2.1. - The Major Features of the Intel 8751 Microprocessor

For the pilot scheme, the full capability of the chip was not exploited. The eight-bit internal architecture was considered adequate, as was the number of input/output ports; indeed only two of the ports were used for data transfer, as we shall see later in the chapter. (Chapter 3 gives details of the 8751 facilities not relevant to this chapter, and also provides a pin-out diagram - Figure 3.1.).

The UART was of immeasurable value to the author in the early 'breadboard' stages of the design work, as it allowed the connection of serial devices (e.g. computers, printers, visual display units [VDU's]) as test units. However, the fact that the UART is an integral part of the 8751 does have its disadvantages - mainly because the UART is timed from the microprocessor's on-board oscillator (master clock), which is referenced to an external crystal. Appendix 1 discusses in detail aspects of the oscillator circuitry and lists the baud rates available from the 3.6864 MHz and 7.3728 MHz crystals. Early versions of the 8751 were available with a maximum clock frequency of 8 MHz, effectively limiting the clock frequency to a maximum value of 7.3728 MHz.

The author had some in-depth discussions with the Intel support staff as regards the preference of oscillator components (discussed

more fully in Appendix 1). The accuracy of the serial data timing is wholly dependent on the stability of the clock circuitry; Intel's recommendation of the use of the on-chip Pierce oscillator with the minimum of external circuits led to the requirement of purchasing crystals quoted for parallel resonance. These were obtainable 'off the shelf' from component suppliers for the 3.6864 Mhz frequency, and therefore this was selected as the operating frequency for the hardware described in this chapter. The need for special clock frequencies (and non-standard crystals) is a disadvantage the 8751 incurs by supporting the on-board serial facilities. Another disadvantage is the implied inability to 'hesitate' the CPU clock - a matter discussed in Chapter 3.

Other features of the 8751 listed in Table 1, although of great value in pursuing development work (described in the later chapters of this thesis), were not exploited in the preliminary work.

2.2. Design Criteria (node units)

Having thus selected a suitable intelligent device around which to base the design, the author then turned to the problem of isolating the total design requirements, i.e. determining the design 'goals'. These are summarized in Table 2.2. below:

* Sixteen opto-isolated inputs per node
* Single 5v supply
* High noise immunity
* Serial half-duplex data transmission
* Minimal cost
* Minimal software/firmware
* Simple PCB artworks
* Watchdog on-board
* Rapid development

TABLE 2.2. - Design goals for initial development work

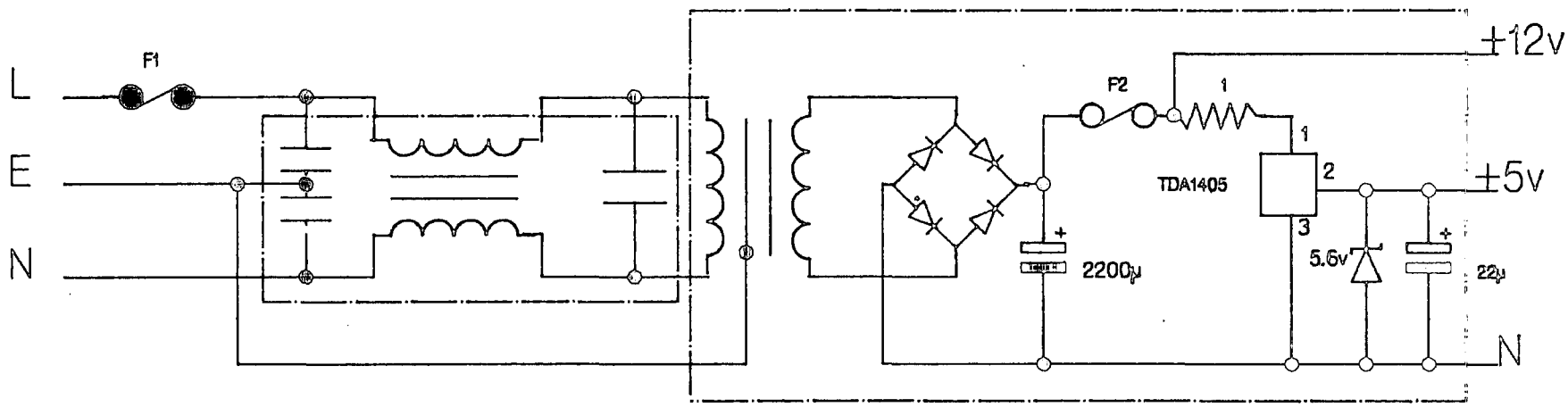


Figure 2.2 - Power Supplies.

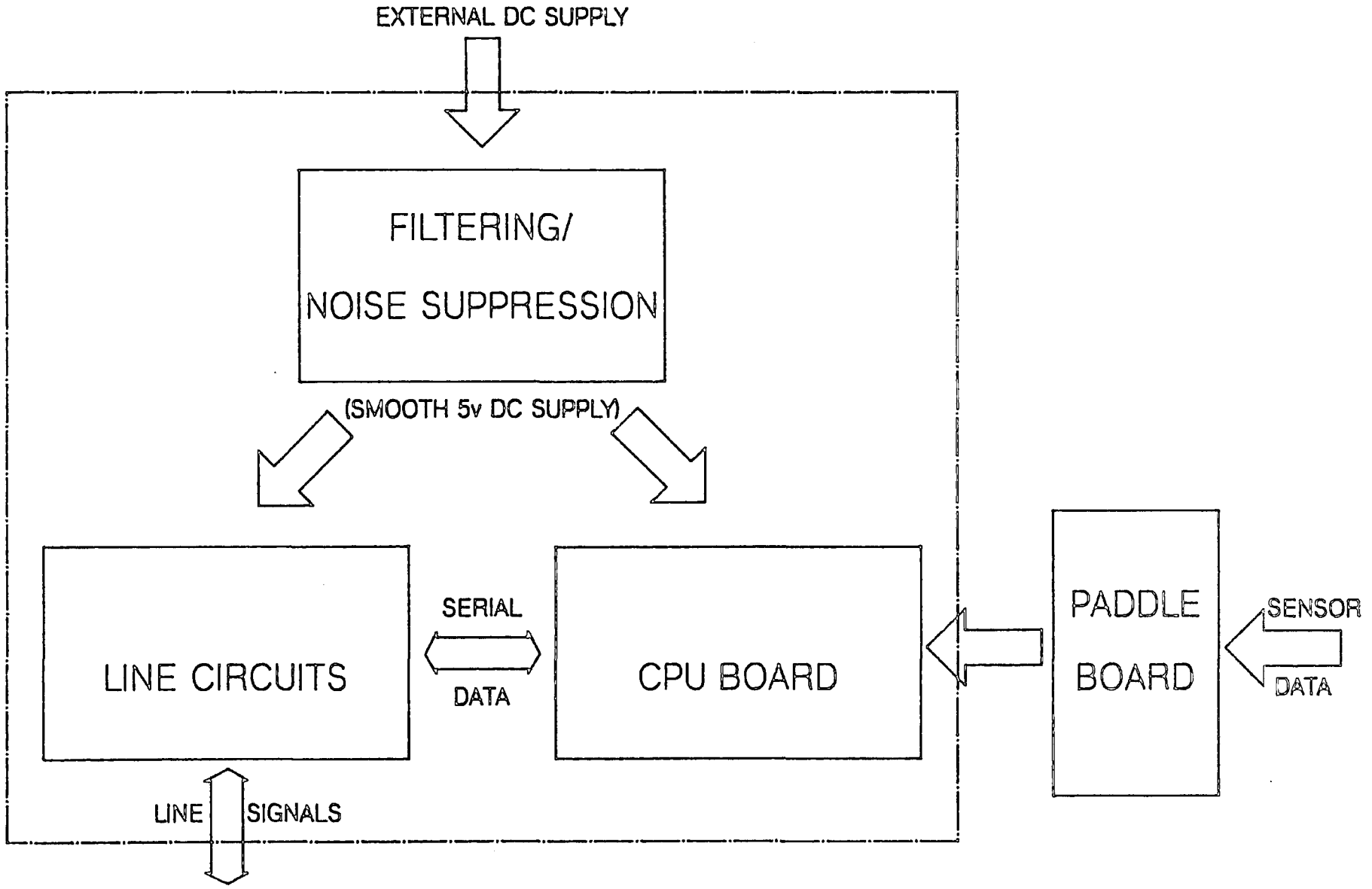


Figure 2.3 – Schematic Diagram of a Node.

The rest of this chapter narrates the author's attempts to meet these design goals for the trial system, and summarizes the achievements and the lessons learnt. The Thesis continues, in the subsequent chapters, to outline the development of the complete Distributed Interfacing System.

2.3. Power supplies

Great attention was paid from the early stages onwards, to the elimination of noise 'spikes' from the electronics. As stated earlier, the transients may be either induced or mains-born - in the latter case most of the transients may be eliminated by a conventional commercial mains filter, although such a filter may prove to be incapable of eliminating completely the largest spikes.

The technique adopted in this prototype system was to filter the mains supply as described, using a Belling-Lee (model L2140/2L) mains filter, before converting to 5v DC using an inexpensive (Farnell D5-05A) sub-miniature power supply (see Figure 2.2.) - and then to attenuate the remaining transients to an acceptable level. Figure 2.3. shows the schematic diagram for each (n1, n2, n3) node; the power (5v supply) arrives on the board at a central location, where it is further smoothed by series RF chokes, a large (100 micro-Farad) reservoir capacitor, and finally a solid-state transient suppressor. It is then fed from this central point to the various circuits in the node. Most of the circuitry of the prototype nodes was located on the main printed circuit board (PCB), with the 8751 CPU and its associated oscillator and reset circuits located on a small 'Piggy-back' PCB.

2.4. Line circuits

The need to transmit a serial data stream over long distances (several hundred metres) in an electrically hostile environment requires more than a simple amplitude modulation system such as, say, RS232 or TTL with line drivers. The commercially available option that springs immediately to mind is the modem; the essence of this device is to generate one of two basic frequencies according to the logic state of the incoming (TTL) signal - i.e., two-state frequency modulation (FM). At the receiving end of the transmission line the signal is demodulated (hence the name, MODulator-DEModulator), typically by a phase-locked-loop (PLL), to recover the original TTL signal.

To design a miniature modem is not a difficult task, as several suitable PLL's are available, the most commonly used being the Signetics NE565. This chip has two disadvantages, firstly it requires a dual voltage power supply, and, secondly, it is inefficiently packaged. A suitable alternative is the NE567, an 8-pin device requiring only a single 5-volt supply. This chip is frequently used as a tone decoder, but has sufficient bandwidth to decode serial data transmitted at rates of the order of 10 kilobaud.

The 567 has an open collector output; the output transistor is driven into saturation whenever a tone of the correct frequency is present at the input. Figure 2.4. shows the 567 in its usual use as a PLL; the resistor-capacitor combination

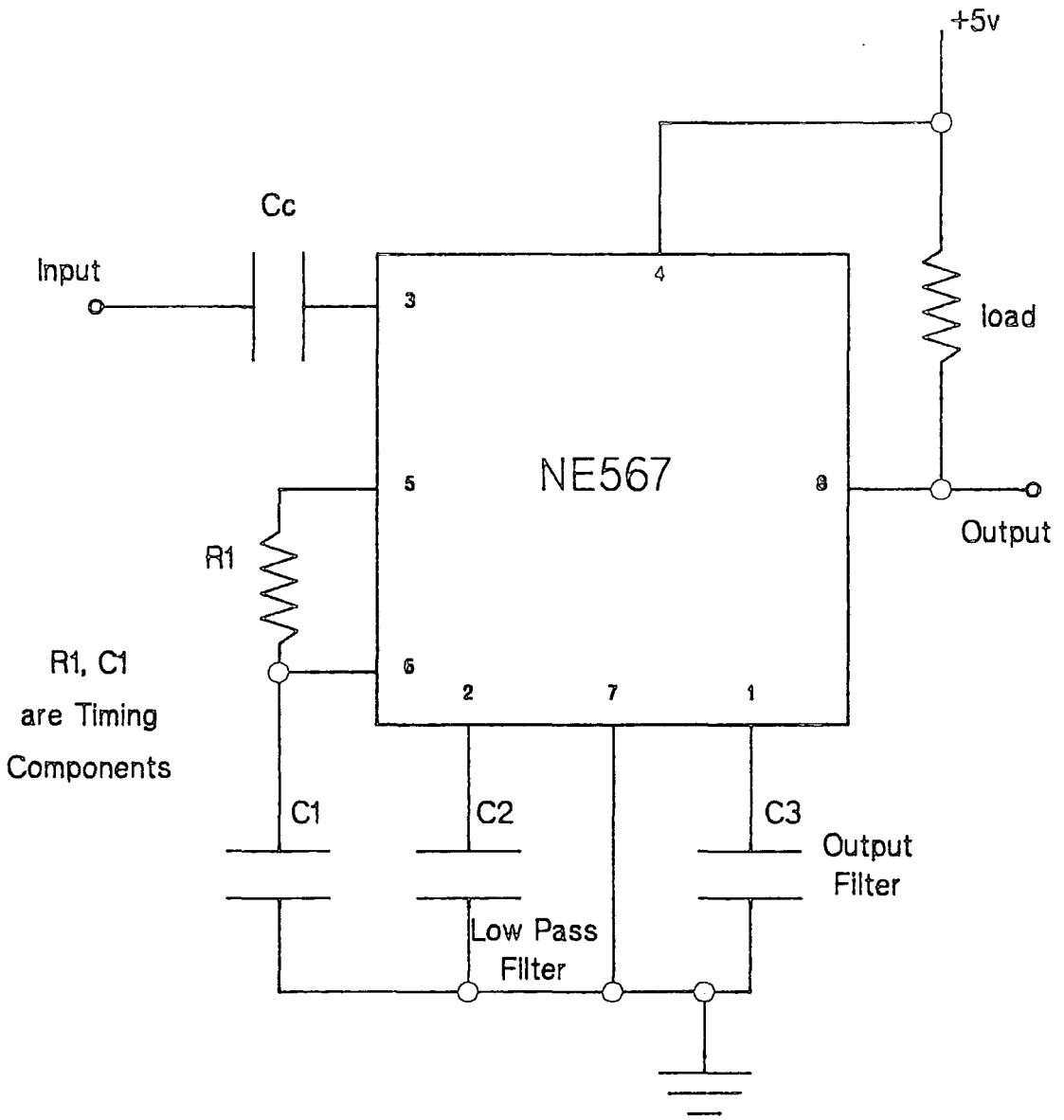


Figure 2.4 - The NE567 configured

As a Phase Locked Loop.

determines the centre frequency, while external capacitors adjust the loop capture range and the output filter response. With a suitable pull-up, the output is TTL compatible. The output is then at logic '1' state when the loop is idle, and at logic '0' when locked onto a tone of the correct frequency. Thus if we consider a transmission system that 'tone-bursts' on logic '0's only, then clearly a '567 can decode the incoming signal into a form immediately presentable to the serial input of an 8751 microprocessor. This is the basis of the transmission system implemented.

The 567 is a versatile chip enabling, for example, the independent use of the on-board voltage controlled oscillator (VCO) for other purposes. This versatility permitted the use of the VCO as an inexpensive tone burst generator, thus allowing the same type of chip to be used as both the encoder and decoder of the line signals. Figure 2.5. shows the basic VCO circuit, which together with the PLL circuit originates from the Signetics application notes (Ref. 2). These circuits configured for a centre frequency of approximately 200 KHz (See Appendix 2) are incorporated unmodified (excepting the addition of supply trimming potentiometers) into the circuit shown in Figure 2.6. The buffered serial data stream is used to pulse the oscillator and generate the logic '0' tone bursts.

The output of the VCO is a pulsed 200KHz square wave, which is TTL compatible, and can be buffered by a conventional TTL line driver, in this case the Texas 75123. The 75123 is used to drive a 75ohm coaxial line, terminated at the characteristic impedance. The serial data stream is also used to gate the

line driver, floating it's open-emitter output on logic '1's - the normal state for an idle transmitter - and allowing other nodes to transmit.

The line receiver circuit is the 567 configured as a PLL, as already described. The minimum values for the two external capacitors (input/output filters) were calculated from the formulae supplied with the Signetics application notes, and were found to give the best performance for noise rejection. Note that the PLL is used to detect tone bursts that occur during the periods corresponding to logic zeros in the original TTL signal; when the '567 detects a tone burst its output is driven LOW. The external pull-up guarantees the logic one state during the absence of the tone-bursts, which of course includes the line idle condition.

2.5. CPU sub-board

The 8751 CPU and its associated reset and oscillator circuitry, together with connectors etc., were mounted on a miniature pre-prepared PCB sub-board, which the author designed for earlier 'breadboard' tests on the 8751. The use of this item as a circuit component eased the development task. The sub-board was mounted on the main PCB by means of short 'stand-offs', both boards being enclosed within a small die-cast box. The sub-board's 20-way connector allowed a ribbon cable connection to an external 'D' socket mounted on the lid of the box. This enabled data input from the opto-isolating 'paddle board' (see below) to be routed to the microprocessor input ports. Only two of the three general purpose ports were used on these prototype nodes as input

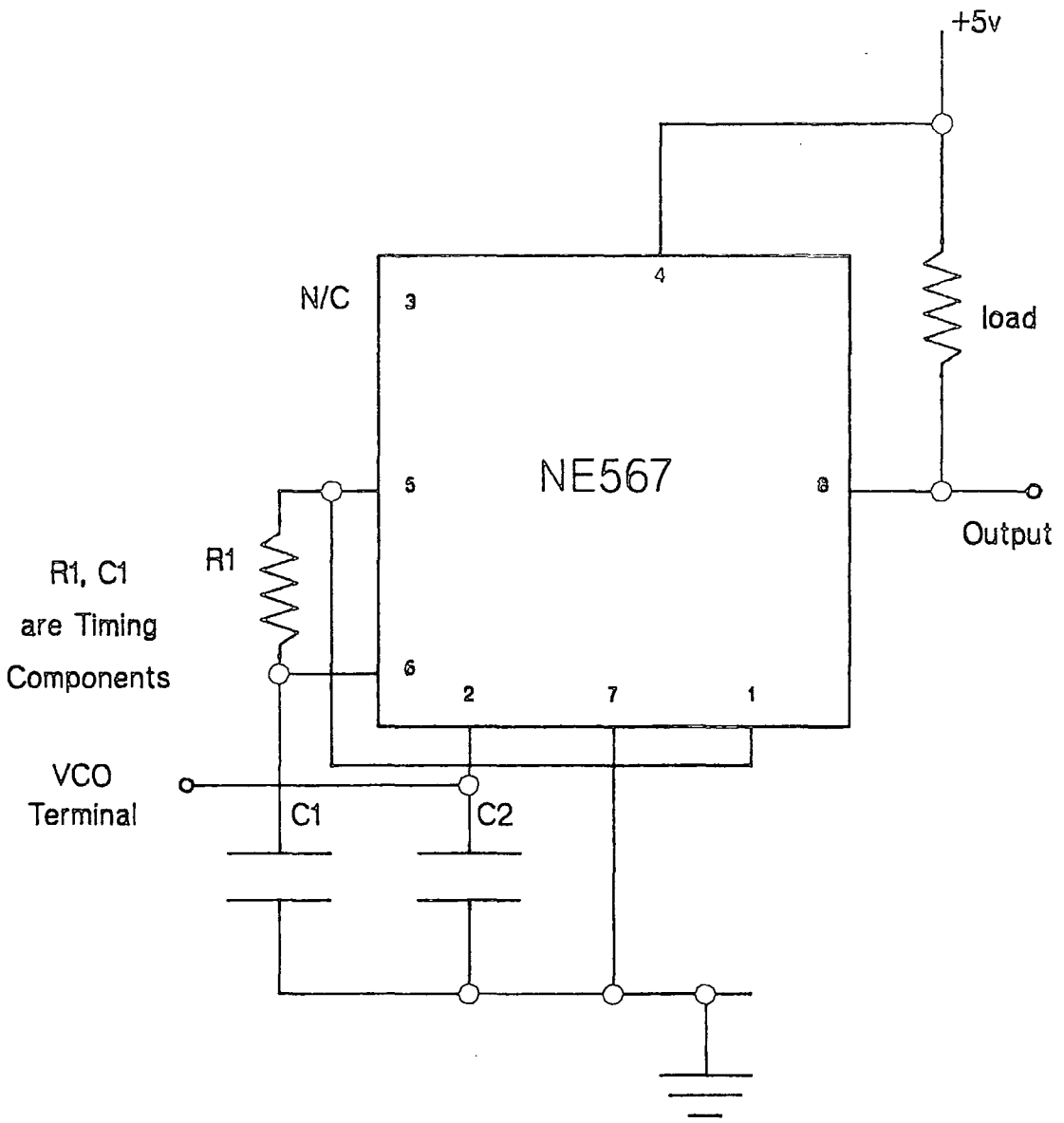


Figure 2.5 - The NE567 configured

As a Voltage Controlled Oscillator.

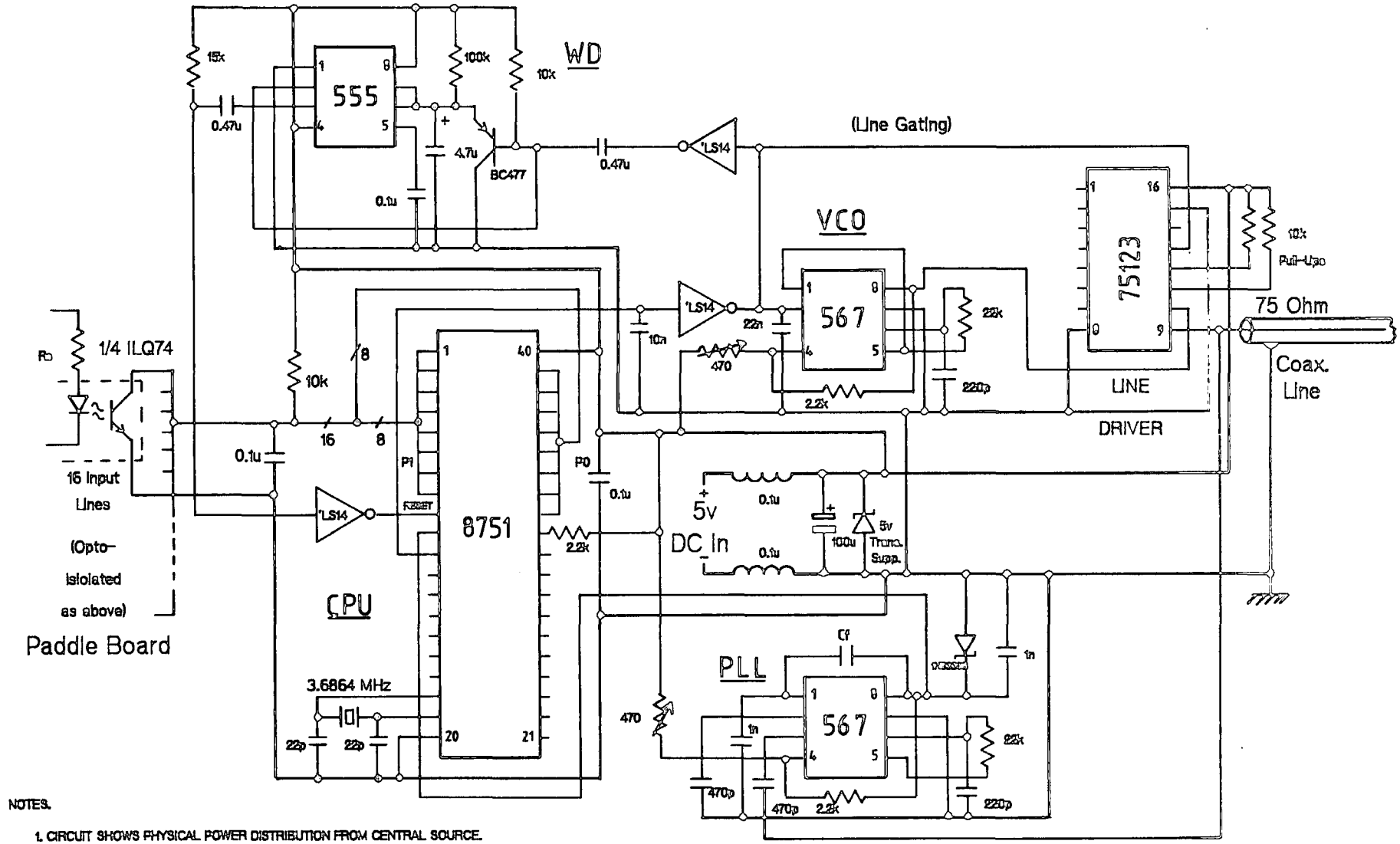


Figure 2.6 – Circuit Diagram of a Node.

ports, the sixteen input lines offered by each node being more than adequate for the pilot scheme.

2.6. Watchdog

Also featured in Figure 2.6. is a 'watchdog' circuit, a retriggerable monostable circuit as found in standard References (Ref. 24). The watchdog is retriggered by pulses from the 8751's serial port; it has a decay time of 500 milliseconds before expiring and pulsing the reset line. Thus, provided the node responds to 'polling' (see below), the watchdog remains in a quiescent state. A watchdog circuit is essential in industrial applications, to recover from situations where the internal registers (e.g. the program counter) within the CPU have been corrupted. Such a situation can occur, for example, as the result of a mains 'brown-out'.

2.7. Opto-input board

This small externally mounted 'paddle board' was used to isolate the sensor input signals (0/20mA) from the CPU input ports, for noise rejection and CPU protection purposes. Four Litronix quad opto-isolator packages were used for compactness. The paddle board PCB featured a row of rigid wire posts, which allowed mechanical connection to a row of DIN rail mounted Klippon connectors. These screw-type block terminals were used to terminate input sensor signal lines.

2.8. Node firmware

Descriptions of the firmware for the various nodes are included in Appendix 2. A brief outline only is given here.

2.8.1. Sensor node firmware

A sensor node consists of the complete circuit of Figure 2.3. above. The firmware (that is, the program held in EPROM) is identical for each node (n1, n2, n3) with the exception of one byte of EPROM containing the node number - 1,2 or 3. Each node compares this node number with the controlling node's current polling character, and if a match is found it then transmits four ASCII bytes, the least significant four bits of each byte corresponding to the nibble present at the input to each quad isolator on the paddle board. Because the characters transmitted are ASCII printing characters, and the polling characters are pure binary non-printing characters, the two types of transmission are easily distinguished.

2.8.2. Controlling node firmware

The controlling or polling node is simply a device for continuously transmitting the node polling sequence; although it does not require the receiver circuits of the sensor nodes, it was constructed using the same PCB, with the PLL chip omitted. The firmware consists of a loop of code which outputs the binary sequence 8,7,..2,1 (a maximum of eight sensor nodes were allowed for, three being used initially), with a delay between each transmitted character of sufficient duration to allow the polled node to respond with the four ASCII characters described above. Thus the system is seen to be simple T.D.M. (Time Division Multiplexing).

2.8.3. Monitoring node software

The monitoring node uses the PLL receiver part of node hardware shown in Figure 2.6. The CPU (and paddle board) are not required, the serial data stream being fed directly to an HP85 serial interface, configured for TTL. A transmission rate of 9600 baud was utilized. The simple software written for the HP85 by the author was sufficient to prove the system. It consisted of a small program written in BASIC, which displayed the state of the sensors connected to system as three rows of sixteen noughts or ones on the screen of the HP85.

2.9. Noise tests

During the development the system was tested for noise immunity by deliberately injecting noise signals into the power lines and saturating the boards with EM radiation. The most sensitive parts of the circuitry were found to be the TTL serial data lines. Excellent results were obtained simply by decoupling the lines with optimum value capacitors. Not surprisingly, the die cast box eliminated the pickup of EM radiation. The box, paddle board and power supply were all mounted in a standard (earthed) metal industrial enclosure when installed on site.

2.10. Review

Several useful lessons were learnt from the pilot scheme. The RF (radio frequency) transmission system worked well. The site engineers were able to run the coaxial cable (the HP85 was sited remote from the line being monitored) through the same ducting used for the fluorescent lighting circuits without

problems occurring with data transmission. It was found necessary to trim the VCO frequency, using the supply trim potentiometer, to the mid-range of the PLL pass-band, to allow for drift due to temperature variations. Clearly a stable crystal oscillator to replace the 567 VCO would be an improvement. The die cast box was not considered a convenient enclosure, as it made access to the boards difficult. The unit was not sufficiently versatile - it could not process commonly encountered analog signals, nor interface standard communications devices such as hand-held terminals.

Despite the criticisms above, the system has performed a useful function for two years with only one malfunction - a dry joint on a PCB. The author's simple demonstration software has been replaced by an elegant software package written by software specialists within the Company's Computer Services Department, which provides statistical printouts and logging facilities on disc files.

After completing the pilot scheme, the author was asked to continue the development of the system into a versatile interfacing system. How this was progressed is discussed in the following chapters.

CHAPTER 3 - THE EXPANDED SYSTEM

The dilemma the author encountered in attempting to expand the system is known in Literature (Ref. 4) as Parkinson's Third Law, and is stated formally thus: 'Expansion means complexity, and complexity decay'. The inevitable increase in size and complexity is the price to be paid for increased flexibility. It was decided to aim at keeping the total unit size as small as possible whilst still retaining flexibility as a major requirement - though these two conflicting requirements, compactness and versatility, created problems from the outset. How, for instance, could the system have the versatility and power of a typical modular Eurocard system yet be compact enough to be accommodated in a standard wall mounting industrial enclosure? - the most common type in use on the Company's sites has an internal depth of only fourteen centimetres.

A decision was taken to split the existing design into four separate small boards; the average chip-count per board would then for the most part dictate the final board size. The principal board in this initial set was of course the CPU board, the other boards being the Watchdog/power input board, the RF board, and DVI board (digital voltage input). All these boards are described in detail below - with the exception of the RF board, which is covered in Chapter 6. We will examine first the choice of the mechanical components for the system - the racking system, the backplane, the connectors and other related components.

3.1. The Eurocard Sub-rack

A local contractor with past associations with the Company was selected to produce the emerging prototype system in small quantities for evaluation. The major function of the Contractor (see Acknowledgements), was to perform assembly work - both electrical and mechanical. As the Contractor already

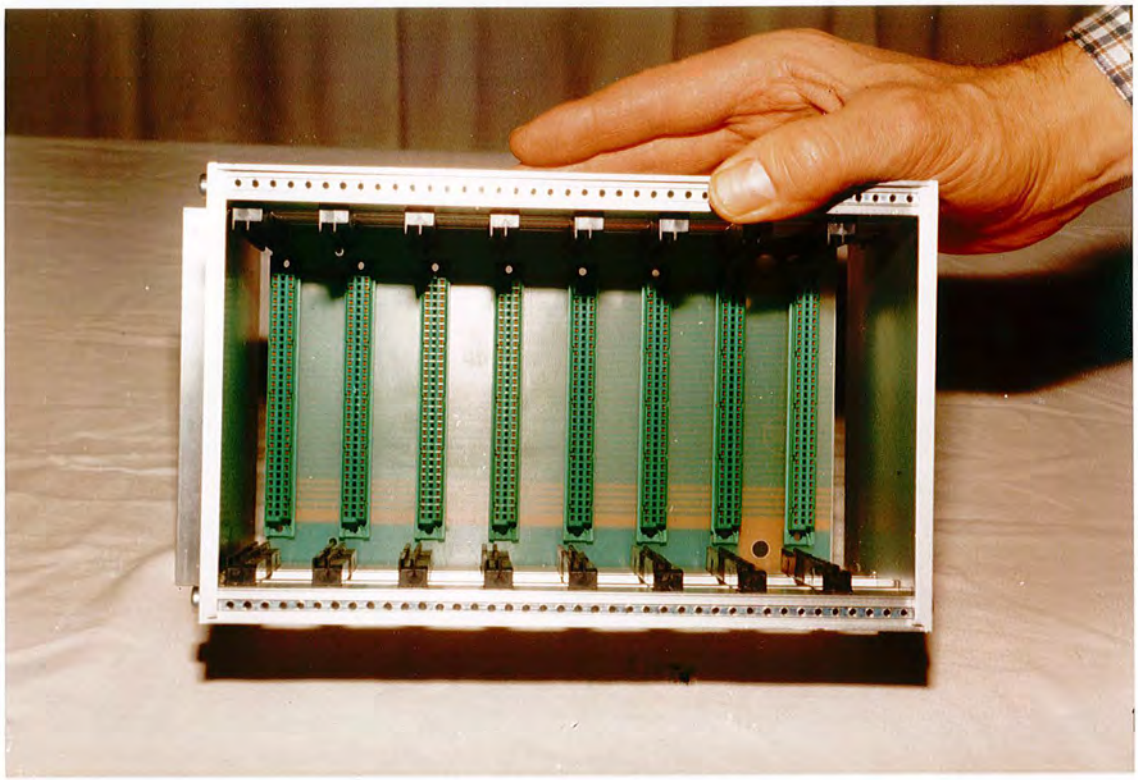


Plate 3.1. - The Sub-miniature Rack

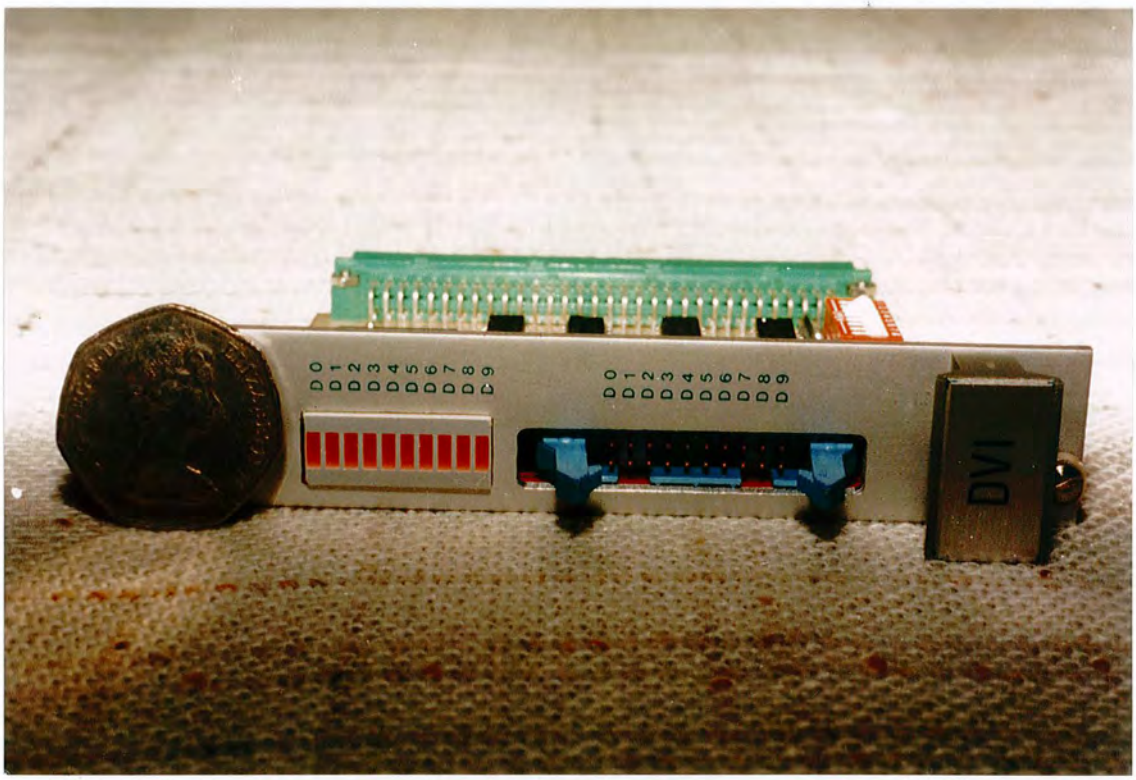


Plate 3.2. - The DVI Board

held stock items of components for other customer's systems, it was pertinent to consult with the Contractor's staff throughout the design exercise as regards circuit components, to aim for a design that would take full advantage of bulk purchase. The Schroff Eurocard racking system (Ref. 29) is a particularly well-designed versatile and compact system, with a professional appearance and low cost. This was one of the rack systems stocked by the Contractor.

Another racking system held in stock and also considered was the ISEP system, and in fact, trial assemblies of sub-racks were made using both the ISEP and Schroff parts (for comparison), but the Schroff system was found to adapt easiest to the shallow depth required. Moreover, the ISEP system was heavier and less elegant in appearance. The Schroff system was therefore selected for the implementation of the sub-rack hardware. Plate 3.1. shows a photograph of an assembled sub-rack. Although most Eurocard racking systems use a standard nineteen inch card frame, the Schroff system allows the user to assemble an eight-inch frame, which is ample for most purposes. Again, although the normal depth for a Eurocard is 160 centimetres, the side plate pressings allow for the long Eurocard, or in our case the half-depth Eurocard (80 mm.). As the card fronts fitted are one inch wide, the sub-rack capacity is eight cards (using an eight-inch rack), as can be seen from the photograph.

The backplane used in the prototypes was again a stock part from the sub-contractors, a simple printed circuit board designed to accept standard Eurocard DIN connectors with rows

'A' and 'C' fully pinned (32 pins each rows) and row 'B' omitted.

Figure 3.1. shows the signal names assigned to the pins.

```
                  C    A
                  o--1-o GROUND
            port 0.0 o  2 o port 1.0
          B port 0.1 o  3 o port 1.1
            port 0.2 o  4 o port 1.2
          U port 0.3 o  5 o port 1.3
            port 0.4 o  6 o port 1.4
          S port 0.5 o  7 o port 1.5
            port 0.6 o  8 o port 1.6 Peri*/memory
            port 0.7 o  9 o port 1.7
           Address 0 o 10 o RESET
           Address 1 o 11 o INTERRUPT 0*
           Address 2 o 12 o Timer 1
           Address 3 o 13 o Timer 0
           Address 4 o 14 o INTERRUPT 1*
           Address 5 o 15 o Clock 1
           Address 6 o 16 o Address Latch Enable (ALE)
           Address 7 o 17 o Prog. Store Enable (PSEN)*
ADD. 15 Port 3.7  o 18 o Relinquish Control (REL)
     14 Port 3.6  o 19 o RESET*
     13 Port 3.5  o 20 o Clock 2
     12 Port 3.4  o 21 o Spare
     11 Port 3.3  o 22 o Spare
     10 Port 3.2  o 23 o Spare
      9 Port 3.1  o 24 o Spare
      8 Port 3.0  o 25 o Spare
       Serial TX  o 26 o Spare
       Serial RX  o 27 o Spare
                  o-28-o WRITE*
                  o-29-o READ*
                  o-30-o +12v
                  o-31-o -12v
                  o-32-o +5v

                                          *Active LOW
```

Figure 3.1. Backplane Connector Pin-out

The backplane tracking consists of a series of parallel lines interconnecting like-named pins on the eight DIN connector sockets. Each signal track is 0.5 millimetre thick, with heavier tracking provided for power lines (and the read/write lines - assigned, for utility, to spare power lines). The backplane, as explained, was a part 'borrowed' from another system; it did not feature intertrack ground

lines, a desirable feature if the system should enter mass production at some future time, especially if the CPU clock rate should be increased. However, during the development work there was no evidence of backplane crosstalk, possibly attributed to both the low CPU clock rate and the wide track spacing on the simple PCB.

3.2. CPU board

The signals present on the backplane connector pins are for the most part determined by the structure and design of the CPU Board; this in turn reflects the facilities afforded by the 8751 - which are fully exploited in this system. The development system (Ref. 18) used for developing test firmware was the Intel PDS (Personal Development System), which the author used to assemble, edit and run programs. Although not as convenient or as powerful as a standard Intel (MDS) microcomputer development system, this cheap portable system proved to be adequate. The emulation vehicle used was the Intel EMV51 (Ref. 19), which is a plug-in option for the PDS. EMV51 is essentially a device that replaces the 8751 on the CPU board with an electronic 'umbilical chord' linking it to the PDS; this allows the user to monitor the internal registers of the processor during the debug stage. Although this thesis repeatedly refers to the 8751, the author used EMV51 exclusively for firmware debugging; for final testing of debugged firmware the 8751 was used and some work was also done with the 8052-BASIC device (see Appendix 8).

The author opted for a 'memory mapped' I/O (input/output) scheme to realize the desired flexibility in interfacing to external systems (Ref. 5). This system is described below in

sub-section 3.2.1. Subsequent sub-sections cover the remaining features of the board.

3.2.1. The Bus and Memory Mapped I/O

The Bus system of the 8751 is derived from the Intel 8085, the successor to the still-popular Intel 8080, whose most famous descendant is the ubiquitous (Zilog) Z80. The latter processors use separate address and data lines, while 8085-type addressing structures multiplex the lower 8 bits of the (16-bit) address lines with the 8 data lines on one 'bus' port usually referred to simply as 'the Bus'. The Bus on the 8751 is assigned to the port 0 lines, whilst the upper eight address lines are mapped to port 2. Figure 3.2. below shows the 8751 pin-out.

```
                  ............
        port 1.0 :1        40: Vdd(5v)
        port 1.1 :2        39: port 0.0
        port 1.2 :3        38: port 0.1
        port 1.3 :4        37: port 0.2
        port 1.4 :5        36: port 0.3
        port 1.5 :6        35: port 0.4
        port 1.6 :7        34: port 0.5
        port 1.7 :8        33: port 0.6
           RESET :9        32: port 0.7
            RxD  :10       31: EA/VDD
P           TxD  :11       30: ALE
O          INTO* :12       29: PSEN*
R          INT1* :13       28: port 2.7 (Add 15)
T             T0 :14       27: port 2.6 (Add 14)
              T1 :15       26: port 2.5 (Add 13)
3            WR* :16       25: port 2.4 (Add 12)
             RD* :17       24: port 2.3 (Add 11)
          XTAL 1 :18       23: port 2.2 (Add 10)
          XTAL 2 :19       22: port 2.1 (Add  9)
 Vss (GROUND) :20          21: port 2.0 (Add  8)
                  ............
                                          *Active LOW
```

Figure 3.2. - 8751 Pin-out

Although nominally designated 'ports', ports P0 and P2 implement the address/data Bus, while P3 provides eight special one-line functions, described later. The remaining port (P1) is reserved for use as eight individual I/O lines (e.g. P1.6 is used as a peripheral memory control line - see below), although it could also be employed as a port during firmware testing, as an aid to diagnostics. During a write to external data memory (see Figure 3.3) the 8751 generates signals on several pins. For example, the signal ALE (address latch enable) is emitted on pin 30; because of the way the 8751 fetches program memory bytes internally (in pairs), ALE is clocked at half the normal rate when external data memory is accessed. This issue is dealt with in more depth in Chapter 4, which covers the RAM (Random Access Memory) board.

The low level state of ALE is used to latch the lower eight bits of the multiplexed address into an eight-bit latch. The latch is shown as U3 (74ALS573) in CPU board schematic diagram (Figure 3.4). ALE is also buffered to the backplane (by U5c - 74LS125) to act as a timing/control signal for peripheral cards. Two other signals are also relevant to external memory access - RD (read - pin 15) and WR (write - pin 16), both active LOW. These signals are two examples of alternative uses of port 3 pins; thus whenever the Bus feature of the 8751 is utilized, use of port 3 as a conventional port is excluded. However, this is a small price to pay for the powerful facilities the Bus

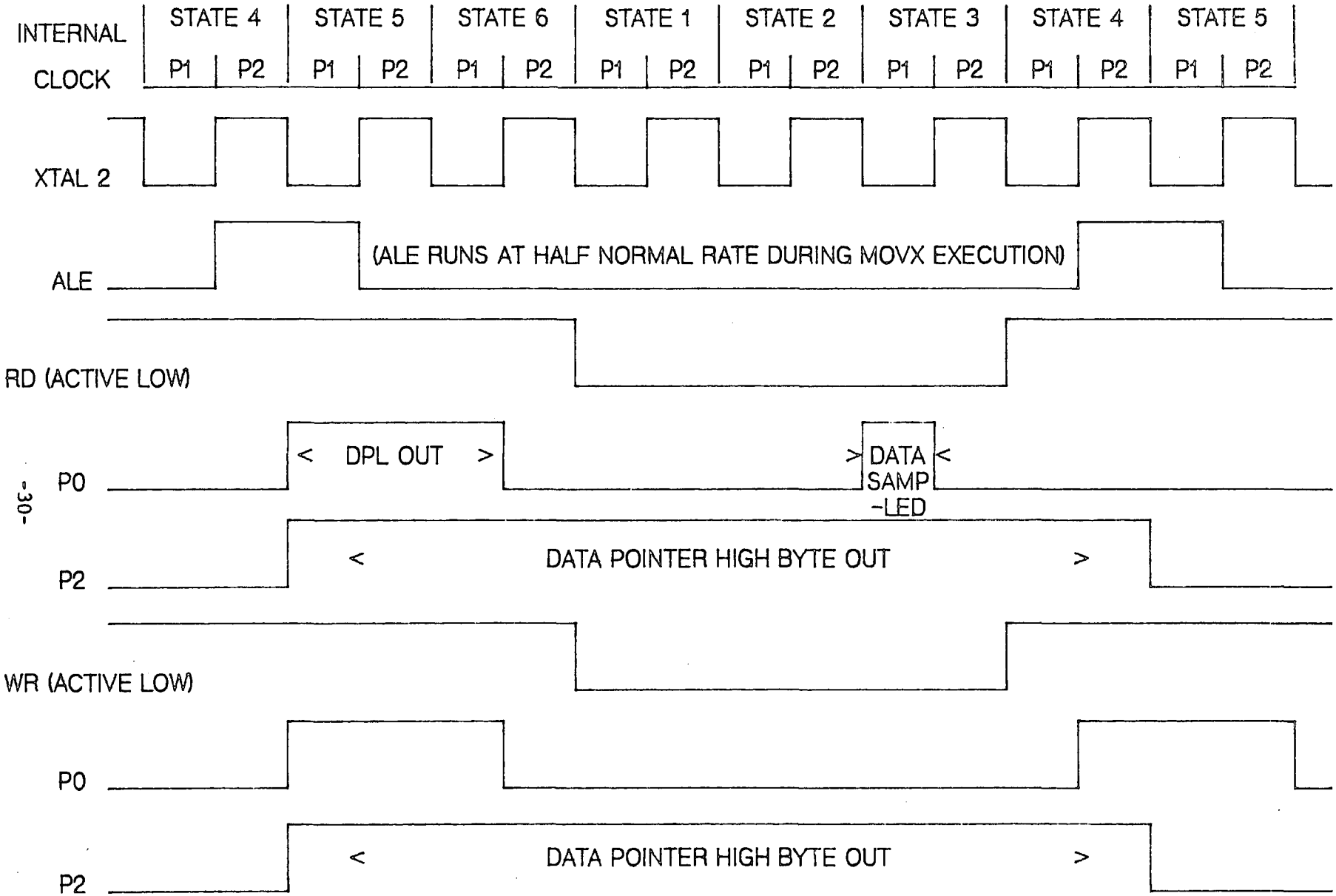


Figure 3.3 – Timing Diagram for External Memory Access.

affords. The signals RD and WR are also buffered to the backplane by U5. They are used to enable data to or from memory and peripheral cards.

To enable peripheral cards to be addressed, four address lines (A12-A15) have been 'borrowed' from the Bus - a technique known as 'memory mapping'. This allows up to sixteen individual cards in the rack to be addressed - more than sufficient for most applications. The CPU sees the peripherals as sixteen high order address locations; there is no conflict between the peripheral addresses and the first 4K of external RAM. This was considered an adequate quantity of RAM at first, but at a later stage in the project the requirement for addressing larger quantities of external memory arose - possibly up to 32K, or perhaps even 64K. To allow for these larger memory requirements, one line of port 1 (P1.6 - PERI(BAR)/MEMORY) was buffered to the backplane (a second, spare line - P1.7, was similarly buffered, for general use). The P1.6 signal line has been gated on all peripheral and memory boards with the relevant address decode logic to select either memory or peripheral accesses. A zero latched by firmware to P1.6 will select peripheral devices; a one latched to P1.6 will select memory access. Details of several other port 1 pins which provide optional facilities (e.g. serial handshake lines for RS232 options) for various peripheral cards, are given in the chapters which relate to the particular peripheral cards.

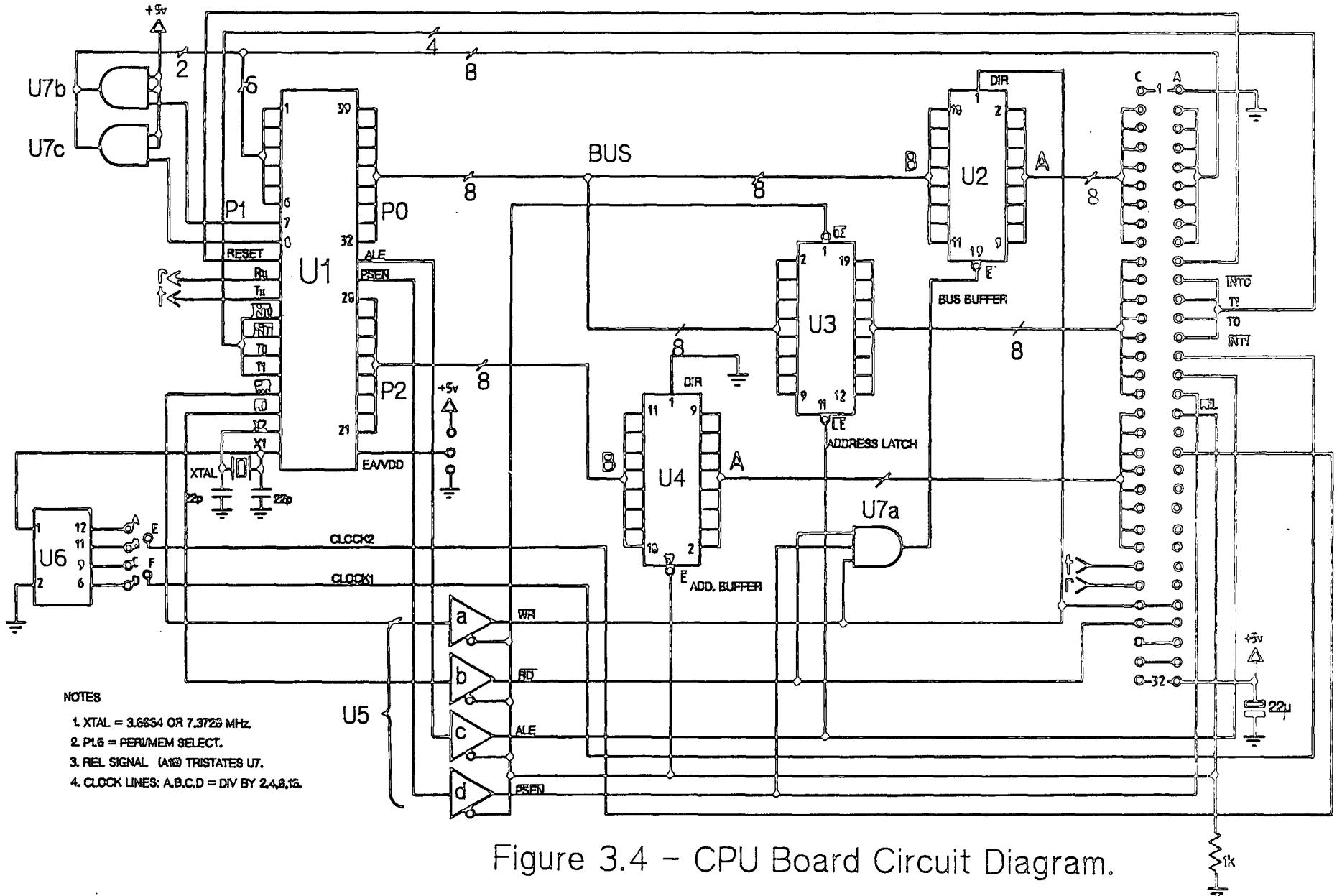


Figure 3.4 – CPU Board Circuit Diagram.

Bus buffers are provided for ports P0 and P2; these are U2 and U4 respectively. Both buffers are 74LS245's - these provide the necessary increased output fan-out and also protect the 8751 against accidental damage. The control of the output enable by the gating provided by U7A enhances this protection in the case of P0 - the direction being controlled by the WR signal. Directional control of U4 is not required, as the upper address lines are uni-directional. A control line has been allocated to backplane pin A18 (REL), to allow another intelligent device to gain control of the backplane lines. This could possibly be, say, a video controller card (a likely future requirement) with a need for asynchronous access to the RAM memory board. Typically the device would participate in an interrupt/status handshake with the CPU board, before completing the Bus release by taking REL high, thus tri-stating the active control/address circuits.

3.2.2. Clock circuitry

The 8751 features an on-board crystal oscillator which requires a minimal set of external components, as shown in Figure 3.4. The oscillator circuit is in fact a single stage amplifier connected as a Pierce oscillator (Ref. 8). XTAL 1 (pin 19) is the input to the oscillator and XTAL 2 (pin 18) is the output. To prevent the extremely small input capacitance of the FETs (field effect transistors) that form the oscillator amplifier 'pulling' the crystal above its

required operating frequency (overtone start-up is disastrous to the operation of the 8751 serial port, for example, which is referenced to the clock frequency), two balanced phase-shifting capacitors shunt each side of the crystal to ground. Fisher (Ref.7) reports suitable values of 30 pF (obtained empirically) for these capacitors giving reliable start-up in the frequency range 500 kHz to 15 MHz, with pulse rounding at the latter frequency. (At the time of writing the 8751 is specified to 12 MHz - although at the start of the project the part was limited to 8 MHz).

Appendix 1 lists some sample serial port baud rates for the 3.6864 and 7.3728 MHz crystals; the baud rate is determined by the clock frequency and by the reload constant loaded into the timer register 'TH1'. Crystal selection is not a straight-forward matter in this case; for most microprocessors, an approximate operating frequency will suffice, but the accuracy of the 8751 serial port timing is important for reliable serial data transfer. Crystals have several selection parameters (Ref. 8), and are quoted as for either series or parallel resonance oscillators. A crystal quoted for series resonance will work reliably in a parallel resonance circuit, but will oscillate at a frequency slightly removed from that quoted by the crystal manufacturer, hence it is important in this case to obtain the correct component. Parallel resonant crystals can be purchased off the shelf from

component distributors for 3.6864 MHz, and so early development work was based on this frequency. Later tests were performed at 7.3728 MHz, using crystals made to order (Ref. 7).

A penalty for having an on-board UART is the impracticality of including 'hesitate' circuitry ('freezing' the CPU by withholding clock pulses) for the clock oscillator; the reception of serial data being essentially asynchronous, clearly it would be unsatisfactory to disable the UART for even short time intervals. In a typical 8080 or 8085 system, for example, the 8224 clock generator (Ref. 9) would provide the hesitate function via the READY input. This line would appear as a backplane signal in such a system, enabling other boards in the system to pause the CPU - typically while awaiting external data. This rules out the use of some families of interface chips - e.g. PIA's (Ref. 16) which provide a hesitate signal to the host CPU - for use on the various interface cards that comprise the miniature rack system. This matter is discussed further in Chapter 5, in the section covering the BCD interface design.

The CPU board provides two clock lines (Clock 1, Clock 2) which may be utilized by other boards in the system, via the backplane connections. U6 is a high speed CMOS clock divider (HC 4024); the high input impedance and the CMOS logic levels of the inputs are an excellent match to X1 pin's low drive output parameters, allowing a direct connection to the 8751's

Pierce oscillator. Links are provided on the board to enable the user to select the required clock rate(s). During testing the author selected the clock 1 rate to be 921.6 KHz; Clock 1 is used by several cards - e.g. ADC (Chapter 7) - whilst clock 2 is provided for possible future use. The jumper configuration used is, of course, partly dependent on the crystal frequency.

3.2.3. Reset

The reset signal, which provides a positive CPU reset pulse on power-up, is provided by the watchdog card (section 3.4.) on backplane pin A10. The watchdog card also features a manual reset push-button which allows an operator to reset the system.

3.2.4. External Memory Features

Data memory external to the CPU is covered in the next chapter; P1.6 is used as a memory select line, with P1.7 being similarly buffered (U7b, U7c) for possible future use. External program memory is also allowed for - the jumper connected to pin 31 specifying whether the program resides on internal or external EPROM (Erasable Programmable Read-Only Memory). Up to 64K of external EPROM may be addressed, although the 4K on-chip is considered adequate for the Company's applications. PSEN (Program Store Enable - active LOW) is the control signal used to gate external PROM's, and is buffered to the backplane by U5d. An inexpensive EPROM-less version (the 8031) of the 8751 is available, should

the Company require the development of EPROM programs in excess of 4K in size, in which case these programs could reside in the external 64K memory areas, on a separate EPROM board.

3.2.5. Miscellaneous - Timers etc.

Two timer pins (T0 and T1) are connected directly to the backplane (pins A13 and A12 respectively), as are two interrupt lines (INT0 and INT1 - backplane pins A11 and A14 respectively); the timer pins may be used (software configurable) as either event counters or one-bit I/O lines (Ref. 1). The interrupt lines are similarly software configured as either active low (input) interrupt pins or general-purpose I/O lines - invariably they are used for interrupts. Normally a device (interface board) requiring attention would force, say, the INT0 line LOW, and the CPU would respond by reading the status register of each card until the interrupting device is identified - an interrupt triggered serial poll sequence referred to herein as an 'interrupt/status handshake'. Usually the sequence completes with an exchange of data between the two boards - an entirely conventional method of servicing 8085 type peripheral chips such as, say, the 8251 USART (Refs. 21,22). An acceptable alternative approach to addressing peripherals considered during development was the use of DMA (Direct Memory Access) using a DMA controller chip such as the Intel 8237, which is designed for DMA implementation on 8085-type bus systems. However,

this would have meant a separate DMA board due to the size of the DMA chip (40-pin) - an unacceptable hardware overhead.

The 8751 serial port pins RXD (Receive Data - pin 10) and TXD (Transmit Data - pin 11) are also connected to the backplane (pins C27 and C26 respectively), to provide serial data to the RF board or the Serial board - both of which are described in Chapter 6. (See this chapter also for details of use of port 1 pins - P1.0 and P1.1 - as serial data handshake lines). The serial port options (baud rate, parity checking/generation, start/stop bits etc.) are all firmware selectable, details of which, together with a sample firmware listing, are contained in Appendix 3.

A small 5v reservoir (22 micro-Farad) has been included on the card, in common with all the other cards (with the exception of the watchdog card, which has larger capacitors associated with the various power lines - see 3.4. below) in the system. The capacitor is sited near the 5v backplane power pins (C/A 32) to ensure a smooth DC line for the whole card. As the card utilizes one single 5v supply, there are no connections to the +12v line (pins C/A 30), or the -12v line (pins C/A 31). Ground connection to the backplane occurs on pins C/A 1. Gridded ground planes and similar noise rejecting PCB layout techniques were used in pre-production artworks for the CPU board (and/where space

considerations permitted on other boards) in an attempt to minimize the pickup of noise (Ref. 17).

3.3. The DVI Board

The DVI (Digital Voltage Input) board was designed to interface digital signals (voltage or current) to the CPU board. Figure 3.5. shows the circuit diagram. The requirement was for an opto-isolated eight-input board, memory mapped to the Bus. The board can be thought of as a memory mapped development of the simple opto-isolator board ('paddle' board) described in Chapter 2. The use of the memory-map hardware and associated device address switches means that a greater number of inputs can be monitored by one node - the rack can be populated by several DVI cards, each with a different device address.

/Continued Over

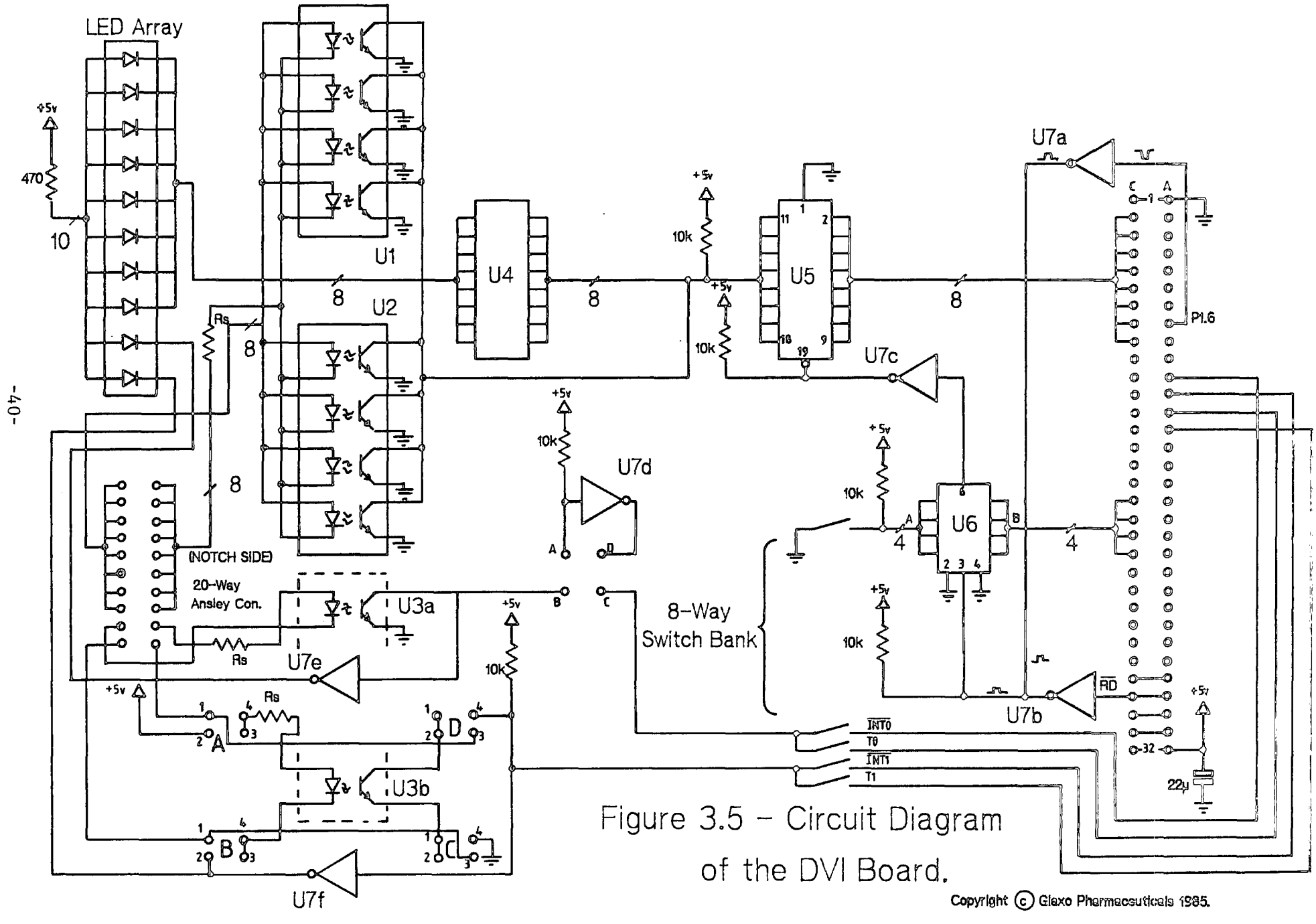


Figure 3.5 - Circuit Diagram of the DVI Board.

3.3.1. Memory mapping hardware

This hardware is described in detail at this stage, as it is a common feature of most cards in the system. The memory mapping device is the LS85 comparator (U6); it was one of a number of circuit elements chosen after consultation with the Contractor's staff, with reference to their bulk component stock. Other comparator's considered were the Fairchild F521 (8-bit) and the 'LS682 (the latter component was used on the watchdog card - see 3.4. below - in preference to cascaded '85 comparators). The comparator has two sets of four-bit input lines (A and B) - when the data present on both sets are equal then a logic one is output on pin 6 (the A=B output). The 'A' lines are connected to a four-way switch bank (switching to ground), pull-ups selecting logic ones when the switches are open-circuit. The switches define the card address within the rack.

The '85 comparator features an input enable, which is driven high by the presence of a logic one on the AND-tied output of two open-collector inverters. One of the inverters (U7a) buffers the signal PERI(BAR)/MEMORY obtained from backplane pin A8 (this is the firmware dependent signal present on P1.6, selecting either memory or peripheral accesses via the Bus, described earlier - see section 3.2.). Another 'LS05 buffer (U7b) is used to invert the active LOW read (RD) signal. Hence the comparator provides an output 'one' on pin 6 when the card is correctly

addressed, provided the input enable condition is satisfied - i.e., RD being LOW and peripheral addressing being enabled. The output is inverted (U7c) to give an active LOW output in order to drive the output-enable pin of an 'LS245 octal buffer (U5) LOW under the conditions detailed above.

3.3.2. Input isolation

The '245 is used to drive data obtained from the outputs of two Litronix ILQ74 quad opto-isolators (U1 and U2) onto the Bus during the read interval. The opto-isolators are the same type of components as used for isolation of inputs in the hardware of the simple nodes of the pilot scheme described in Chapter 2. A current in the range 4-100 milliamps through a photodiode in U1 or U2 will drive a LOW output on the (collector) output of the corresponding phototransistor. (A series resistor, user-specified, is used for current limiting on each input - allowing input signals to be sourced from whatever DC supply is convenient). Pull-ups on the outputs ensure TTL levels; the signals are also fed to a high-gain (Darlington) transistor array, which is used to drive eight elements of a ten element bargraph array, used in this application as an LED front-panel signal state indicator. The ULN2803A (U4) incorporates internal base resistors that ensure TTL compatibility.

3.3.3. Connector hardware

It is useful to mention at this stage the reason for the choice of front-panel connector for the input

signals. The connector used is the T and B Ansley 20-way polarized socket (Farnell Part No. 612 2004ES), designed to accept the corresponding polarized ribbon-connector plug. The socket features two side-mounted lock/eject tabs as well as right angled PCB solder pins for ease of assembly. The socket is of the type used as a component of the Pheonix FLK-20 20-way C-rail connector - allowing a simple one-piece ribbon cable connection between C-rail signal cables and a DVI card in a typical installation. The FLK-20 is a compact connector, splitting the 20-way signals into two tiers (ten signal connections and ten return or ground connections) of screw-tightening cable clamps. Plate 3.2. shows a photograph of a prototype DVI card.

3.3.4. Handshake Options

The availability of two spare signal/earth pairs on the front panel socket (and two spare LED indicators) prompted the addition of two opto-isolated handshake lines to the card, to allow a possible alternative use - as an opto-isolated parallel bus interface. The opto-isolator (U3) used is the Litronix ILD74, a dual isolator similar in performance to the quad package (U1, U2). Both halves of the isolator can be used as input lines (the isolator outputs are open-collector - the 8751 incorporates pull-ups within the input ports) ; the U3a output may be optionally inverted (using jumpers AB and DC) or connected (jumper BC) directly to the selected backplane line - either an interrupt

line or timer input (although, of course, the line selected may simply be a general purpose input line, according to the 8751's firmware configuration of the port pin). A typical use of this option would be as an input flag line for a remote parallel interface. U3b allows, by means of jumpers, the implementation of either a second input flag line (jumper 1 and 4), or an output control line (jumper 2 and 3).

As a general comment it can be stated that the options described above were added simply because the space was available on the board; most applications which require the use of the interface will omit U3, the card serving an eight-input passive isolation device. As the three inverters (U7d, e and f) are already present on the board, little extra production cost is incurred for the additional facilities. The marginal value of these extra facilities preclude the addition of the status register hardware found on other boards, so the DVI board is excluded from the list of serial pollable devices - i.e., it cannot take part in an interrupt/status handshake with the CPU.

3.4. Watchdog card

This card (see Figure 3.6.), is a development of two sections of the system described in Chapter 2 - the power input and transient suppression section, and the watchdog circuit. A third function - node identity - is added. Each section is described in detail below.

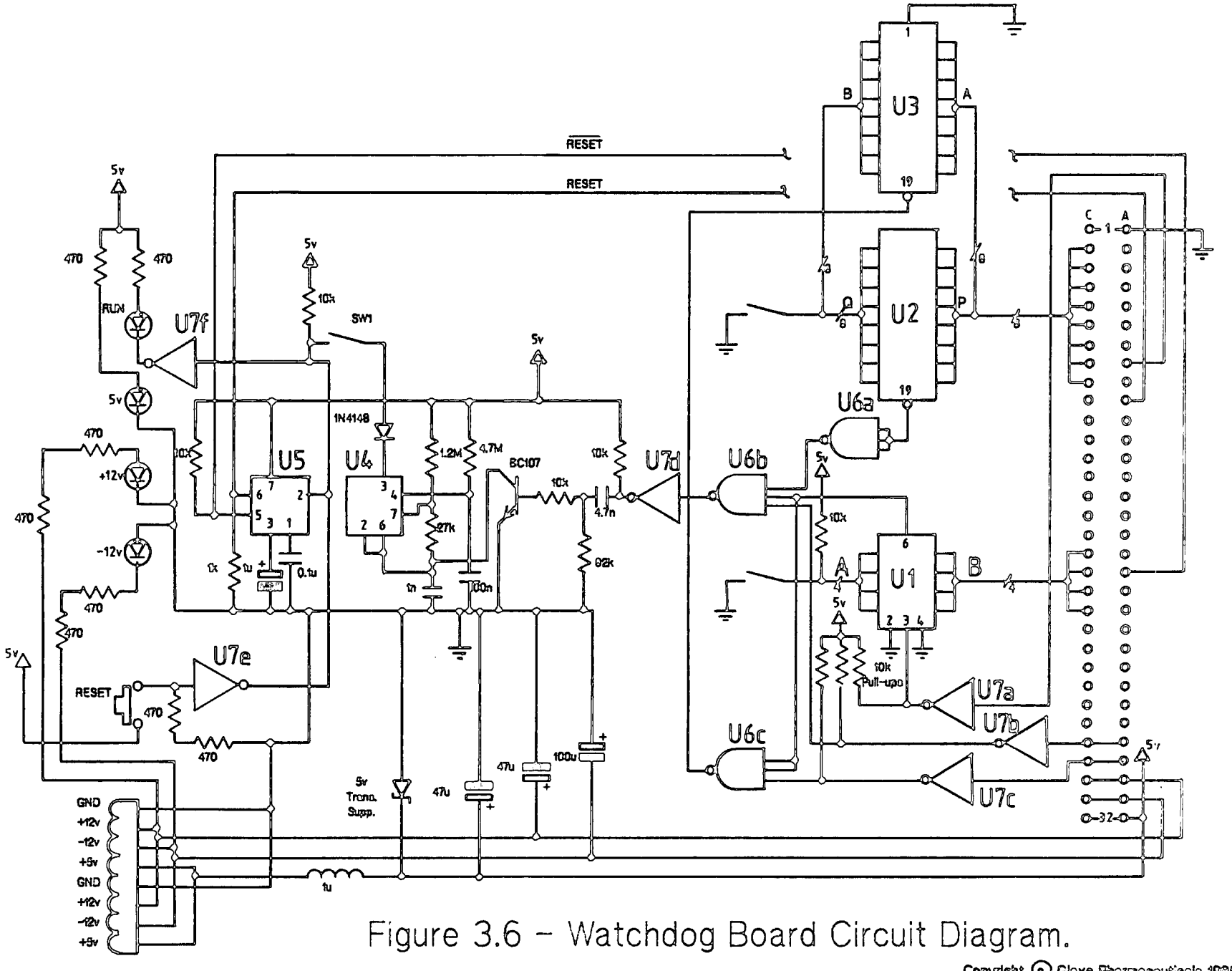


Figure 3.6 – Watchdog Board Circuit Diagram.

3.4.1. Power input

Because of the presence of RS232 buffers, ADC's (Analog to digital converters) etc., etc., on certain cards in the system, the original single-supply requirement was relaxed. Power on this system is derived from an external commercial switch-mode power supply, which by it's nature offers further buffering against mains-born noise. The outputs of the power supply (see Appendix 10 for further details) are +5v, +12V, -12V and Common, which are admitted to the system by a cranked plug (Cannon type AXR-3-15) inserted into a mating 4-way socket (Cannon type AXR-3-32), flush-mounted on the front panel. The minimal depth of the plug ensures that the total depth of the whole assembly is not excessive with respect to the internal depth of common wall-mounting industrial enclosures. The circuit diagram of the card (Fig. 3.6.) reveals that the 5v supply is conditioned in the same way as in the pilot scheme; the other supplies are less critical and are simply decoupled (47 micro-Farad capacitors) before being routed to the backplane.

The operation of the power supply is monitored on the front panel of the card by LED's (light emitting diodes); also present on the front panel are a push-button, LED and switch which are associated with the manual reset and watchdog enable functions, described later.

3.4.2. Watchdog Circuits

This circuit (Ref. 3) is similar to the '555' timer circuit used in the original 'node' system (see Chapter 2); the same system is used - a train of pulses are used to retrigger the 555 (in this case configured as an astable) - but the source of trigger pulses is different. The CPU must source the pulses in the form of data bytes on the Bus - the watchdog card is memory mapped in the same manner as the DVI card, for example. (Rather than repeat the description of the '85 comparator circuitry, the author will refer the reader to the DVI board circuit description in section 3.3.1. in future). When the card is selected by a 'MOVX' write command within the CPU, the eight-bit '682 comparator (U2) compares the data byte on the bus with the settings on an eight-way switch register (note that the pull-ups for the switches are incorporated within the '682), and outputs an active LOW signal if equality is detected. The output pulse is buffered to the '555' (U4) retrigger input; active LOW output pulses from the '555' are diode-coupled to the RESIN input of U5 (see below).A manual push button ('RESET') is provided on the front panel, with LED indication; a toggle switch (SW1) is also provided, to disable the watchdog function during system testing.

The '555' used on this board is a variant of the 'standard' 555 timer, the Texas TCL555CD. This is a new product, a low-power CMOS type, allowing the use

of smaller value timing components to achieve the corresponding decay periods. This adds to both the compactness of the circuit (the timing capacitor is quoted representatively in units of nano-Farads rather than, say, micro-Farads and hence is relatively small and stable) and to its timing range.

An extra chip (U5 - Texas TL7705) has been included on the Watchdog board to monitor the 5V power supply rail and detect 'brown-outs' or major supply fluctuations. The chip has a 'SENSE' input (pin 7) which is, in this case, connected directly to the 5V supply line. The SENSE input will detect supply fluctuations and the chip will output a RESET pulse to reset the active devices connected to the backplane reset line(s), the output pulse timing being determined by an external (pin 3) capacitor. The chip provides an active LOW open - collector output via an internal n-p-n transistor and an active HIGH open-collector output via an internal p-n-p transistor. These outputs are provided with external pull-up/pull-down resistors and connect directly to the backplane reset lines. The reset function may also be triggered by an active low input pulse on pin 2 (RESIN), which may be sourced from the watchdog output (described earlier) or the buffered (U7e) RESET push-button.

3.4.3. Node Identity

If the interfacing system is to be a 'node' in a

distributed system, the node identity (it's number in the system) needs to be accessible to the CPU firmware. For the pilot scheme, the node number was stored as a constant in EPROM inside the 8751, but this is inconvenient for the user, as it is necessary to 'blow' (i.e., EPROM program) different versions of the firmware according to the node's identity.

To cater for convenient access to the node number, and ease of visual discrimination, the comparator switches also double as node 'identity' switches; when the CPU firmware uses a MOVX instruction to 'read' from the card memory-mapped address location, the address decode logic will enable - via a '245 buffer (U3) - the data onto the bus. The 8751 firmware would perform this access as part of its initial RESET sequence during start-up, and incorporate the node number into its self-test calculations; at a later stage the regenerated node number would be written back to the watchdog card to service the watchdog timer.

3.5. Review

After an initial DC check-out, the three boards described in this chapter were tested in combination, plugged into the backplane of the sub-rack. The PDS system's EMV51 emulator was used in lieu of the 8751 to enable rapid de-bugging. Most of the testing was performed using the emulator's high-level macro language, which enabled a convenient and thorough system test.

3.5.1. CPU board

The CPU board was tested with both the 3.6864 MHz crystal and the 7.3728 MHz crystal. No problems were experienced at either speed with signal crosstalk. Particularly pleasing were the CLOCK 1 and CLOCK 2 signals, which exhibited a clean square waveform when viewed on an oscilloscope (Phillips PM3215). The waveform (see Appendix 1) displayed a 50% duty cycle with minimal ringing. The author tested the 8751's response to interrupts using the DVI board's options (see 3.5.2. below). Response to reset etc. is dealt with under 3.5.3.

3.5.2. The DVI Board

Problems were experienced with this board concerning the values of the pull-ups; the board did, in fact, perform satisfactorily from the user's point of view - i.e., it functioned correctly as a system component as originally designed. However, a logic probe placed on the U5 inputs indicated an invalid signal logic level for the theoretical logic '1' state - an effect caused by the pull-down action of the current flowing into U4's internal base resistors. Substituting 3.3k or 4.7k resistors for the 10k pull-ups solved the problem.

The introduction of lower value pull-ups also speeded up the rise-time of pulses associated with open-collector ('LS05) gates - although the exponentially rising waveforms observed during testing

had no detrimental effect on the operation of the logic. Later (production) versions of the board used SIL and DIL resistor networks to minimize the component population and improve reliability; this is true also of other boards in the system, which is why the reader will sometimes encounter, for example, a pair of series resistors (e.g. Figure 3.6) in a circuit where a single larger-valued component would suffice.

3.5.3. Watchdog Board

This board was tested in several 'breadboard' versions and at the time of writing a production design has yet to be finalized, although the version shown in Figure 3.6. performed satisfactorily. Tests were made with the CPU board to check the correct operation of the reset, node identity and watchdog functions. Although the watchdog circuit was found to function correctly, the author was unhappy about the large number of discrete components associated with the watchdog circuit, giving a cluttered PCB layout compared with other boards in the system. Also, the presence of many discrete components on a PCB increases assembly effort/cost and reduces reliability.

The author has experimented in replacing the 555 circuit with a second TL705 chip - configured similar to U5 but used as a watchdog timer. To use a TL7705 as a watchdog requires the use of the RESIN input as the input of the retrigger pulses; in this mode the

TL705 is pulsed into the reset mode and held there - an absence of pulses causes the device to revert to its quiescent state, causing an output pulse on pins 5 and 6. This unconventional use of the TL7705 is still experimental, but offers clear advantages in circuit layout.

CHAPTER 4 : MEMORY EXPANSION

Although the 8751's 4k of EPROM and 128 bytes of RAM are sufficient for many applications, the small quantity of RAM (in particular) will be too limited for some of the Company's applications. It is a common need to buffer slow peripherals - especially devices such as serial printers - with large (e.g. 2k) RAM buffers. On the other hand, any local 'number-crunching' tasks that the system may be called upon to perform may prove demanding as regards EPROM space. The 8751's address architecture allows for both data and program memory to be expanded externally, each to 64k maximum.

This chapter describes the RAM board in detail and discusses EPROM memory expansion; as the Company does not require the development of an EPROM board at the time of writing, the author has limited his activities to providing the necessary backplane signals, thus allowing the addition of the board at a later date. As there are obvious similarities between the two memory systems, many points concerning circuit design that are covered in the description of the RAM board apply also to the design of an EPROM board.

4.1. The RAM Memory Board

Memory systems for microprocessor data storage are normally based on one of two types of memory: 'static' RAM's or 'dynamic' RAM's (often referred to as SRAM's or DRAM's). SRAM's are usually preferred for small (up to 64k) memory stacks, as they are totally self-contained. They are however, more expensive to manufacture per storage element than DRAM's. The major drawback with DRAM memory is the need to 'refresh' the memory - i.e., read (and write back) every bit in the memory at the 'refresh rate' quoted by the I.C. manufacturer.

Refresh control is a task unsuitable for a busy CPU, and is often implemented with special refresh and 'arbiter' (it is necessary to resolve the conflicts that arise from simultaneous refresh and data requests) circuits. The overheads in cost and size that the circuitry incurs prohibits the economic use of DRAM memory in small systems. Thus, SRAM's are the obvious choice for many systems designers. However, another alternative has recently become available - IRAM's ('integrated' RAM's - also known as 'pseudostatic' RAM's), which offer a cheaper alternative to SRAM's.

IRAM's are DRAM memories included on the same chip together with the necessary refresh and arbiter circuits. To the user the external characteristics make IRAM'S appear to be compatible with SRAM'S - that is, they occupy the same JEDEC (Reference 10) International Memory Site for 'byte-wide' memories as SRAM's, and share the same ease of application. The major difference between the two products lies in the interpretation of one all-important parameter - Access Time.

When the 8751 issues a 'MOVX' instruction for the purpose of reading data from external RAM memory, the read timing profiles shown in Figure 3.3. apply. ALE (which runs at half normal speed during a MOVX instruction) is used as the main timing pulse, the memory (SRAM or IRAM) being enabled by (and, therefore, timed from) either the rising or falling edge. In the case of the Intel 2186 asynchronous IRAM, the falling edge is used. (The 2186 will be used as a convenient model for discussion, for the time being). The RD signal is used to gate the IRAM's output buffers onto the Bus; although the data need not be available from the RAM at this point, it must arrive

before the 8751's read sampling interval, which occurs during the eleventh clock period after ALE goes LOW, as shown in the diagram. Thus, the critical timing (Tc) for the 2186 is the time between the falling edge of ALE and the start of the sampling period (SP); this represents ten complete clock cycles. Table 4.1. below shows the timing figures for Tc at three different clock frequencies, and the access times for three different versions of the 2186 (-25,-30,-35 indicates 250nS, 300nS and 350nS respectively. These are the access times without refresh contention).

Clock freq. (MHz)	Tc (nS)	Access Time with refresh 2186-25	2186-30	2186-35	(nS)
3.6864	1898.9	675	800	950	
7.3728	949.4	"	"	"	
11.0592	632.9	"	"	"	

Table 4.1 - Access Time (with refresh) for the 2186 IRAM.

From the table above, for example, it can be seen that the 350nS part is (marginally) too slow for the 7.3728 MHz clock rate, so the 300nS version would be the correct choice. The 350nS part is suitable for the slower (3.6864 MHz) rate, and, of course, is cheapest. The author experimented during development with both of these versions of the IRAM, and also with an 8k x 8 bit SRAM's, (e.g., the Hitachi HM6264 - See Appendix 11). The latter part was found to be satisfactory (the access time of 150nS made it a possible contender for use as a program memory store), but was significantly more expensive than the 2186, which was therefore preferred for data storage.

Figure 4.1. shows the circuit for the RAM memory board, based on the 2186-30 part. It is expected by the author (as a result of enquiries) that Intel will double the size of the IRAM within the next

12 months (the first of the current series of Intel IRAM's was released as a 4k x 8 bit part, which was subsequently replaced by the 8k x 8 bit part); to allow for this links have been incorporated into the design to cater for the anticipated upgrade. The circuit is based on a stack of four IRAM's (U3, U4, U5 and U6), giving a total memory capability of 32k x 8 bit board. The memory may be mapped into either the 0-32k or 32k-64k address range, using further links. Thus, if 64k of memory is needed, then it is necessary to use two boards, until the upgraded IRAM becomes available. The circuit is now described in detail.

4.1.1. Chip enable Gating

A 2186 memory cycle commences when Chip Enable (CE - pin 20) is driven LOW. The signal that provides the transition is, as indicated earlier, ALE - but it is necessary to gate the signal with others (Reference 11) to prevent illicit access, particularly during power-up. The IRAM is self-resetting when the 5 volt supply is applied, provided CE, OE (output enable - active LOW) and WE (write enable - active LOW) are inactive. This is certainly true in the case of the last two named signals, but ALE needs to be gated out, as it commences clocking immediately the 8751 is initialized. Pulsing CE before the 2186 has terminated its internal reset procedure would result in incorrect operation of the device, so ALE is gated with RESET to prevent this occurrence. The signal is also gated with the memory/peripheral select line, to inhibit access during peripheral transfers via the Bus (see the previous chapter). Table 1 below shows the

truth table for the chip enable gating, which is implemented with NAND gates (Figure 4.1.- U1a, b and c).

M	R	A	Q	
0	0	0	1	
0	0	1	1	
0	1	0	1	M=Memory Enable
0	1	1	1	R=(NOT) Reset
1	0	0	0	A=(NOT) ALE
1	0	1	1	
1	1	0	1	
1	1	1	1	

Table 4.2. - The Truth Table for Gating ALE

A two-to-four line decoder (U2) is used to select the individual 2186's. The open collector version ('LS156) is used to allow the versatility in IRAM component choice mentioned earlier. When the 8k RAM's are used the Sl and Sh inputs of the '156 select the 0-32k or 32k-64k range; wire links are employed to configure the inputs to the '156. Only the 'A' section of the '156 would be used with the 16k devices, and only one half of the '156 is ever enabled, according to the option selected - the appropriate link combinations are shown on the circuit diagram.

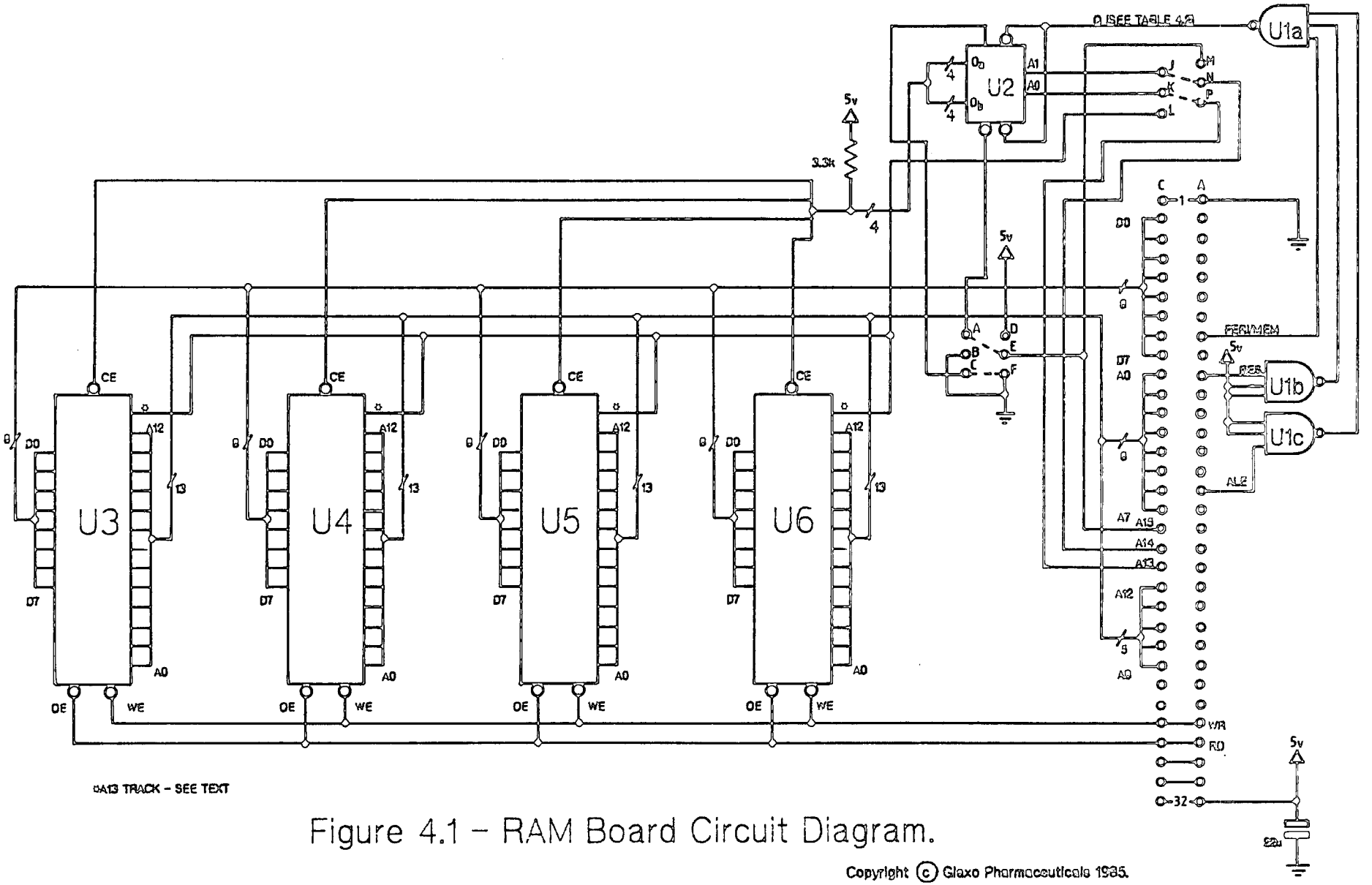


Figure 4.1 – RAM Board Circuit Diagram.

When the card is linked for the 0-32k range, A13 and A14 are connected to the A0 and A1 inputs to the decoder; A15 is connected to the S1 input of the 'B' section of the '156 the 'A' section being disabled, with Sh held LOW. The gating of the enable to the decoder us deferred until U1a pulses LOW - on the falling edge of ALE, whereupon the selected CE is driven LOW, and the memory cycle for the chip commences.

4.1.2. Reading from data memory

A data memory read cycle commences with a negative-going CE transition as outlined above. The address information on lines A0-A12 is latched into the IRAM at this time. The IRAM then accesses the addressed data byte as part of a standard 'read-modify-write' cycle, as is common with other types of read/write memories. Within the time-period detailed above, the data is present in the RAM's output buffer, to be gated to the Bus by RD (which is connected to OE - pin 22). The rest of the read-modify-write cycle is internal to the 2186 - consisting of writing the unmodified data back into the memory location. This part of the cycle is most relevant to WRITE transfers, described below.

4.1.3. Writing to Data Memory

The write timing is less critical than the read timing (see above), the data being latched into the IRAM on the falling edge of WR. The standard R-M-W

cycle commences as for READ transfers (described above), on a negative CE pulse. The latched (by WR) data byte from the Bus is written to the addressed memory location in lieu of the previously fetched byte (hence the 'modify'). Once the data byte has been sampled by the IRAM, the timing of the remainder of the write cycle is in general unimportant to the operation of the host CPU, and it is therefore true to say that the read timing is generally the more limiting factor in the application of RAMs. Further details on the operating conditions of the 2186 and other memories discussed herein can be found in Appendix 11.

4.2. External EPROM Facilities

The author has allowed for the addition of external EPROM by providing the relevant buffered signals (PSEN, together with the existing set of signal lines) on the backplane. Clearly, the design of a separate EPROM board is not a major undertaking; indeed, it is possible to extend the existing RAM board design to so that the board accepts either RAM or EPROM - the use of 16k x 8 bit EPROM's would allow the modified board to accommodate external (to the 8751) firmware programs extending up to 64k in size. The author's development brief from the Company does not allow for such work, so the provision of external EPROM facilities remain a future development.

4.3. Review

The RAM board was soak-tested by repeatedly writing and reading back chequerboard patterns (patterns of alternate noughts and ones) throughout the address range of the memory

array (four 2186's). The circuit was tested in 'breadboard' form, and the board was tested in the rack in conjunction with the CPU (using the EMV51 emulator) and watchdog boards.

Although the circuit components all performed satisfactorily under test, it will probably be necessary to substitute 8k x 8 bit SRAM's (See Appendix 11) for the 2186's in production boards. At the time of writing the 2186 is suffering unfavourable price competition from the SRAM's (See project Review - Chapter 9); the 2186 is otherwise the preferable component - being fast enough for data memory and having in its DRAM structure an inherent low power consumption. Production artworks for the RAM board are under preparation and incorporate the techniques recommended by Intel (Ref. 10) for address/ground/power line tracking - e.g., gridded ground planes - and show an acceptable component/track density, suggesting the board will be suitable for volume production.

During the course of the development of the RAM board the author experimented with a memory-mapped version of the RAM board with an on-board page register which could be loaded via the Bus; this design was discarded after consultation with the users of the system, due to the inconvenience of the 2K paging system adopted in the circuit. The design did have some potential application, though, especially for use as a video data store, as several RAM boards (using different memory map addresses) could be referenced within the same rack. The memory mapping hardware for such a board in basically the familiar 'LS85 comparator circuitry introduced in the previous chapter.

Of relevance to the possible future development of RAM boards for the system is the recent introduction (Reference 30) of hybrid 32k x 8 bit (and also 128 x 8 bit) RAM chips. These parts use surface mounted components (notably four 8k x 8 SRAM's) on a DIL carrier, giving a pin-out conforming to JEDEC standards, and offer the possibility of designing a page addressable RAM board with an 128k x 8 bit (or greater) capacity.

CHAPTER 5 : COMMUNICATIONS INTERFACES

To interface to a variety of commercial devices, a number of boards were designed to support standard protocols. For example, a typical device to be found on or near one of the Company's production lines is a small industrial balance (for testing the weights of samples of specific products) - invariably fitted with either a 'current loop' (CL) interface or a BCD (binary coded decimal) interface. A typical device requiring two-way data transfers is a small (e.g. desk-top) microprocessor, such as the H.P. (Hewlett Packard) 80 Series - possibly used as a local 'host' for a data logging application. This device would be interfaced to the system using an HPIB (Hewlett Packard Interface Bus) interface as first choice, as the HPIB is an integral (HP86) interface, as it is with some other manufacturer's computers.

Probably the commonest communications interface protocol is RS232 (Ref. 12), the only protocol supported on many types of microprocessor; it was therefore essential to develop a board to interface to RS232. Each of the boards designed to enable the user of the system to interface to the protocols outlined above are now described at length.

5.1. HPIB Board

It would be inappropriate to attempt in this thesis to describe HPIB in detail; the complexity and depth of specification of HPIB can be grasped by examining the International Standard IEEE488 (Ref. 13). The author has included (Appendix 4) a brief overview, together with details of the Bus Adapter chip (Ref. 14,15)

The advent of a number of Bus Adapter chips has of late considerably eased the engineer's task when faced with an HPIB

interface design. Several chip-sets are available (e.g. from Texas Instruments, Motorola, Intel, etc.), but the Fairchild 96LS488 has the unique advantage of not requiring a dedicated host microprocessor - the device has enough on-board intelligence to handle the Bus protocol asynchronously. This allows the user to address the HPIB Bus via the Adapter chip using a simple set of handshake lines, facilitating read/write and status interfacing.

The 96LS488 has the further advantage of in-built 48 mA Bus buffers for the Management/Data Bus logic. This means that the single Fairchild chip provides a direct interface to the HPIB Bus that is equivalent to the three-chip solutions provided by other vendors - yielding advantages in size, cost and complexity. Indeed, as regards size, the 96LS488 solution was probably the only viable design option for the board area available. A disadvantage of the 96LS488 is its inability to handle the controller function - i.e. it is restricted to talker/listener functions. As the talker/listener mode is the expected operating mode in the applications foreseen for the equipment, this is not seen as a major disadvantage; the design of a separate HPIB controller card is not a difficult task, but such a card is not required by the Company at the present time.

Figure 5.1. shows the circuit of the interface. The full facilities (Appendix 4) of the 96LS488 (U1) have not been exploited, due to in part to the size restriction of the half-Eurocard board, but also because of the fact that they are in excess of the Company's requirements. The mode inputs have been preset to the 'fast talker/listener option', which is the

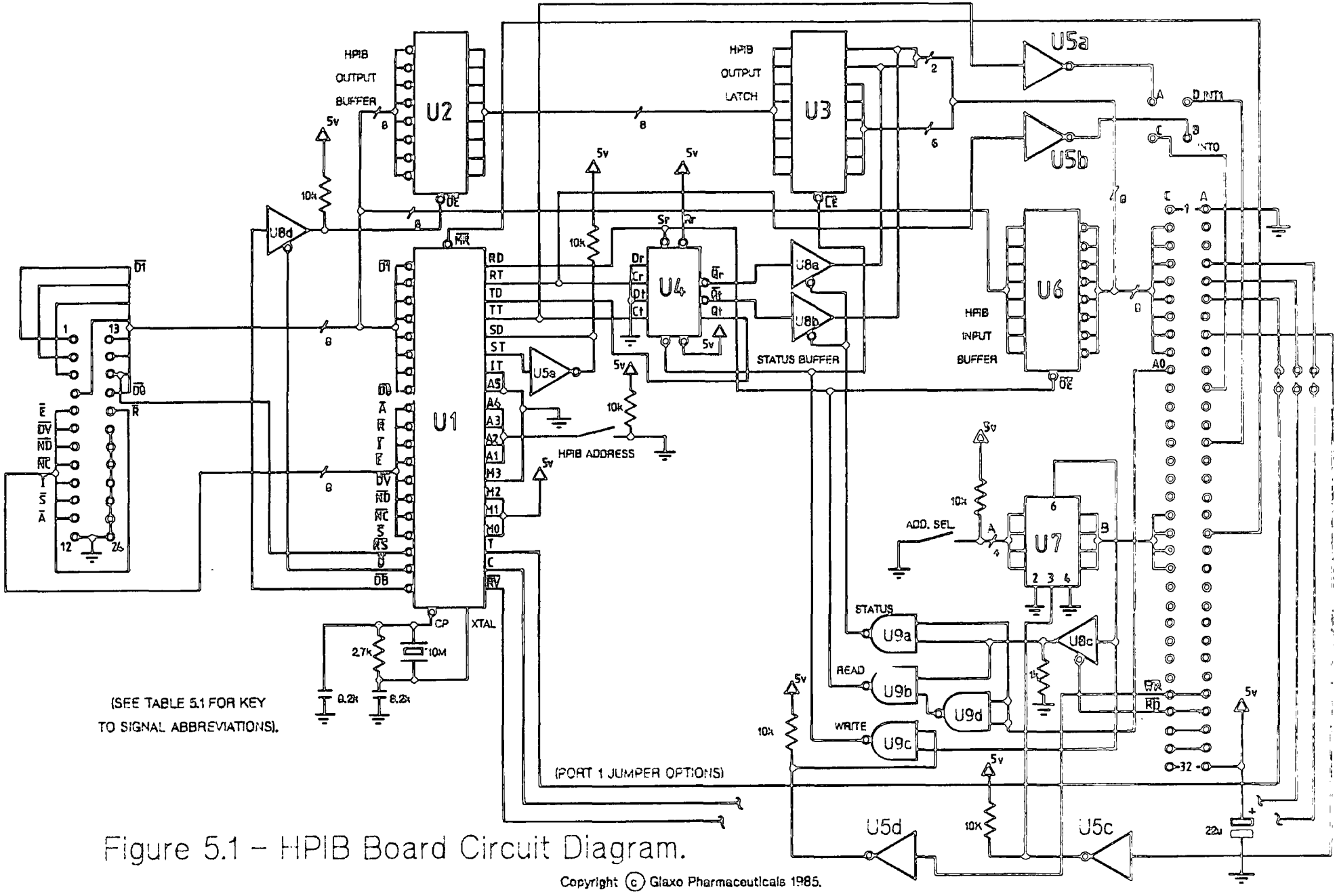


Figure 5.1 – HPIB Board Circuit Diagram.

correct choice for the tri-state Data Bus drivers used externally. These are implemented with the 74F240 (U2) package, which has the required 48mA capability as specified in IEEE488. This provides the buffering for data bytes presented to the HPIB when the Talker function is being implemented. The bytes are held in the '574 octal latch (U3) until the HPIB is ready to accept; this is achieved via the triple-wire handshake (Appendix 4), which is handled by the 96LS488 (U1) - the DRB (active LOW) signal enabling the data onto the HPIB. Extra gating has been used (Ref: 14, 15) to eliminate the extra DRB pulses that the '488 emits during serial poll response; these are active LOW pulses used to multiplex status bytes (if required) onto the HPIB, but the author has preferred to implement the simpler IEEE488 option - that is to drive DIO7 in response to serial poll, which the '488 achieves using its RQS output (pin 23).

Circuit Abbreviation	Signal Group	Signal Name
D1 - D8	HPIB Data Bus	DIO1 - DIO8
E	HPIB Man. Bus	EOI
DV	" "	DAV
ND	" "	NRFD
NC	" "	NDAC
I	" "	IFC
S	" "	SRQ
A	" "	ATN
R	" "	REN
RS	Poll response option	RQS
D	Status multiplex	D/S/E
DB	Data enable	DRB
RD	Handshake	RXRDY
RT	" "	RXST
TD	" "	TXRD
TT	" "	TXST
SD	" "	STRD
ST	" "	STST
IT	" "	IST
T	Control	Trigger
C	"	Clear
RV	SRQ Prompt	RSV

Table 5.1. - Key to Signal Abbreviations

The transmit handshake pair (TXReady and TXStrobe) set/reset the transmit handshake latch U4a (74LS74), the other half of the D-type flip-flop (U4b) being used to implement the corresponding receive handshake latch. Open-collector 'LS05 buffers (U5a, U5b) are used to drive the backplane interrupt lines INT0 and INT1; the jumper set provided allows the required combination to be selected. The card is memory mapped (U7/U8c) using the '85 comparator circuit common to the other memory mapped cards already described (see Chapter 3). Gating of WRITE and the '85 output derives the (rising edge) latching pulse for the '574 latch enable input, allowing the data bytes from the backplane Bus to be admitted to the interface.

The data held in the (U3) latch is enabled to the HPIB Bus by U2 when DRB pulses LOW (this occurs when U1 sees the listening device signalling that it is ready for data - via NRFD). As indicated above, extra gating is used (U8d) to inhibit DRB enabling U2 during poll response. A two input OR gate was used on the prototype version of the card to gate DRB and D/S/E, but this required an extra package, and the card is already highly populated. Hence a spare function in the LS125 package has been pressed into service as a logic gate as a pre-production experiment to attempt to ease circuit tracking problems. (Another section of the package - U8c - has been similarly experimentally used as a logic gate to gate RD with the U7 output signal).

The card receives bytes from the HPIB when in Listener mode. A 74LS540 inverting buffer is used to accept bytes from the Bus - it is unnecessary in this case to latch the byte, as it is

the responsibility of the Talker device on the Bus to hold the byte on the Bus until the triple handshake is complete. This will be the case when the 8751 takes the data byte via a backplane READ, which also drives the RXStrobe handshake line; the '488 takes its cue from the RXStrobe (active HIGH) pulse, and completes the HPIB handshake with the Talker device.

The receive and transmit status bits latched by the '74 are enabled onto the bus during a status read operation via a '125 tri-state buffer. The READ signal is gated with A0, thus mapping data (NXX0 - where N is the card address) and status (NXX1) into adjacent memory addresses; using contiguous memory map areas for related peripheral functions is a recurring technique used on several cards in the system, including the BCD and UART cards, whose descriptions follow.

Plate 5.1. shows a photograph of a prototype HPIB board. The visible components on the front panel are the 24-way IEEE 488 connector and the rotary hexadecimal switch used for HPIB address selection. Early prototype boards used discrete transistors to accomplish the open - collector interrupt functions implemented by U5a and U5b in figure 5.1. - this is generally true to most boards in the system. The pre-production designs published in this thesis are the embodiment of a consistent philosophy - elimination of discrete circuit components wherever possible, minimization of circuit components, and optimization of tracking densities on all production PCB's.

5.2. BCD Board

The BCD board (Plate 5.2.) features it's own single-chip

microprocessor, an Intel 8748. The author decided on this design approach after considering and rejecting various other circuit configurations involving 'semi-intelligent' chips such as PIA's (Ref. 16) and suchlike. The requirement was for an interface capable of handling the variety of BCD output devices that are currently manufactured. These items of equipment often feature thirty or more signal output lines which can release their data simultaneously - consider, for example, eight BCD digits requiring four lines per digit - a total of thirty-two output lines. Alternatively, the equipment may have been designed to emit the data in 'bit-parallel-byte-serial' form, in which event the number of signal lines would be relatively small.

Every BCD device is equipped with handshake lines; often there is a 'flag' line (sometimes more than one) to indicate to the 'host' that the data is available on the data lines. The flag line may be either active HIGH or LOW, pulsed, or held at a logic state for the data duration - also the flag line transition may occur before or during the period the data is valid. A handshake line operating in the reverse direction (an output from the host) - usually termed a 'control' line - may also be required, to tell the device to perform a function and/or release data. The control line may drive an active HIGH or LOW signal, and it's timing requirements may be as varied as those of the flag line. To allow for all permutations of logic levels etc., the interface needs the uncommitted logic of a microprocessor, which also can handle the buffering of the data and the data exchange with the 8751.

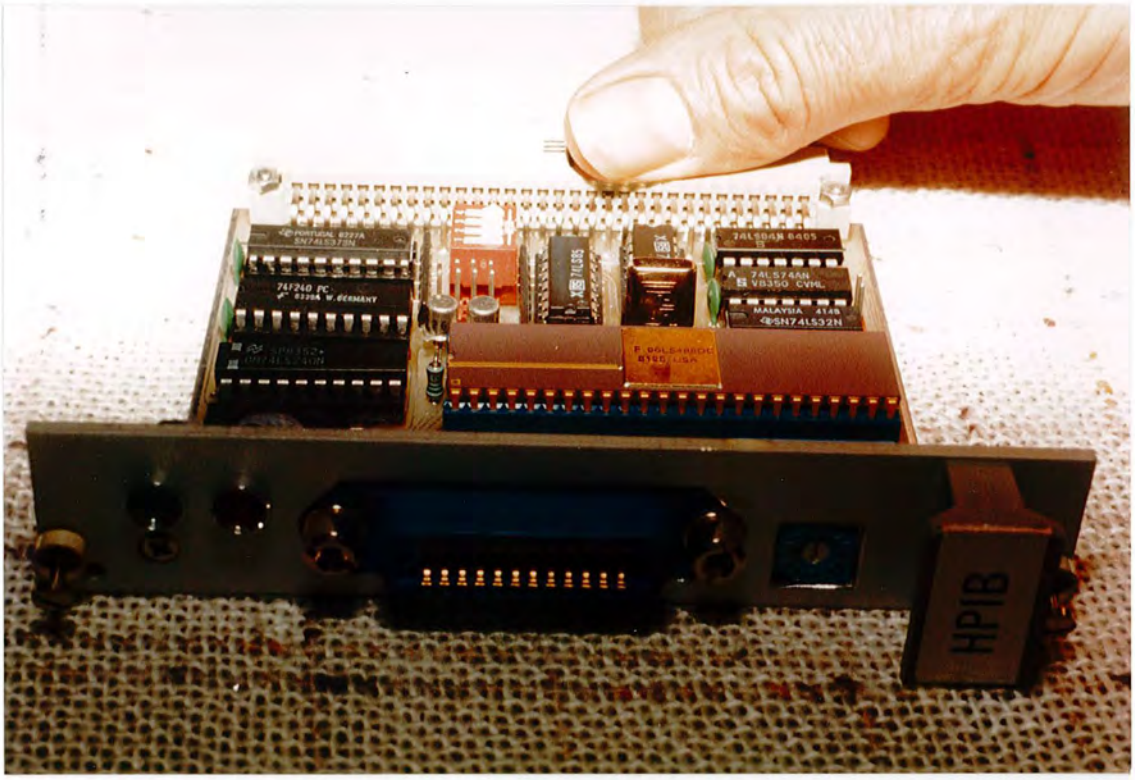


Plate 5.1. (above) -

The HPIB Board

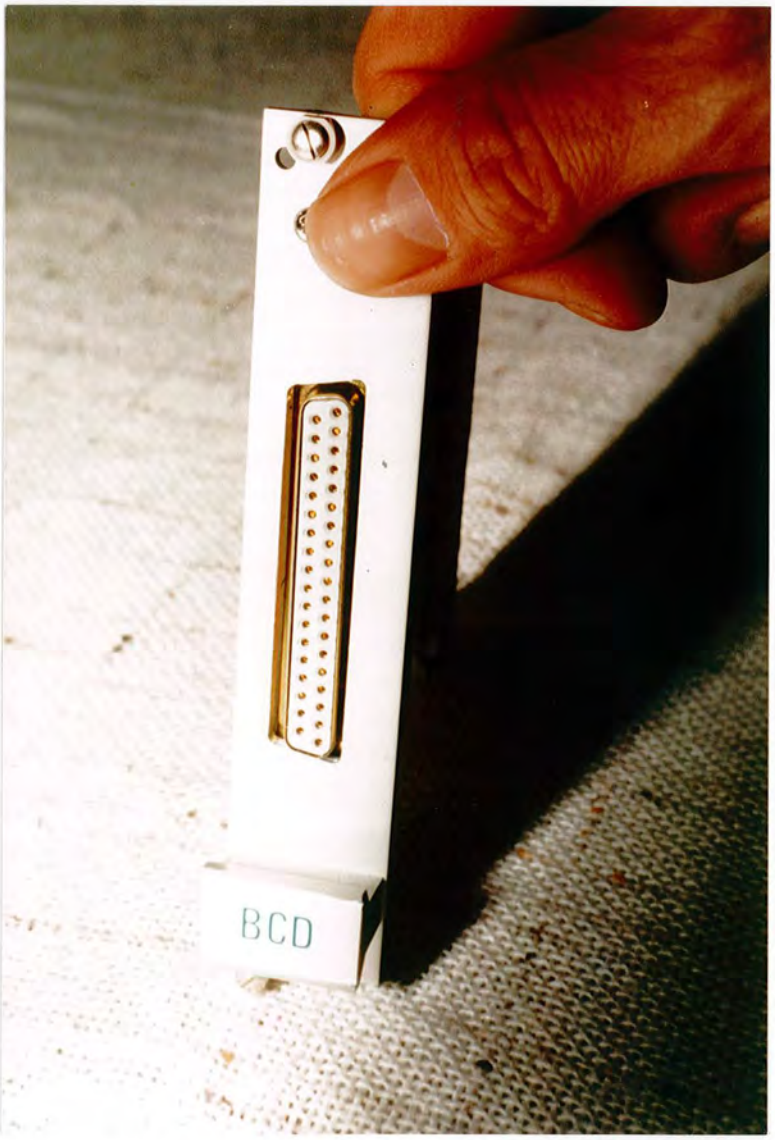


Plate 5.2. (right) -
The BCD Board

The 8748, the forerunner of the 8751, remains an important single-chip microprocessor in its own right - being still the first choice of many designers for small low-cost systems. This is mainly due to the second-sourcing of the device - bringing the unit price down to less than fifteen pounds (1984). The microprocessor (Appendix 5) has similar (but less comprehensive) features to those of the 8751 - the most obvious differences being the amount of EPROM (1K), number of ports (3) and lack of serial I/O. The ability to extend the device's I/O facilities using the associated 8243 I/O expander greatly enhances the 8748's range of application. The 8748 uses part of one of its ports (port 2) as an expander Bus; the four-bit Expansion Bus allows 8243's to be addressed using special instructions within the 8748 which make the 8243's appear as a transparent extension of the 8748's I/O.

Figure 5.2. shows the circuit design of the BCD board. Each 8243 has four four-bit ports which may be individually addressed by firmware within the 8748. Two (U2, U3) 8243's have been assigned to the 8748 (U1) in this application, allowing a range of eight BCD digits to be read simultaneously - if required. The chip enables for the 8243's are driven by 8748 port pins, timing signals being emitted automatically on the PROG pin. The front panel of the BCD card houses a thirty-seven way 'D' -type socket, thirty-two pins of which are the input connections for the expander chips; two of the remainder are connected to Ground and the 5v line, the rest being handshake signal lines. Two buffered (input) flag lines (U7e, U7f) are provided, and one buffered (output) control line (U7d).

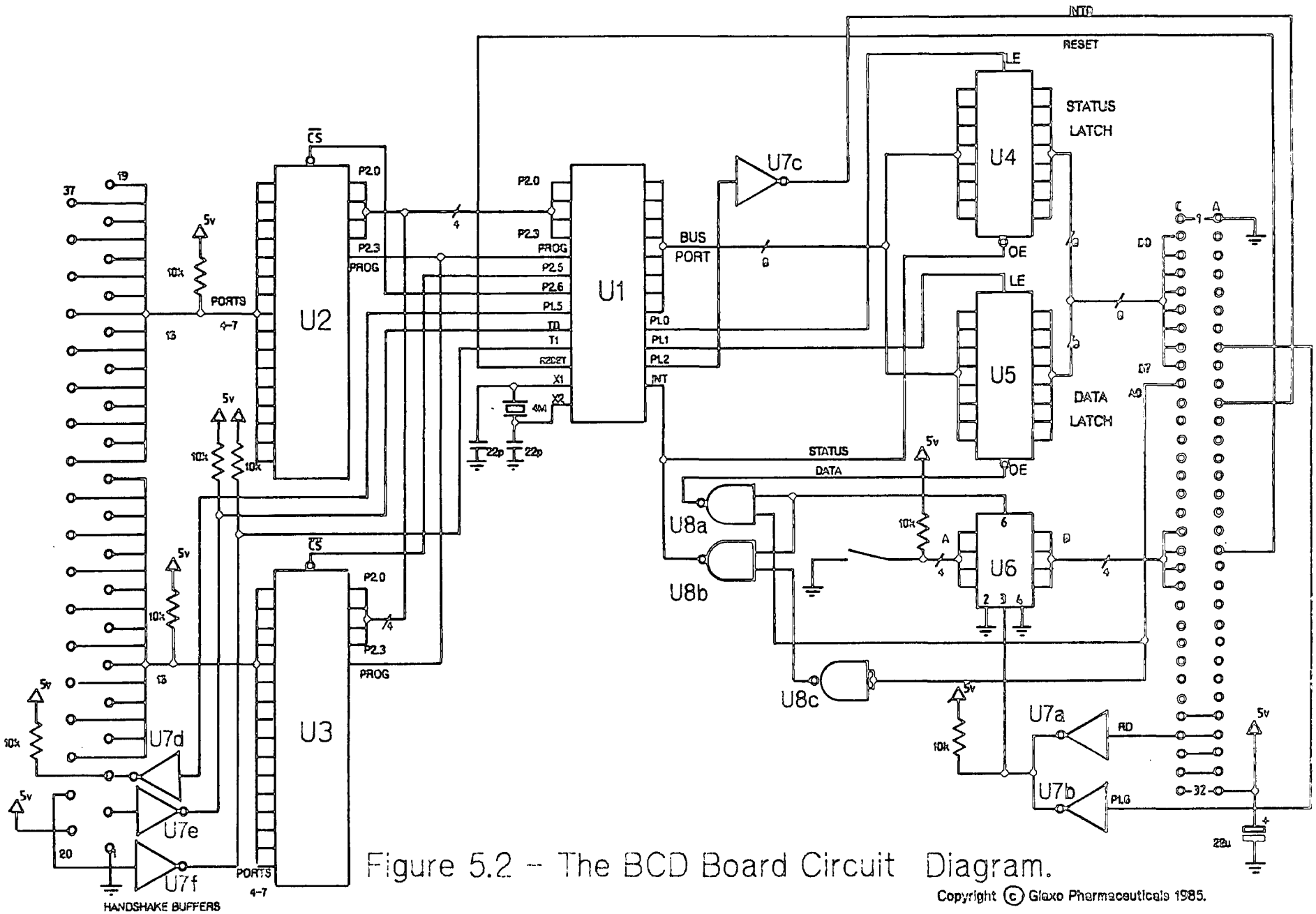


Figure 5.2 - The BCD Board Circuit Diagram.

The Bus port of the 8748 (port 0) is used as an output port (NOT as a Bus, but in its alternative usage as a conventional port) to drive data into the two '574's, which are employed as latches (U4, U5) for status and data bytes. Data to be latched is first loaded into the port 0 output latches within the 8748, before the appropriate latch enable is taken through a low-to-high transition; the latch enables are driven via spare port pins, as for the 8243's. The eight-bit latches are preferred to four-bit latches so that BCD digits may be passed to the 8751 in pairs rather than singly, thus minimizing the burden on the 8751 interrupt handling firmware; the BCD interface forces an interrupt of the 8751 after it has updated the data in the latches. The status latch is used to pass the byte count to the 8751 and to indicate (status bit seven) that it was the device originating the interrupt. The interrupt is driven by an 8748 port pin, which is buffered by an open-collector inverter (U7c) to the INT0 interrupt line on the backplane, lack of board space inhibiting the use of jumpers to allow the user a choice of interrupt lines.

When the 8751 is interrupted by the 8748 it polls the status registers (serial poll) of the devices within the rack until it locates the device that originated the interrupt. In the case of the BCD interface, the 8751's next action would be to read the data byte; as the output of U8b (Status add. decode) is also wired to the interrupt pin of the 8748, this initiates an 8748 interrupt, thus informing the device that it is time to update the latches, i.e. set up the next status byte. A description testing of the firmware for the 8748 can be found

in Appendix 6, together with a description and flow diagram relating to 8748/8751 interaction.

5.3. The UART Board

The UART board is one of three serial interfaces developed for the system (descriptions of the RF Board and the Serial board can be found in Chapter 6; the latter interfaces are both slaves to the serial port). It allows the user to address a number of serial devices other than the single device addressable via the serial port. With the aid of a number of UART cards the system user can conveniently implement, for example, a serial multiplexer. A circuit diagram of the UART card is shown in Figure 5.3; the interface supports both RS232 and 20 mA current loop, the latter being fully configurable to either active or passive loop on both transmit and receive lines. Appendix 9 details how the various options may be selected on the card.

5.3.1. The 8251A USART

The 8251A device is an industry standard 8085-Bus-type serial device originally developed by Intel, but now second-sourced by several companies - NEC parts were used for the prototype tests. Although the 8251 is officially designated a USART (Universal Synchronous/Asynchronous Receiver-Transmitter), the author, after discussions with staff in the Company's Computer Services Department, reached the conclusion that the synchronous facilities offered by the device were not required; therefore this part of the USART remains unused within the interface design, and the

device is referred to as a UART for simplicity. The 8251A is designated U1 in the circuit diagram; being a standard 8085-type peripheral, it is a simple matter to interface the UART to the system Bus.

Unlike, for example, the 6402, the UART is a software configurable device, the configuration being achieved by loading its internal Bus-addressable command registers via a series of commands from the host processor (8751) - full details of these commands and other aspects of the operation of the UART, and facilities available, can be found in the device's data sheets (Ref. 21) and the Intel Application Note (Ref. 22). The status registers inside the UART may be read by the host's firmware at any time, to determine the current state of the device. A control line, assigned to address line A0, is used to select transfers of data or status/commands (according to the state of the read/write lines) as shown in Table 5.2. below:

A0	R	W	Interpretation
0	0	1	Read data
0	1	0	Write data
1	0	1	Read status
1	1	0	Write Command

Table 5.1. - Showing the use of A0 to select registers within the 8251A

5.3.2. 8251A-8751 Bus interface

The interfacing of the device to the 8751 is, in truth, a straight-forward implementation of the

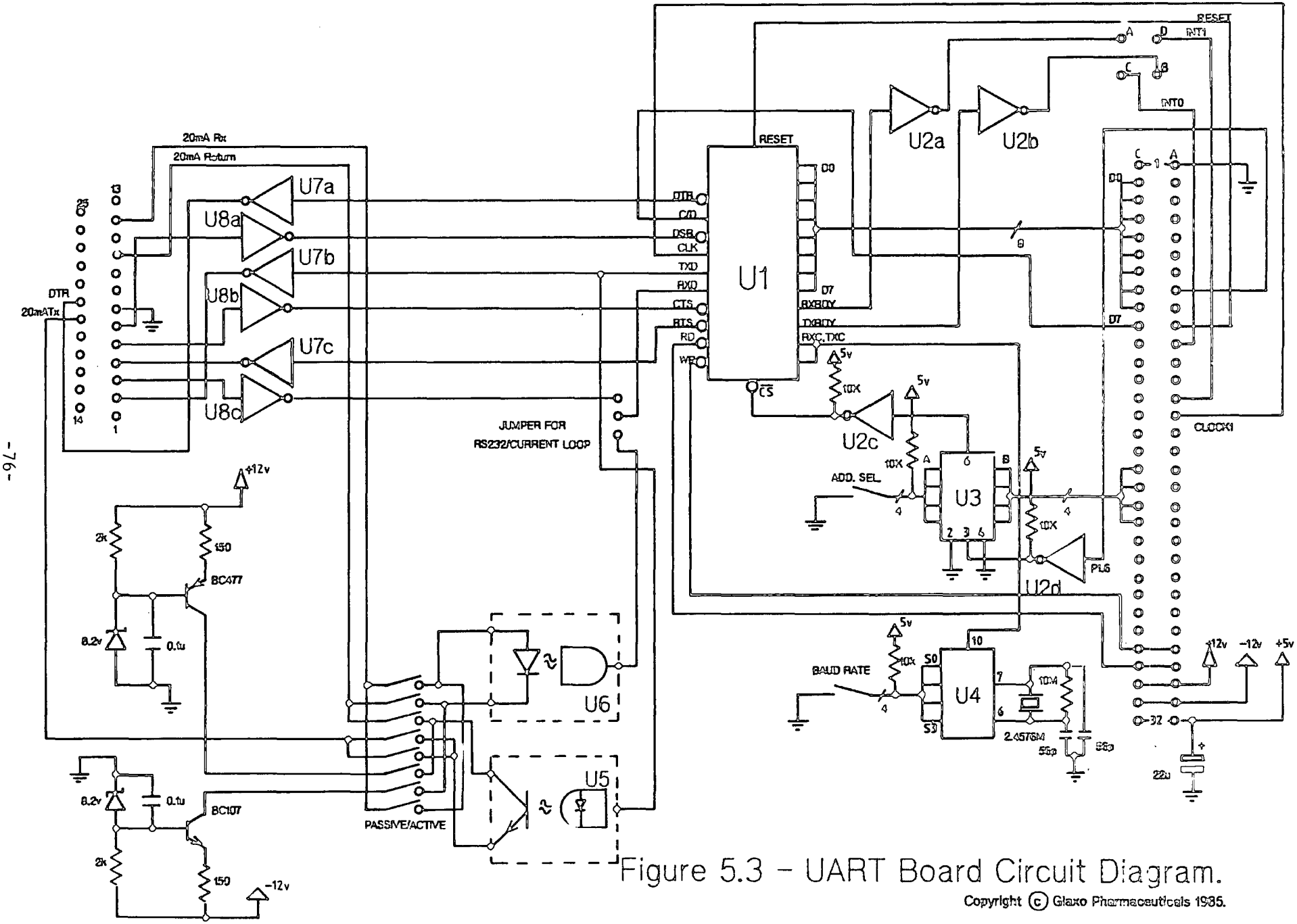


Figure 5.3 – UART Board Circuit Diagram.

Application Note listed above, the 8751 having the 8085 Bus structure as related earlier in this thesis.

The same '85 (U3/U2c) comparator circuits (covered in the DVI Board description, Chapter 3) as used on other memory-mapped boards are used to select the UART, when addressed. Also, in common with circuitry on other boards, the backplane signal on P1.6 (peripheral/memory select) is inverted (U2d) and used to enable the comparator. Transmit/Receive Ready pins drive the backplane interrupt lines (INT0/INT1 - selected by jumpers) via open collector buffers (U2a/U2b).

5.3.3. Baud rate generation

The board rate is selected via a switch bank (see Table 5.3. below) connected to an industry-standard 4702 baud rate generator (U4), allowing the user to operate in asynchronous full-duplex mode at rates up to 9600 Kilobaud.

S3	S2	S1	S0	Baud rate
0	0	0	0	/
0	0	0	1	/
0	0	1	0	50
0	0	1	1	75
0	1	0	0	134.5
0	1	0	1	200
0	1	1	0	600
0	1	1	1	2400
1	0	0	0	9600
1	0	0	1	4800
1	0	1	0	1800
1	0	1	1	1200
1	1	0	0	2400
1	1	0	1	300
1	1	1	0	150
1	1	1	1	110

Table 5.2. - Baud rate selection

The baud rate generator's (16x) output is used clock both the transmit and the receive circuits of the UART - lack of board space prohibiting independent clock circuits. Thus the UART board cannot be used for applications where differing clock rates are required on the receive and transmit lines. This is a rare requirement, and is not seen as a major limitation.

5.3.4. Handshake lines

The UART supports the primary (CTS/RTS) and secondary (DTR/DSR) handshake signals associated with RS232; these I/O lines may be examined or modified (where appropriate) via the 8251A's internal status/command registers. Details of these registers are given in Appendix 9 - which also deals with the initialization of the device. The Company's use for the UART board on projects requiring serial interfacing with handshaking are confined to RS232 - hence the handshake lines have not been provided with current loop options. The options are allowed for only on the serial data lines, the common use being to communicate with a small (e.g. Mettler PL200) industrial balance via current loop (current loop has a superior noise immunity relative to RS232), such a balance either requiring no handshaking whatsoever, or providing a 'software handshake' with control characters (such as XON/XOFF) in the data stream.

5.3.5. RS232/Current Loop Drivers

As explained earlier, the interface can support both current loop and RS232 protocols. The driver circuits are common to both the UART and the Serial Boards and are covered in detail in the next chapter, which deals with the Serial Board (section 6.1) and the RF Board. The Serial board lacks the ability to support the RS232 secondary handshake (DTR/DSR), although it does support the primary handshake (CTS/RTS). Hence the 8251A's secondary handshake lines are also provided with RS232 buffers; the buffers used are Fairchild ul488 (U8) and ul489 (U9) types (Ref: 23), which are industry-standard parts available from a number of alterative sources. The current loop transmitter (U5, HCPL-4100) and receiver (U6, HCPL-4200) circuits are covered in the next chapter, section 6.1.2.

5.4. Review

All the communications boards were tested in prototype form in combination with the CPU and watchdog boards, using the PDS's emulator to monitor the testing. Some use was also made of the serial Board (Chapter 6) as detailed below. All boards were given a DC check-out before system testing commenced.

5.4.1. HPIB Board

An HP85 desk top computer was used as a host device for the HPIB tests, being capable of supporting the controller/talker/listener modes of operation. Figure 5.4. shows the physical connections of the various devices used for testing. The HPIB board was tested

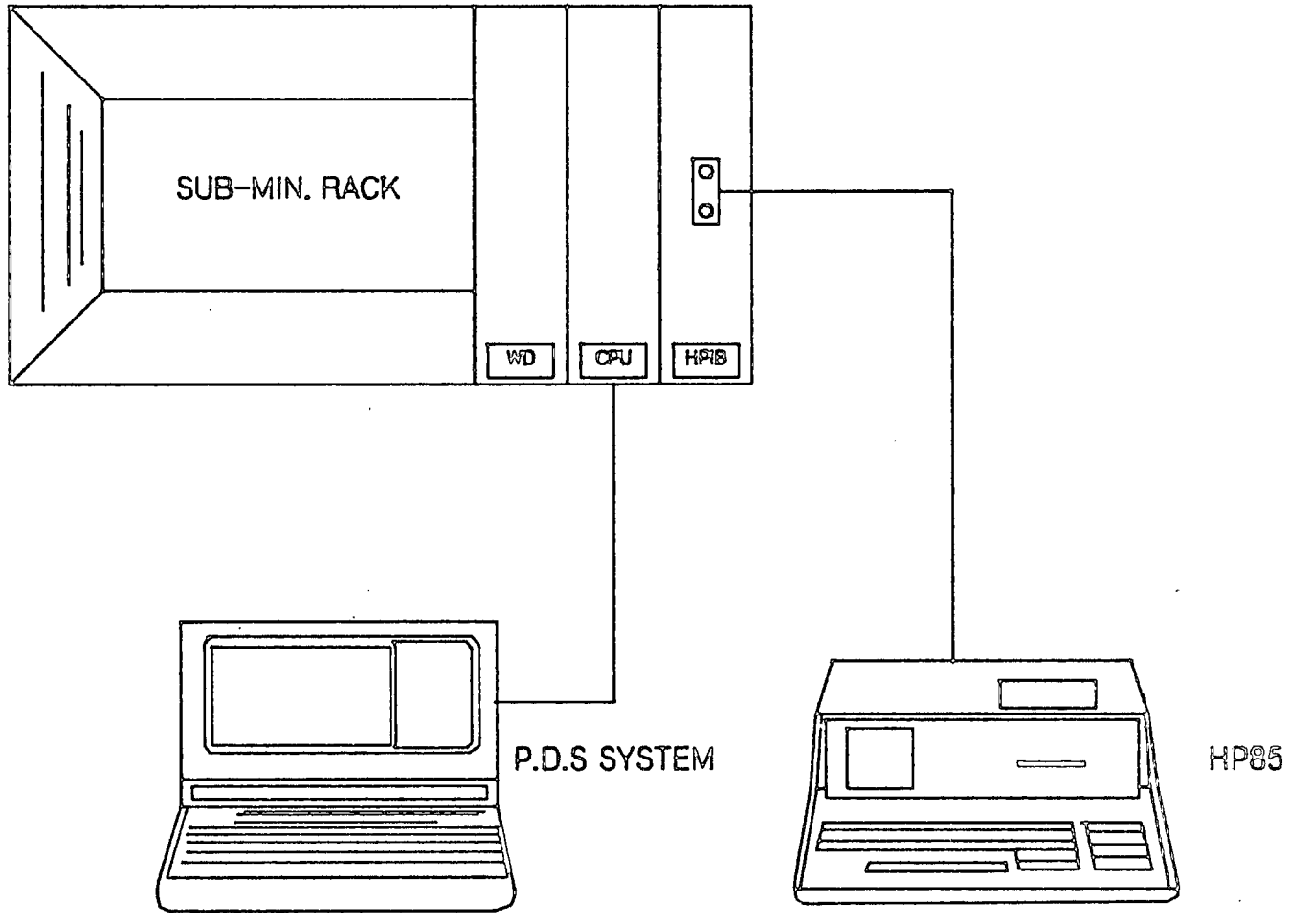


Figure 5.4 - Schematic Diagram showing HPIB Board Testing.

for its talker/listener functions and for its handling of clear, trigger and SRQ options.

5.4.1.1. Talker/listener tests

Because of the efficient triple handshake that is the heart of HPIB (Appendix 4), testing of bidirectional data transfer on the HPIB was relatively straight-forward. To transfer data to the HP85, a small PDS emulator routine was written (using the high-level macro-language); a PBYTE command was used to write a data byte to the HPIB card and the status register was examined using a second PBYTE command. The routine repeatedly wrote data bytes to the HPIB on detection of the 'Tx Ready' state by performing an infinite loop. The HP85 BASIC program consisted of a simple ENTER - and - DISPLAY loop. A compare statement was added to this simple loop at a later stage, to allow soak testing of the board. The modified program checked that consecutive data bytes were identical and counted any errors. Repeated tests used different data patterns designed to check-out all the Bus lines.

Testing of the Listener function of the HPIB interface was essentially the reverse of the above procedure; the HP85's simple

BASIC program consisted of an OUTPUT statement within an infinite loop. (The HP85's participation in the Bus handshake is transparent to the high-level BASIC language). The emulator's program loop consisted of instructions to read the status register (to check for the arrival of characters from the HP85) and to transfer data bytes as they arrived (as indicated by the status bits), displaying the bytes on the screen of the PDS they were received.

The only problem with the testing of the Talker/Listener functions concerned the power-on state of the interface; after power-on the U8a/U8b status bits are incorrect - a problem that could be remedied by applying a power-on reset to the U4, or by issuing a dummy data read in the 8751 host firmware. Otherwise, no logic errors were detected during testing.

5.4.1.2. Miscellaneous

A small assembler routine was written to check the operation of the INT0 and INT1 lines, which are driven by U5a and U5b. Interrupt vectors were used to vector the interrupts to memory traps. The HP85 was used to drive the trigger and clear option lines remotely, and also to test the operation of the SRQ line via the RSV option

on port 1. All the above tests proved satisfactory. As TRIGGER and CLEAR are pulsed signals, they are of marginal value - R/L would be a better choice to either, as a fixed logic level is maintained. SRQ is a very useful HPIB signal, and well worth implementing; it affords a means by which the 8751 may interrupt the host computer.

Despite the efficient operation of the interface, it remains too complex (i.e., the chip density is too high) for mass production and some further circuit rationalization is desirable. One possibility is to replace U2 and U3 with a single component; several CMOS latches are made with high-drive open-collector bipolar output stages which may be suitable.

5.4.2. BCD Board

Lack of an 8748 assembler for the PDS limited the testing of the BCD interface to a check-out of the various logic elements that are present on the board - the 8243's, the 8748, the 574 latches etc. as outlined in Appendix 6. These logical tests allowed the integrity of the interface design to be established. More sophisticated firmware routines for the 8748 will be written at a later date by the staff of Company's Computer Services Department.

The physical layout of the board is dominated by the three major chips (U1, U2 and U3), which means that only minor improvements can be made to component densities, which are high on this board. The simple U4/U5 latch system used to communicate with the 8751 host is effective in operation but suffers from two drawbacks: firstly, it is unidirectional, and secondly it is redundant in component utilization. A possible improvement in the design would be to replace the data latch with a bidirectional buffer, allowing both reading and writing (with modified gating) to be performed. The data latch is not strictly required as the U1 port 0 latches can perform the latch function for the data byte.

5.4.3. UART Board

The 8251 UART, as stated earlier, is a standard Intel Bus part, and as such presented few problems to interface to the 8751. Testing the device consisted of initialization of the 8251 (setting up options in it's internal register) and checking out the serial data flow through the UART. (Initialization is covered in Appendix 9).

Checking the device's serial data handling was achieved using a similar test arrangement to that shown in Figure 5.4., with the UART board replacing the HPIB board in the rack. The HP85 (using the HP85's serial interface) monitored transmitted data and sourced the data for the 8251's receiver tests in a similar manner as for HPIB tests. Tests were made

at several different baud rates to a maximum rate of 9600 band without any difficulties being encountered.

The above tests were made with the current loop and RS232 options in turn (the '85's serial interface supports both options). The active and passive options on the current loop interface were both tested. When testing (either RS232 or C.L.) without handshake lines, it was necessary to hold the CTS line high (pin 5 of the 25 way connector) - clearly the provision of +/- 12v supplies on the 'D' connector would be useful. The action of the various handshake lines was checked by means of the HP85's control/status registers, and the corresponding control/status registers within the 8251.

Compared with the BCD board the UART board has a relatively low density of chips per unit area of board and is therefore a suitable device for mass production. Some of the discrete devices (transistors, diodes, resistors) may be replaced by SIL or DIL parts; suitable transistors arrays are discussed in Chapter 6. DIL or SIL resistors arrays are used on all boards for pull-ups etc., the aim being to produce boards composed of SIL or DIL components aligned parallel to the 32-way DIN connector, permitting the board to be flow soldered.

CHAPTER 6 - THE SERIAL PORT INTERFACES

Earlier in the thesis (Chapter 3) the 8751's on-board UART was briefly discussed. This chapter discusses the device more fully (below), and gives details of the two slave interfaces designed to service the serial port.

The serial port of the 8751 is serviced by on-board full-duplex USART - which will be treated in this thesis as a UART, as the system does not utilize the synchronous facilities at present. The baud rate (See Appendix 1), which is generated using an internal counter, is a function of the crystal frequency; the upper limit is 9.6 Kb for the 3.6864 MHz crystal, and 19.2 Kb for the 7.3728 Mhz crystal.

The serial I/O is routed through a single buffer (SBUF) whose state is monitored by a serial control register (SCON), which also allows the user to configure some of the many options available - such as 8/9 data bits, interrupts etc., etc. These and other registers (see Reference 1) are used to generate functions associated with parity checking, receiver/transmitter disabling and so forth. When operated in 9-bit UART mode, the 8751 has a most useful master-slave mode of operation which is ideal for polling devices in a multi-drop communications environment. Further details of the various modes of operation of the serial port are contained in Appendix 3.

6.1. The Serial Board

Externally (See Plate 6.1.), the board resembles the UART board described in the previous chapter - i.e., it has the same 25-way 'D' type socket on the front panel, and supports the same protocols (RS232 and current loop). However, the interface is, unlike the UART board, a simple buffer-type interface with no inherent intelligence. The lack of an intelligent device with internal command/status registers on

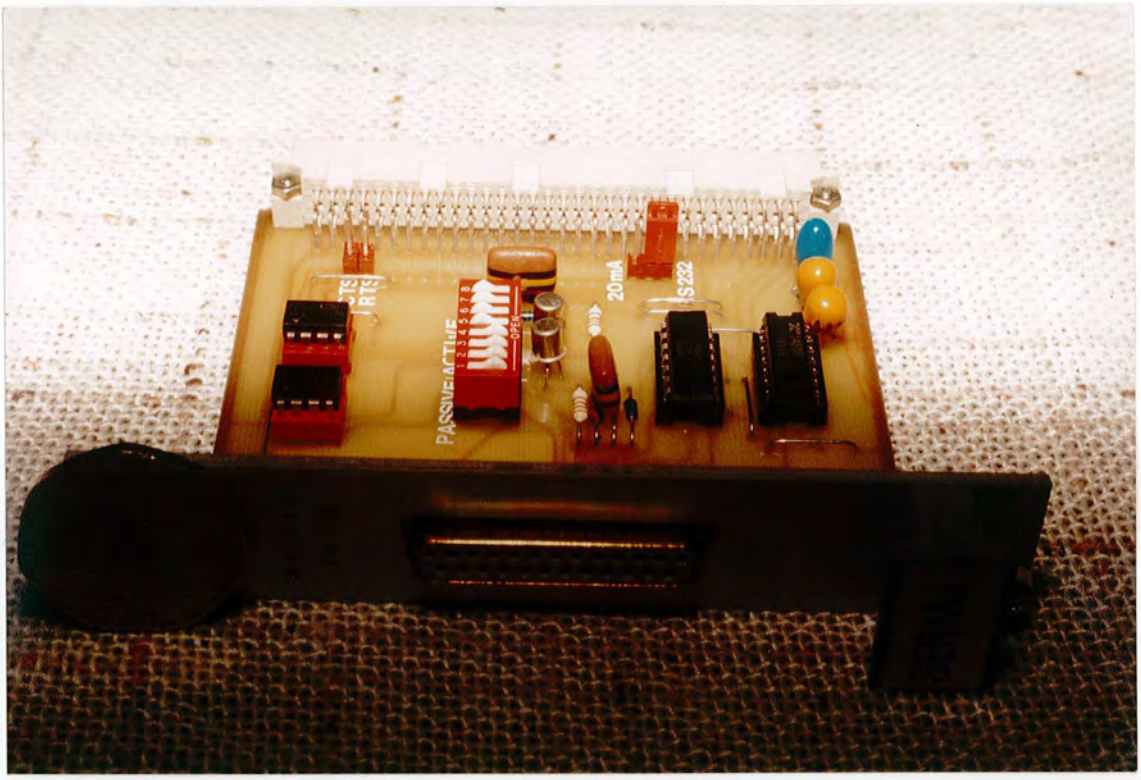


Plate 6.1. The Serial Board

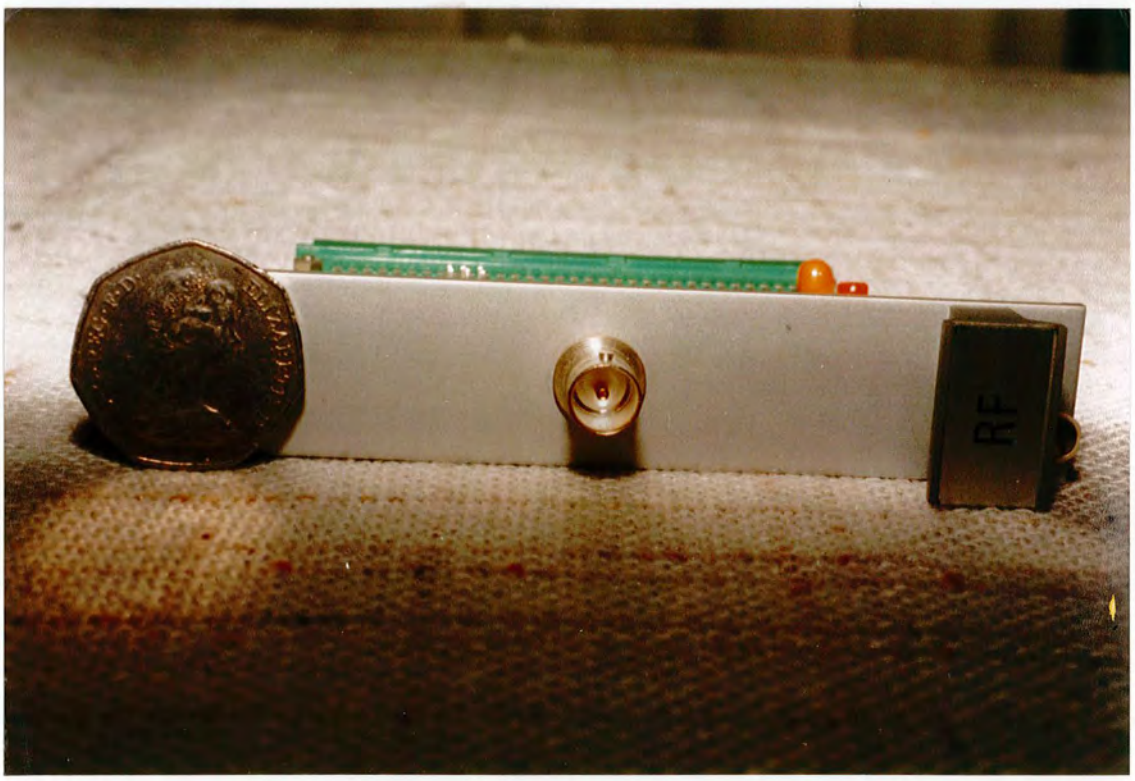


Plate 6.2. The RF Board

the board has certain drawbacks, and the interface supports RS232 handshaking with difficulty - see section 6.1.1.2. below.

Another design problem that the author encountered was the means of baud rate selection - to make the baud rate switch selectable on the board would have meant the introduction of memory mapping hardware, as used on other boards (e.g., the DVI board, described in Chapter 3), resulting in a hybrid board, with data being accessed directly via the serial port, and the baud rate being read from a memory-mapped buffered switch bank - an inelegant solution. Because of the rather limited deployment that the serial board will experience (for most applications the RF Board will be preferred) the author decided that it was sufficient to allow the Baud rate to remain simply as a firmware selectable parameter.

6.1.1. RS232 Facilities

The ubiquitous RS232C protocol (Ref: 12) is the standard protocol for a profusion of applications, originating from data transmission (modem) standards (CCITT V24 - Ref. 33) but now commonly used for computers, printers, terminals and many other devices. It utilizes transmission signals with a wide voltage swing (typically, the range -12v to +12v is implemented), data signals having reverse polarity with respect to handshake signals. The operating range as defined is limited to 50 foot of transmission line, although RS232 signals are often transmitted over significantly longer distances at modest baud rates - low frequency signals are less

affected by the dispersive properties (Ref. 25) of the line.

6.1.1. RS232 Data Signals

The 8751 serial port pins are directly connected (see Chapter 3) to the backplane (pin C26 = Txdata, pin C27 = Rxdata), and support TTL level signals. The serial board buffers these signals using either RS232 buffers or current loop drivers/receivers (see 6.1.2. below); a jumper is provided on the receive line to allow selection of the option required. The circuit diagram of the serial board is shown in Figure 6.1; the transmit data line is buffered to pin 2 of the 'D' socket by U1a (RS232 quad line driver - Fairchild ul489). Data received on pin 3 is buffered to backplane pin C27 by U2a (RS232 quad line receiver - Fairchild ul488). The sensitivity of the line receiver can be adjusted by varying the voltage on pin 3 of the package, but on the prototype board the pin was connected to the supply rail via a pull-up resistor. This proved an adequate system for short-range communication, which is the mode in which the board will operate. A potentiometer acting as a potential divider could be added to the production versions of the board as a sensitivity control later, if required.

6.1.1.1. RS232 Handshake Signals

The possibility of using memory mapping hardware to read baud rate selection via a

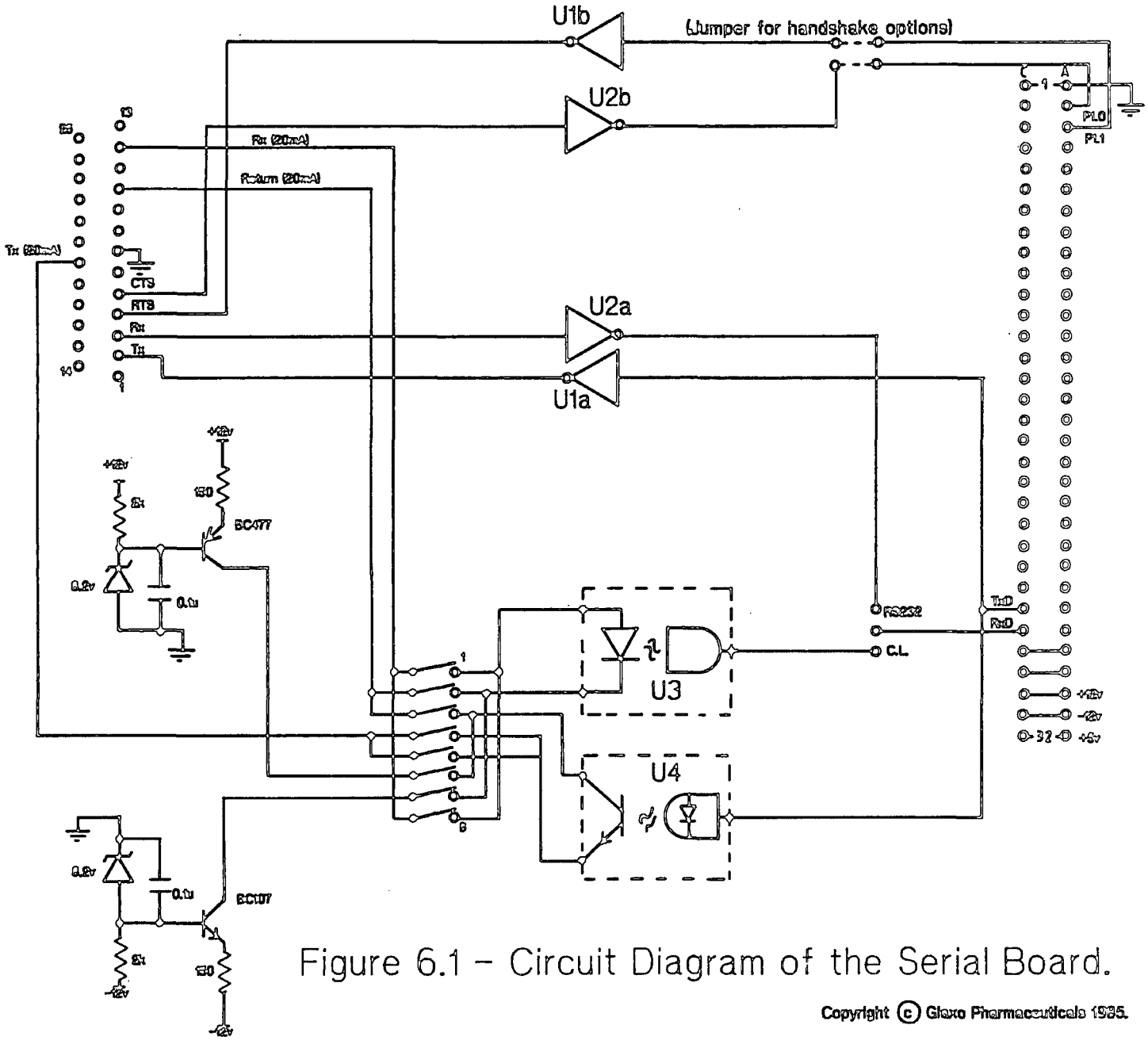


Figure 6.1 – Circuit Diagram of the Serial Board.

buffered switch bank further suggested to the author the possibility of using memory mapping hardware for reading the state (using a '245 buffer as a status buffer) of the RS232 input handshake lines and controlling the RS232 output handshake lines (using a '574 latch as a control register). Again, this approach was rejected, because (as stated before) of the rather clumsy and inelegant hybrid solution that would result, and also because of cost - the memory mapping hardware and status/control circuits, baud rate switches and buffers would result in a design requiring a relatively expensive double-sided PCB artwork. The author was advised by potential users of the system of the strictly limited applications that were foreseen for the serial board, and that a fairly unsophisticated design was required. As a result of these comments, the author decided on an inexpensive solution - to buffer the RTS and CTS handshake lines (as for the data lines above) with RS232 buffers and to jumper the buffered signals to the backplane. U1b (ul488) is the output handshake buffer, which drives RTS on pin 4 of the 'D' connector, while CTS (pin 5) is buffered by U2b. The two bits of port 1 are

'borrowed' when the handshake option is implemented (P1.0 and P1.1). For the reasons given in the previous chapter, (section 5.4 - the UART board) no attempt has been made to provide handshake facilities for the current loop options.

6.1.2. Current Loop Facilities

The need for current loop options on the serial interfaces was explained in the Chapter 5.Current loop protocol is simple to comprehend, being derived from the original telegraph signalling system: a logic 'one' is indicated by the presence of a 20 mA current in a closed loop - hence the name. The absence of loop current (open circuit condition) indicates the logic 'zero' state; these signals states are often referred to as 'marks' or 'spaces' - common terms in serial protocol parlance. The equivalent of the telegraph Morse key in modern current loop interfaces is the bipolar transistor - which is usually conveniently incorporated in an opto-isolator, thus implementing the dual functions of isolation and switching in a single device.

Two chips recently introduced (by Hewlett-Packard) enable engineers to design current loop Rx/Tx hardware with the minimum number of circuit components. These items are the current loop transmitter (HCPL-4100, U4) and receiver (HCPL-4200, U3) which have TTL compatibility as an in-built feature. The TTL data stream entering pin 6 of U4 is converted into an

isolated switched output without the need for further external components. Similarly, current loop data passing through U4's isolation diode is converted into a TTL data stream which is presented the output (pin 7). The receiver chip contains buffers with 0.8mA of input threshold hysteresis incorporated into their input circuits; both chips are much more sophisticated than simple opto-isolators and are specifically designed for 20 mA current loop circuits. The internal circuits of the devices contain buffers which provide a metered current feed to the isolation LED's providing compensation for LED light output degradation, thereby improving reliability.

The Txdata signal from the backplane is fed simultaneously to both the RS232 buffers and the current loop circuits. Received data from either source is connected to the Rxdata backplane signal line by means of a two-way jumper, as shown in Figure 6.1. Options (See Appendix 9) are provided for both passive and active current loops in the receive and transmit circuits. The current loop sources (BC477 circuit) and sinks (BC107 circuit) are conventional base-referenced bipolar circuits; decoupling is provided on the base of each transistor in an attempt to minimize destablization of the reference voltages by environmental noise.

6.2. R.F. Board

The purpose of this board (See Plate 6.2.) is to allow serial communication over long distances (e.g. 500 metres); it

is derived from the R.F. transmit/receive hardware used on the original scheme. The two halves of the circuit are described separately below. The circuit diagram is shown in Figure 6.2.

6.2.1. R.F. Transmitter

This section of the circuit is similar to the original version described in Chapter 2, the major difference being the choice of oscillator. The use of the 567 as a VCO in the original system resulted in a cheap simple design, but the 567 lacked the necessary temperature stability, requiring the oscillator to be trimmed when the temperature inside the die-cast box had stabilized; this was unimportant for the pilot scheme, as the equipment was left running permanently, and therefore required no further adjustment. It should be stressed that drift was expected - in fact it is specified in the data sheet for the '567 (Ref. 2) - and not excessive. However, the use of trim potentiometers is clearly undesirable, so a quartz oscillator module (PXO-1000) was used instead. The oscillator frequency chosen (200 KHz) was sufficiently close to the original frequency as to allow the majority of the component values of the receive circuitry (see below) to remain the same. The oscillator allows dynamic frequency shifting by means of external programming pins (unused pins are held LOW by internal pull-downs); data logic zeros present on pin 2 (see Figure 6.2) drive the oscillator to a frequency outside the pass-band of the PLL's (see below), thus reducing cross-talk risks, as discussed

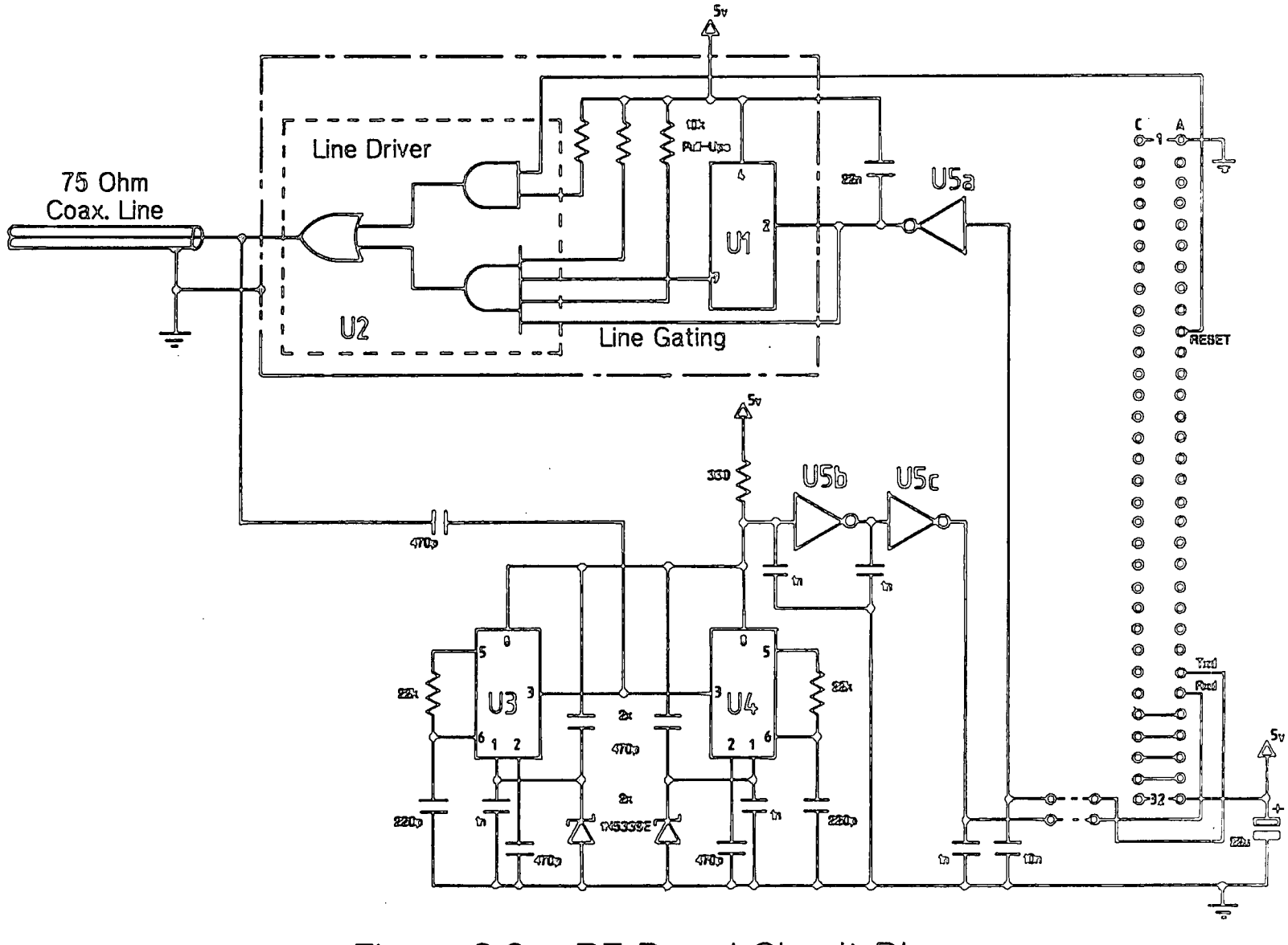


Figure 6.2 - RF Board Circuit Diagram.

in Chapter 2. The line driver circuitry is the same as described in that chapter, except for the additional gating of RESET, to eliminate start-up noise.

6.2.2. Receiver

The receiver uses the successful '567 PLL circuit used in the original system, but with one major difference - two identical PLL's are used in tandem. One of the undesirable characteristics of the 567 circuit was it's tendency to 'spike' to logic ones within the start-bit interval. This somewhat rare yet annoying habit was attributed to charge build-up during the line idle state. The effect could be easily monitored on an oscilloscope by repeatedly transmitting the binary character 01010101 (ASCII 'U'). The author's first attempts to eliminate the effect using analogue filters proved more cumbersome and less effective than the technique finally adopted - two tandem PLL's forming a 'statistical filter'.

The basis of the technique hinges on the open collector output characteristics of the circuit. Tying the two outputs to a common collector load has one important implication - the feedback conditions of the two loops are inter-linked; thus, because the logic LOW state predominates, a loop attempting to break out of lock will be restrained by the other. This is so because no two loops have exactly the same characteristics, because of minor variations during

manufacture. Thus, because the rogue transients are so narrow, and because they occur during different time intervals (in fact, usually only one loop 'spikes' during any particular start-bit interval), the output of the receiver PLL's is a clean noise-free serial data stream. Spare Schmitt trigger functions are used to square up the pulses.

To prevent the loops responding to RF noise from the transmitter's line driver circuitry, the oscillator and the driver were enclosed in a screening can. (The (earthed) can is shown as a chained line in Figure 6.2.).

6.3. Review

Both the serial and RF boards were tested with the PDS Emulator/Rack/HP85 arrangement used in previous tests (see below). Both boards operate using data transmitted from the serial port, and so the same 8751 firmware (see Appendix 3) was utilized to test both boards.

6.3.1. Serial Board

The test arrangement used for checking-out the UART board (see chapter 5) was utilized for testing the serial board. The HP85 serial interface (Type 82939A Option 001) was initially configured for RS232 and used to test RS232 data transfer to and from the rack system, without handshaking. The handshake options were then tested in turn using the CTS option (port pin P1.0) to control data transmission from the 8751

and the RTS option (port pin P1.1) to control data reception from the HP85.

For the tests described above, the emulator's high-level macro language was used; for a true simulation of real-time data a small assembler routine (listed in Appendix 3) was used to soak-test the board at 9600 baud. For this test the HP85 was disconnected from the 25-way 'D' connector and replaced by a link joining pins 2 and 3. The HP85's serial interface was then modified to receive TTL signals (this is a simple modification to the circuit board of the device, involving connecting a spare cable lead directly into the board's UART input, thus by-passing the RS232 buffer) and used to monitor the output of U2a (Figure 6.1.). For this purpose the RS232/CL jumper was removed, and the U2a jumper post used as a monitor point. Using this system the board was soak-tested to check-out the circuit buffers and the serial port stability. A simple program within the HP85 was used to confirm data consistency - the character 'U' being used as the test character for transmission.

After these tests were completed the current loop circuits were tested. For this purpose the HP85's serial interface was configured for active current loop(s) (transmit/receive - the option 001 interface is a versatile interface supporting R2332 and active/passive current loop) and the Serial Board (Rack) was configured for passive current loop(s). The current loop (jumper) option was selected and the

HP85's serial interface was plugged back into the front panel socket. Data was exchanged in both directions between the 8751 and the HP85 and the tests repeated with the active and passive roles reversed.

Several further tests were carried out with different current loop devices (e.g., industrial balances) substituting for the HP85. For these tests the emulator's high-level command language was used to control data exchange. The instruments with current loop interfaces in common use within the Company in general have generous (e.g. 15-25 mA) margins on the line marking current specification, and current measurements showed the line currents to be well within specification for all cases.

In general, the serial board proved to be a simple trouble-free interface, and should prove useful for many applications. Tests using the 8052-BASIC CPU (Chapter 8) showed that as well as acting as the console interface, the board could be simply modified to enable a printer to be interfaced to the system. This suggests that an extra jumper option would be worth adding; it is quite feasible to drive serial data in and out of spare part 1 pins using software timing loops - indeed, this is a technique commonly used to facilitate serial data transfer on microprocessors not equipped with a serial port (Ref. 32 offers suitable algorithms for the MCS-48 family). The switch-bank used for passive/active current loop selections (See Appendix 9) was an item selected from

the Contractor's stock (after discussions with the Contractor) and was used because it is a low-cost well-stocked item. However, it allows the user to make switch selections that are undesirable, and the options would be better selected by a fool-proof jumper nest, similar to the system used to select DVI board options (Chapter 3).

The current loop sources and sinks (the UART board uses the same circuitry) being based on discrete devices, would be better implemented with DIL packages for ease of mass production. RCA, for example, manufacture several useful transistor arrays, e.g. the CA3096 and the CA3097 - the latter part being of particular interest as it incorporates suitable transistors and also a zener diode.

6.3.2. RF Board

Figure 6.3. shows schematically the system used for testing RF Boards. Two boards were used within the same rack. Each board was fitted with a BNC 'T' connector - each of which was fitted with a 75 Ohm line terminator. The two connectors were joined together by means of a short length of 75 Ohm coaxial cable, each end of which was terminated with a BNC socket. One of the boards (1) was connected to the backplane serial port lines, while the other (2) was isolated from the lines by removing the jumpers (See Figure 6.2.) from the board .

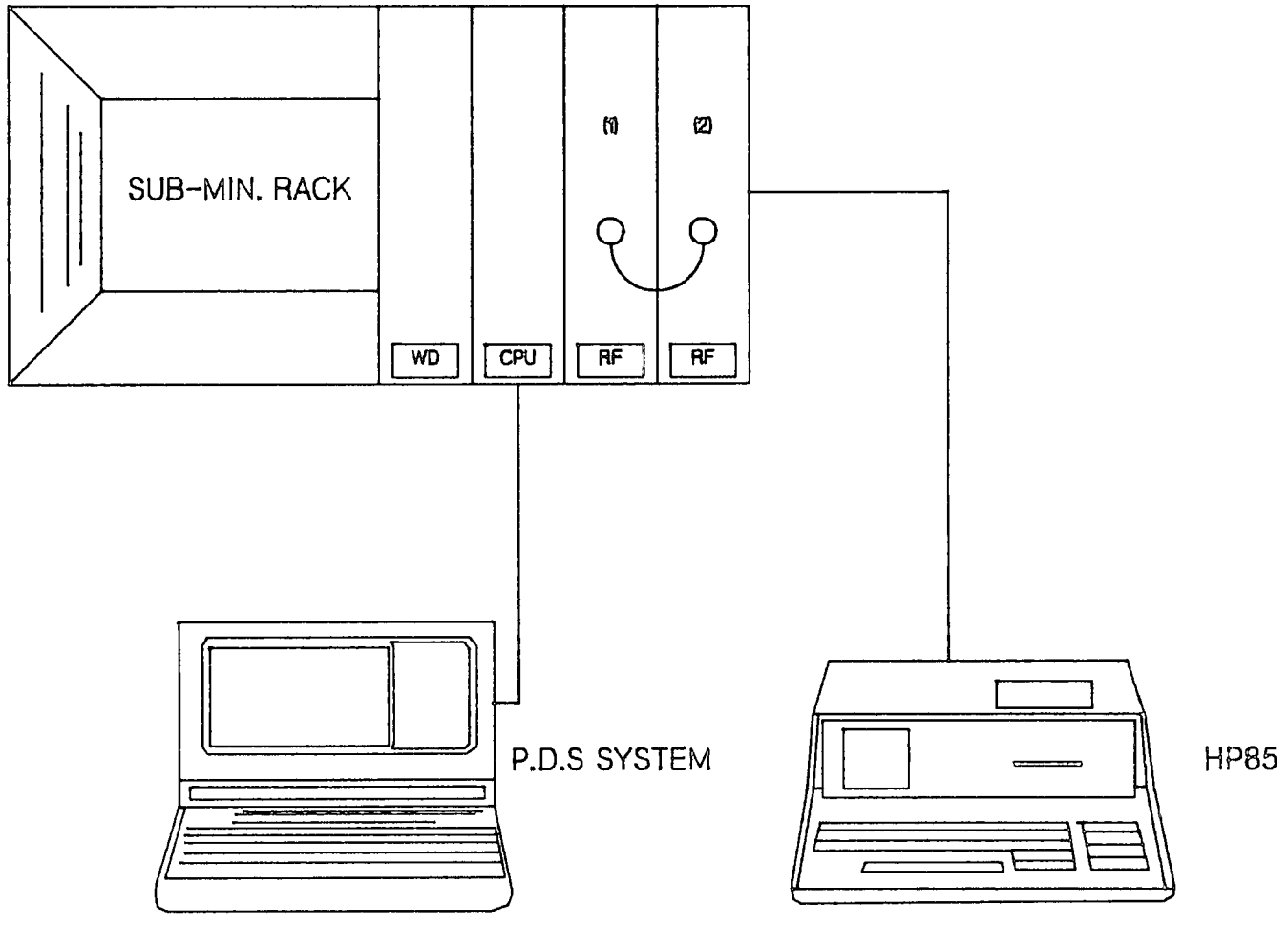


Figure 6.3 - Schematic Diagram showing RF Board Testing.

The HP85's serial interface was modified for receiving TTL data signals (see previous sections) and used to monitor signals from the board's (2) PLL output. The receiver line jumper pin and ground provided the signal connection points. Using this system and the program listed in Appendix 3, the HP85 was used to monitor the serial data and confirm validity, in the same manner as for the soak-tests of serial boards. During soak tests the RF boards worked well, without data corruption or spurious outputs. Long-line (300 metres) tests were performed which confirmed the operation of the tone-burst decoding over extended distances.

The replacement of the original VCO (Chapter 2) circuit with a crystal oscillator module corrected problems associated with temperature dependent frequency variations (See Appendix 2). Further testing of the RF board, however, revealed that the 8751's UART was more critical to the PLL output pulse width than the UART on the HP85's serial interface. This caused problems with incorrect data reception on some batches of prototype boards. Appendix 2 shows the effect of pulse narrowing, observed on an oscilloscope monitoring the PLL's output. Clearly this reduction in duty cycle is a function of loop lock speed, which suggests that a faster loop (the 567 is normally used for tone decoding applications at relatively low frequencies) might be used to improve performance. The NE565 (the original choice - see

Chapter 2 - rejected because of the lack of split 12v supplies for the pilot scheme hardware) is a suitable candidate for substitution, but several other loops recently introduced have excellent specifications.

CHAPTER 7 - THE ANALOGUE BOARD

Frequently there is a need to monitor and record analogue measurements associated with production equipment on the Company's manufacturing lines - e.g., conveyer speeds, temperatures, pressures etc.. Such measurements are typically obtained via transducers which output signals in the range 0-10v, or 4-20 mA. (The latter system is often considered 'superior' as it has an 'active zero', which in theory allows an open-circuit (0mA) condition to be recognized; in reality, the equivalent performance can be obtained using, say, 0-0.2v as the guard band, and 0.2-10v as the signal range).

The following design goals were set for the Analogue (Input) Board, which would fulfil the Company's needs for current projects:

* 8-bit ADC
* On-board reference voltages
* 8 input channels
* Single-ended inputs
* On-board memory

These requirements were met using a design based on the AD7581 - a unique 8-channel ADC (Analogue to Digital Converter - Ref. 26) from Analog Devices. The design is based on the application notes that accompany the device's specification, and the application information supplied in 'Electronic Design' (Ref. 27).

7.1. The AD7581

The ADC employed in the design is a special hybrid device - it is both an ADC and a 8x8 bit memory in a single package. These can be thought of as two independent devices, with separate functions; the ADC section performs the tasks associated with a multi-channel ADC, while the memory and the digital logic circuits provide a clean, convenient interface to

the (8751) host. Each mechanism is covered briefly in the two sub-sections below; the reader is referred to the references listed above for further details.

7.1.1. The 8-channel ADC

The eight analogue inputs are scanned by an analogue multiplexer at a rate determined by the frequency of an external clock. An output is available to indicate the channel currently being scanned, but this generally is irrelevant to the host - see 7.1.2. below. The output of the multiplexer is fed to a conventional successive approximation type ADC; the speed of conversion is proportional to the external clock rate. At the end of the conversion the result is passed from the SAR (successive approximation register) to the memory stack.

The ADC features single ended inputs and a common analogue ground, which may be tied to the digital ground if desired (depending on the application). The manufacturer's specification states that the AGND-DGND voltage must not exceed the supply voltage, which indicates that for some applications a proprietary external signal conditioning unit will be needed.

The reference for the ADC (VREF) must be supplied off-chip, and the same observation applies to the offset (BOFS) for the internal comparator; the external circuitry (see 7.2. below) required is minimal, and allows the designer to implement a wide range of custom designs.

7.1.2. The Digital Logic

The '7581 contains it's own internal address latches, and needs no further external circuitry for interfacing to 8085-type Buses. The signal connections that comprise the Bus interface are ALE (See Chapter 3), three address lines, and the eight Bus lines. The address lines (A0-A2) are used to address the eight memory locations within the '7581 that contain the conversion results for the corresponding analogue channels. The clock input is, in general, not relevant to the timing of the digital interface, with the exception of the channel identification output (STAT - not required).

The designer is not usually interested in identifying the channel currently being scanned by the analogue multiplexer within the 7581; normally an ADC would interrupt the host CPU to prompt for attention after completing a conversion - but the 7581 removes the burden of ADC interrupt servicing from the host by storing the conversion result in its internal RAM array. The user extracts the result at any convenient time; to the host, the ADC simply appears to be a 8x8 bit external RAM (c.f. Chapter 4). The 7581 deals with any contention problems associated with the RAM, and the fast access time specified for this section of the device (200 ns) means that there are no timing problems. The data is latched (when valid) on the Bus for the duration of the chip select pulse.

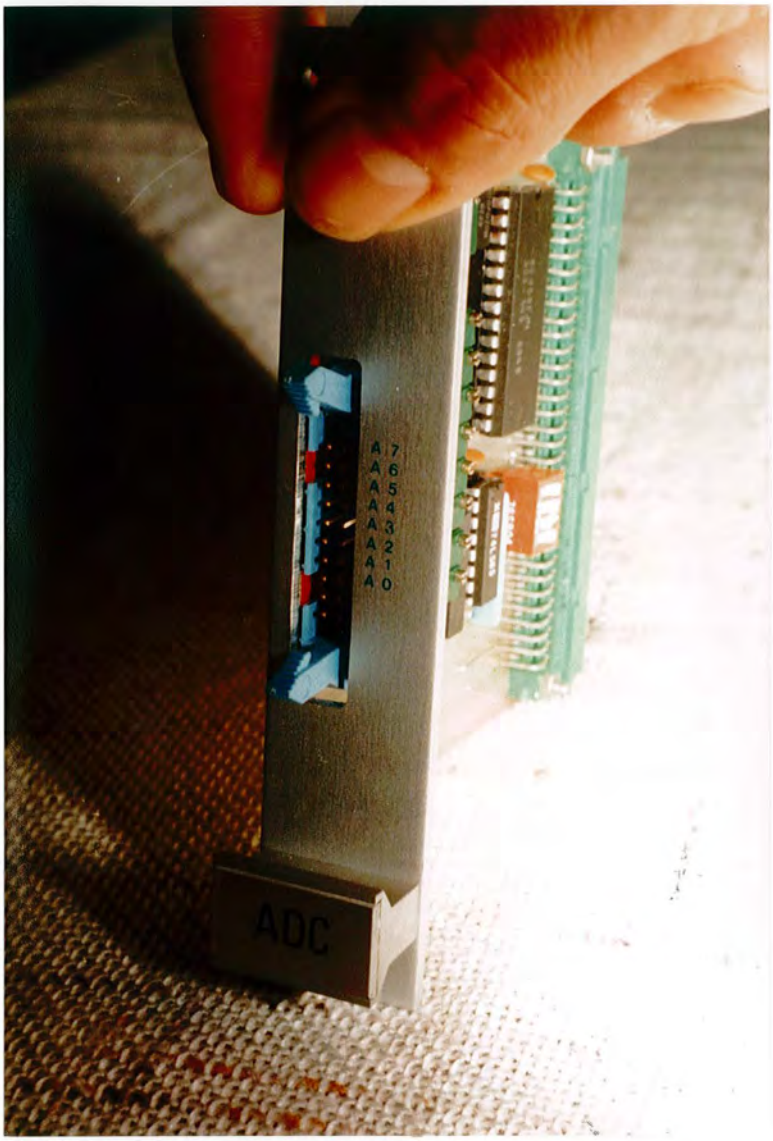


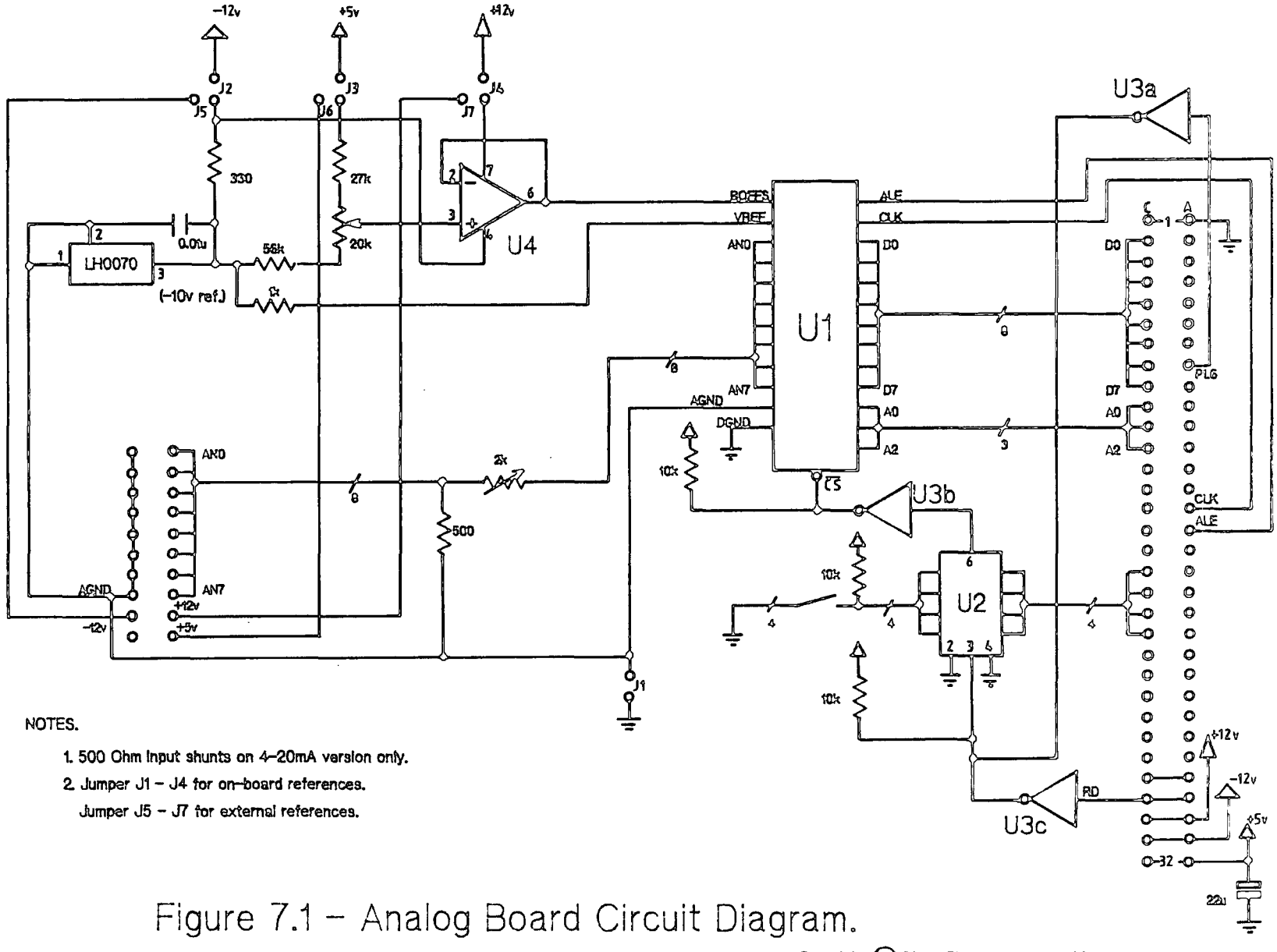
PLATE 7.1. - THE ANALOGUE BOARD

7.2. The ADC interface design

Figure 7.1. shows the design of the ADC board; it can be seen to be a relatively straightforward implementation of the A.D. application notes (Refs. 26 and 27). The ADC's Bus and address lines connect direct to the backplane, as does ALE, without the necessity for further external components. Chip select (active LOW) is achieved using the familiar (see previous chapters) '85 comparator logic, the comparator itself being enabled by the presence of both PERI (Peripheral Select - active LOW) and RD (Read - active LOW), both signals being buffered by '05 (U3a, U3c) inverters with AND-tied outputs. A further buffer (U3b) is used to invert the logic high signal from the comparator (U2) to enable the 7581 (U1) via the chip select pin (CS - active LOW). The backplane clock (Clock 1) signal (see Chapter 3) is used as the source for the clock input (CLK - pin 15) to the ADC.

Provision is made for an on-board reference or a reference taken from external supplies; the reference voltage determines the range of the internal comparator. The voltage is taken from a -10v stabilized LH0070 reference chip, which is fed from a -12v supply referred to analogue ground. The offset (BOFS) - pin 1) voltage is supplied from a potential divider network (between 5v and -10v) buffered by an ADOP07 operational amplifier configured as a voltage follower (U4). The choice of using on-board supplies (the penalty is the loss of isolation) or external supplies is made via jumpers J1 to J7.

The use of the (27k) offset-trim potentiometer, together with (2k) trim potentiometers in series with the analogue



NOTES.

1. 500 Ohm input shunts on 4–20mA version only.
2. Jumper J1 – J4 for on-board references.
   Jumper J5 – J7 for external references.

Figure 7.1 – Analog Board Circuit Diagram.

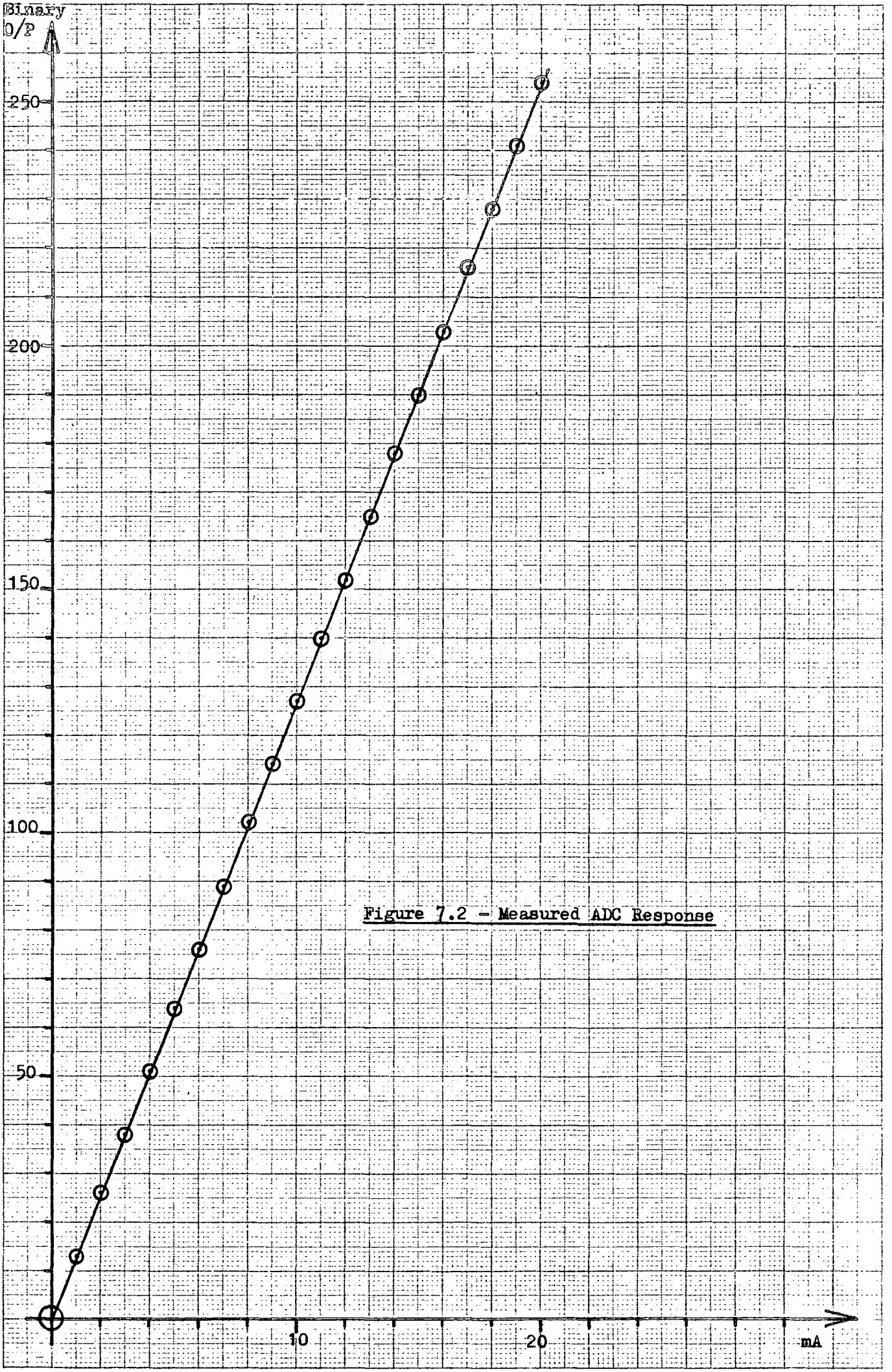


Figure 7.2 - Measured ADC Response

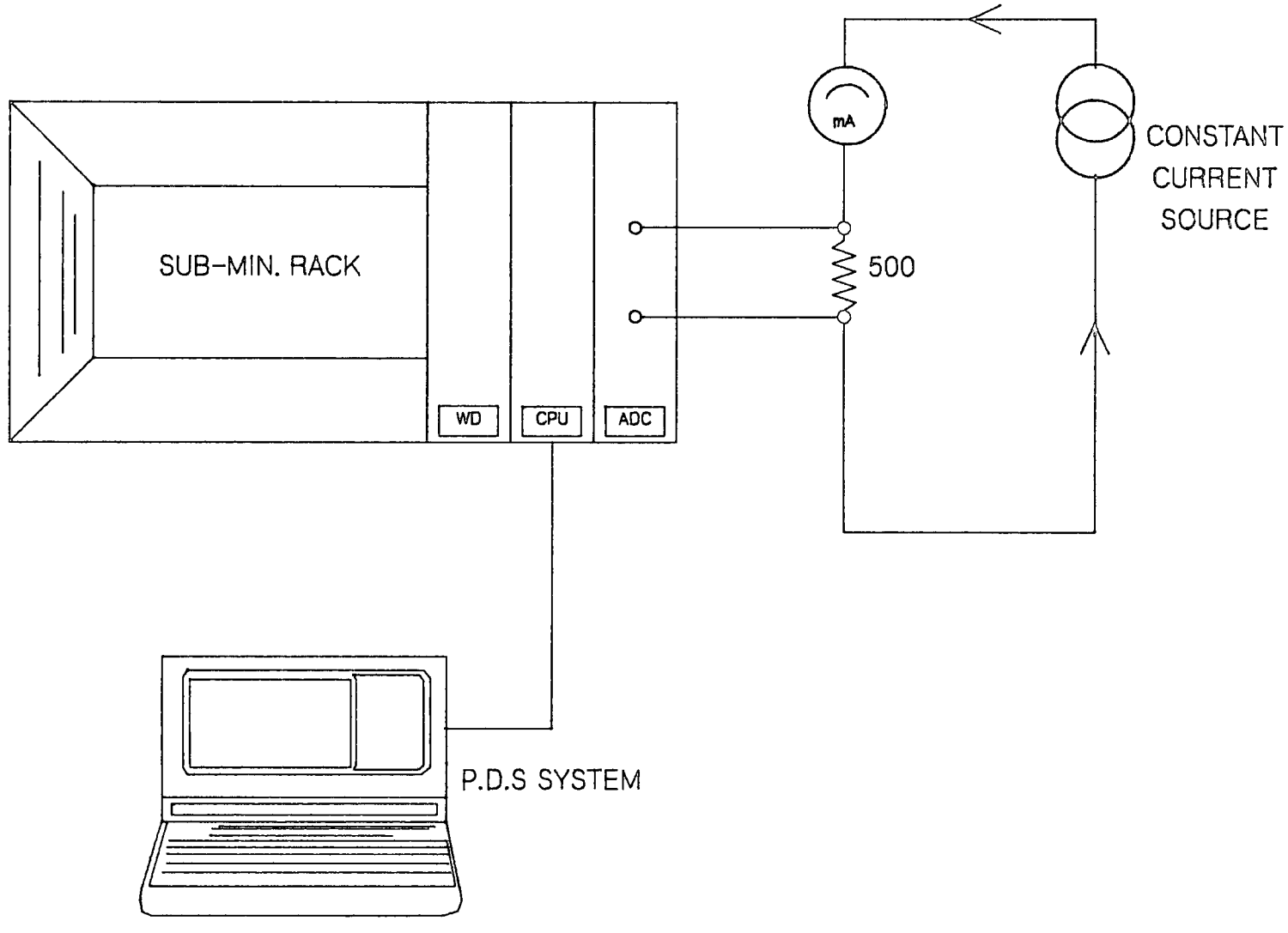


Figure 7.3 – Test Rig for ADC evaluation.

inputs allows the user to trim for optimum span and zero set conditions. The circuit is designed primarily to cater for 0-10v input signals, but optional 500 ohm shunts allow for 4-20 mA signals to be accepted. There is no isolation comparable to that provided on the DVI board (Chapter 3), due to lack of space; opto-isolators would be inappropriate, due to the non-linearity inherent in such devices. The author feels that such isolation would be better provided by an external isolation panel, many excellent units being available on the market at present.

7.3. Review

The AD 7581LN chip was used for the circuit check-out; this is the most accurate part available, with a quoted accuracy of $\pm$ 1/2 LSB. The interface's offset null and gain adjustments were made in accordance with the manufacturers data sheet directions to give an input range of 0-10v. In order to check out the current response (4-20mA transducers are preferred to 0-10v by the Company's Instrumentation Engineers) the test configuration shown in Figure 7.3. was used. A pre-settable constant current source (Time Electronics Model 505) was used to supply current to an external 500 Ohm (0.1% wire-wound) shunt resistor; a series ammeter (Beckman T100B Industrial Multimeter - 0-20mA range) was used to monitor the input current (Channel 7 was used for the tests).

Table 7.1. below summarizes the results obtained. The ADC output was monitored via the PDS display, using EMV51's REPEAT and 'PBYTE' commands to set up a monitoring loop, displaying the readings on the screen of the PDS.

Input Current (mA)	ADC O/P	Theoretical O/P
0.00 ± 0.005	0	0
1.00 "	13	12.7
2.00 "	26(5)	25.4
3.00 "	38	38.1
4.00 "	51(2)	50.8
5.00 "	64	63.5
6.00 "	76	76.2
7.00 "	89	88.9
8.00 "	102	101.6
9.00 "	114	114.1
10.00 "	127	127.0
11.00 "	140	139.7
12.00 "	152	152.4
13.00 "	165	165.1
14.00 "	178	177.8
15.00 "	190	190.5
16.00 "	203	203.2
17.00 "	216	215.9
18.00 "	228(7)	228.6
19.00 "	241	241.3
20.00 "	254(5)	254.0

TABLE 2.1. ADC TEST

It can be seen from the above results that, within the limitations of instrumentation accuracy and 'flicker' on ADC LSB, that the ADC chip performed within the manufacturer's accuracy specification. The chip is impressively linear in its operation as can be seen from the graph of these results (Figure 7.2.); tests on the remaining channels produced comparable results.

Physically, the board design was good, a limited number of components being used to realize the interface. This resulted in an uncluttered PCB layout on a board employing single-sided tracking. The interface will be suitable for future mass-production, with a low assembly cost.

CHAPTER 8 - RUNNING MCS BASIC-52 ON THE SYSTEM

Developing programs whose source code is written in assembly language is a time-consuming task. An alternative is to write programs in a high-level language such as Pascal, FORTRAN or BASIC. As an option, Intel provides the user with PLM-51, a PL1-like language which may be compiled using a disc-based compiler residing on a development system. The user still has to contend with the tedium of the familiar edit/compile/link/run cycle inherent in debugging programs.

The ubiquitous BASIC language is familiar to many people outside the programming fraternity and is a boon to many engineers, scientists and, indeed, laymen; this popularity is in part due to the simple structure of the language, although it is for the most part a result of the fact that the language is usually implemented as an interpreter rather than a compiler - allowing the user to enter the program and type 'RUN' to begin immediate execution. Intel have catered for users who wish to run BASIC on MCS-51 family systems by offering a novel variant of the 8052AH processor - the 8052AH - BASIC chip.

8.1. The 8052AH-BASIC Microprocessor

The 8052 single-chip microprocessor is a recent addition to the MCS-51 family (see Ref. 1). It is a ROM-based part with 8k of ROM available to the user. There are other significant enhancements to the 8031/8051/8751 specification, but these need not concern us in this chapter. Intel have themselves (Ref. 20) developed a stand-alone BASIC system consisting of a BASIC interpreter, library routines and EPROM blowing logic all of which is contained in the 8k ROM space of 8052AH-BASIC. The BASIC supported by the system is known as 'MCS BASIC-52'.

Plate 8.1. shows the 8052AH-BASIC chip side-by-side with an 8751 chip, for comparison. The number of pins, layout of ports, etc. is the same for each processor. The BASIC chip uses the 8052's serial port as a dedicated TTL-level console interface; it also uses the eight bits of PORT 1 (see Section 8.2) for special functions. Because it is a member of the MCS-51 family, the BASIC chip may be substituted directly for the 8751 on the CPU board of the system described in this thesis.

MCS BASIC-52 has much to offer the user - Table 2.1. below lists some of the features.

* Full BASIC interpreter in ROM
* Floating-point maths
* EPROM file commands
* Interrupts handlable by BASIC/Assembler
* Linkage with assembler programs
* Programs may reside in RAM or EPROM

TABLE 2.1. - FEATURES OF MCS BASIC-52

The chip supports a remarkably comprehensive BASIC language - far superior to the 'TINY BASICs' offered on other ROM-based systems. In order to function, the chip requires, as a minimum, an address latch, 1k or more of external RAM memory, timing components, and a suitable serial console device. After RESET or power-up, the system expects the operator to press the space bar on the console, whereupon it automatically determines the baud rate of the console device and signs on with the message

```
* MCS-51 (tm) BASIC Vx.x*
READY
>
```

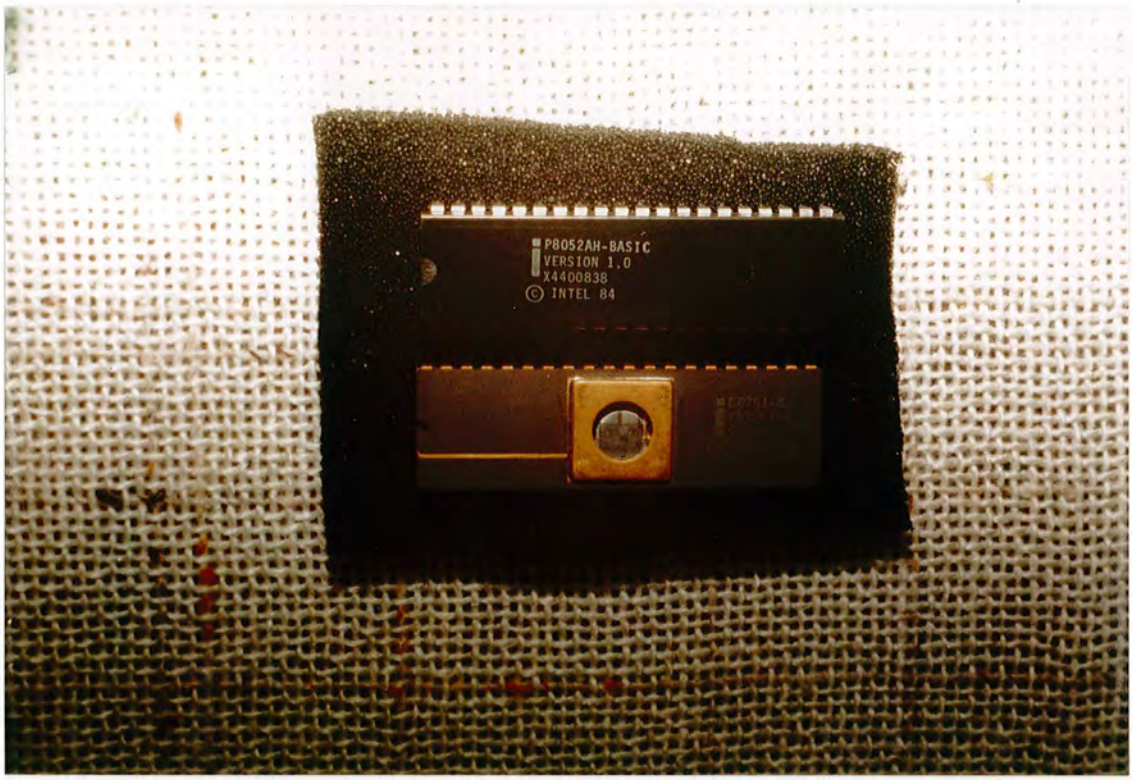


PLATE 8.1 - ALTERNATIVE CPU CHIPS

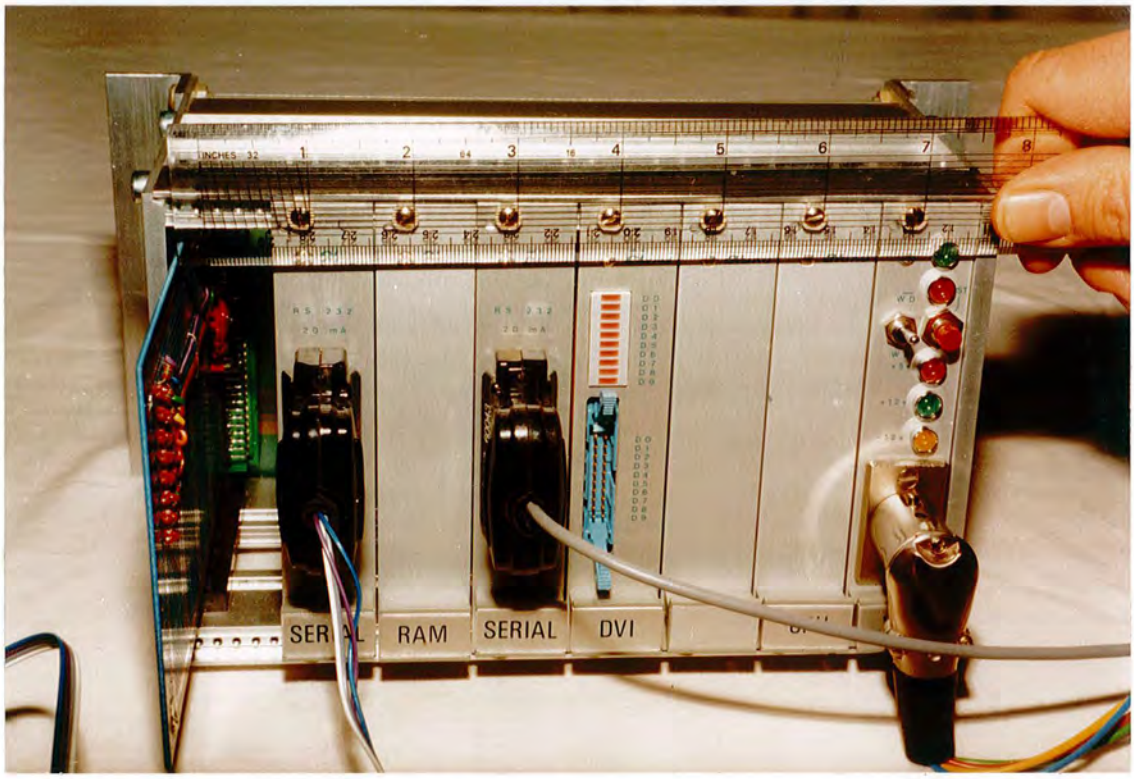


PLATE 8.2 - THE 'BASIC' SYSTEM CONFIGURATION

The final character in the sign-on sequence is a prompt to the operator to begin entering BASIC commands. More details concerning the system facilities, memory configurations and other features can be found in Appendix 8.

8.2. Initial Check-Out

The author was able to check-out the BASIC system using a miniature sub-rack and a number of the boards described elsewhere in this thesis. The required boards were:-

(i) CPU board with 8052-BASIC CPU installed.

(ii) RAM board with 8K IRAM installed in lower address space.

(iii) Watchdog/power board with watchdog disabled.

(iv) DVI board with select code set to 8.

(v) Serial board for console device interface.

(vi) Serial board (modified - see below) for use as printer interface.

(vii) Port 1 monitor board (see below).

Plate 8.2. shows the cards installed in the sub-rack, ready for testing. Some minor modifications to the boards were needed before the system could be run; Port 1 is used by the BASIC chip for special purposes, purposes which clash with the use of some of the Port 1 pins on various boards in the system. In the case of the board configuration outlined above, the only port pin affected is P1.6, which is used on RAM/peripheral boards to distinguish between memory and peripheral accesses via the data bus. To test out the system it was necessary to isolate the control signal on the RAM and DVI boards, and tie the signal to the appropriate enable level (ground or 5v), in either case, using discrete resistors. In the case of the

watchdog card, it was sufficient to remove the relevant chips on the card, as it was needed only as a power supply link. The DVI select code was set to '8' - i.e., Bus address 8000H, in order to avoid conflict with the RAM chip's address range.

MCS BASIC-52 uses port pin P1.7 to support an optional list device - i.e. it treats the pin as a serial output channel. The user selects the baud rate with the BAUD command in BASIC. A standard serial card was modified by disconnecting the TTL serial lines from the backplane connector pins on the card and connecting the P1.7 pin on the connector to the RS232 output buffer on the card. This converted the card to a makeshift printer interface to a standard EPSON FX-100 printer.

The author used as a convenient console device a small 'TRANSTERM' portable terminal; this was configured to operate at 110 baud, to enable the listing of programs to be observed. The TRANSTERM was connected to the system via a standard serial interface board. After power-up the space bar on the console was pressed, whereupon the sign-on message (described previously in this chapter) immediately appeared on the terminal's small LCD monitor.

After receipt of the 'READY' prompt, the author was able to enter BASIC statements into the RAM memory via the console. The system uses the first 1k of RAM for program stacks, the rest of memory being available to the user. As well as at the 3.6864 MHz clock rate, the system was tested using the 7.3728 and 11.0592 MHz (using faster memory - see Appendix 11) crystals, and found to function correctly. Plate 8.3 shows BASIC running on the system.

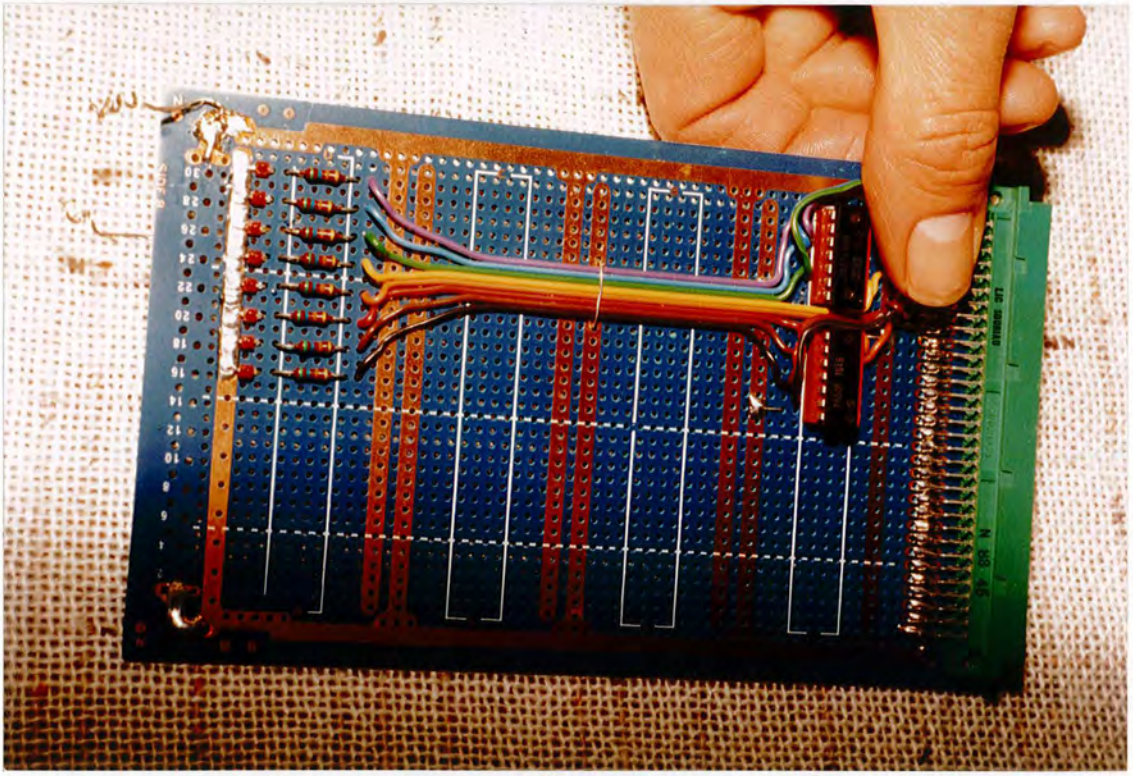
PLATE 8.3. MCS BASIC-52

RUNNING ON THE SYSTEM

PLATE 8.4. - (BELOW)

THE PORT 1 MONITOR

BREADBOARD



8.2. Port 1 Functions

The BASIC system has a 'command' mode of operation which was used initially to confirm the system operation. The author used the 'PORT 1' commands to drive the port 1 lines on the backplane. A simple PORT 1 monitor card, shown in Plate 8.4., was used to display the state of the logic levels on the lines on LED's. Figure 8.1. shows the circuit of the breadboard.

Executing the commands

```
PORT 1 = 0

PORT 1 = 00FFH
```

via the console confirmed that all eight Port 1 lines were being driven.

To check-out program sequencing, the following program was entered and run:

```
10  REM DRIVE EACH PORT 1 LINE
20  P= 1
30  FOR I=1 TO 8
40  FOR J = 1 TO 100
50  REM DRIVE PORT PIN
60  PORT 1=P
70  NEXT J
80  REM SELECT NEXT PIN
90  P = 2*P
100 NEXT I
110 STOP
```

The program's function is to drive each port 1 pin in turn, holding the pin in an active state for the duration of the FOR loop occurring between statements 40 - 70. In practice, when the program was run the LED's were seen to energize in rapid sequence, confirming CPU operation and program execution.

The Port 1 functions (see Appendix 8) relevant to the current configuration tested were the PWM command and the LIST=

command. The function of the PWM command is described in section 8.3; the LIST= command directs listing to a printer rather than the console device - port pin P1.7 is used by MCS BASIC-52 as a serial channel for the listings.

The command BAUD 1200 was issued via the printer to configure the printer port to match a 1200 baud Epson FX-100 serial printer attached via a cable to the serial port interface. A small BASIC program was entered via the console and the command

LIST#

was issued, which confirmed the operation of the printer channel. The printer port facility was found to be most useful for taking quick listings of the programs under test, as the two line console used by the author was not convenient for listing programs. Later the author borrowed a standard HP2392A terminal from the Company's mainframe installation, which proved to be a very acceptable device for program debugging.

8.3. Sensor Interfacing

Having tested out the program sequencing, the author proceeded to investigate the accessing of sensor data via the DVI card. The system admits two possible ways of achieving access to the card:

i) By an assembler routine called from BASIC.

ii) By a special BASIC-52 'XBY' command, which allows the user to access external memory locations directly from a BASIC program.

The first option requires the development of a small assembler subroutine, which would then be assembled into object code and stored in an external EPROM. The rather exacting

requirements of the external memory layout (see Appendix 8) precluded further investigation by the author. Possibly this work will be pursued at a later date (see Section 8.4). The use of the 'XBY' command was investigated by means of the BASIC program listed below:

```
 10 REM PROGRAM READS DVI CARD
 20 REM GIVING AN AUDIO FEEDBACK
 30 A = 8000H
 40 X = XBY(A)
 50 PRINT X
 60 H = X + 20 : L = H
 70 N = 100
 80 PWM H, L, N
 90 REM OUTPUT TO DGO CARD
100 XBY(A) = X
110 GOTO 40
```

Before running the program a prototype 'DGO' card (a digital output card was added to the list a cards required by the potential system users, towards the end of the work covered by this thesis. It performs the opposite function to the DVI card, i.e. it provides eight isolated output lines, the state of which are monitored by front panel LED's) was inserted in the spare slot in the rack (see Plate 8.1). The card address was set to 8, but no conflict occurs as the DVI card is an input only device, while the DGO card provides the complementary function .

The program transfers data from the DVI card by means of the XBY command in statement 40. The parameter 'A' is the external memory address of the DVI (and DGO) card. The decimal value of the eight-bit data byte ('X') is logged on the console. The PWM (pulse width modulate) statement at location 80 causes the output of a square wave pulses on special function pin P1.2. The program selects the pulse high time (H) and pulse low time (L) at statement 60; the number of cycles (N) to be output is

selected by statement 70. An amplifier (a simple audio signal tracer) was coupled to the output of the P1.2 buffer on the monitor card as shown in Figure 8.1, to monitor the audio output. The minimum specified values of the high and low parameters are both 20; hence it is necessary to add 20 to the value X read from the DVI input lines. The program copies the input data out to the DGO card at statement 100, before looping back to statement 40 to fetch fresh data.

A simple DIL switch-bank arrangement was used to connect the positive (DVI) inputs to an external +5v supply, the negative inputs all being commoned to the supply's ground. 330 ohm series resistors were used for current limiting. The program performed as expected with this arrangement, reading the values from the DIL switches, logging the value on the console, producing an audio output dependent on the value read and copying the input value to the DGO output latch (the value was monitored on the front panel display).

8.4. Enhancements

To verify the system further the author experimented with simple modifications to the above program to check-out the DO-WHILE and DO-UNTIL statements as alternatives to the simple GOTO loop. The author found the BASIC-52 version of BASIC in general both powerful and flexible. The EPROM file handling and related extensions to standard BASIC were not investigated as the author did not have the appropriate hardware configuration. The necessary modifications to the CPU board design and other aspects of the system were considered outside the scope of the author's design brief.

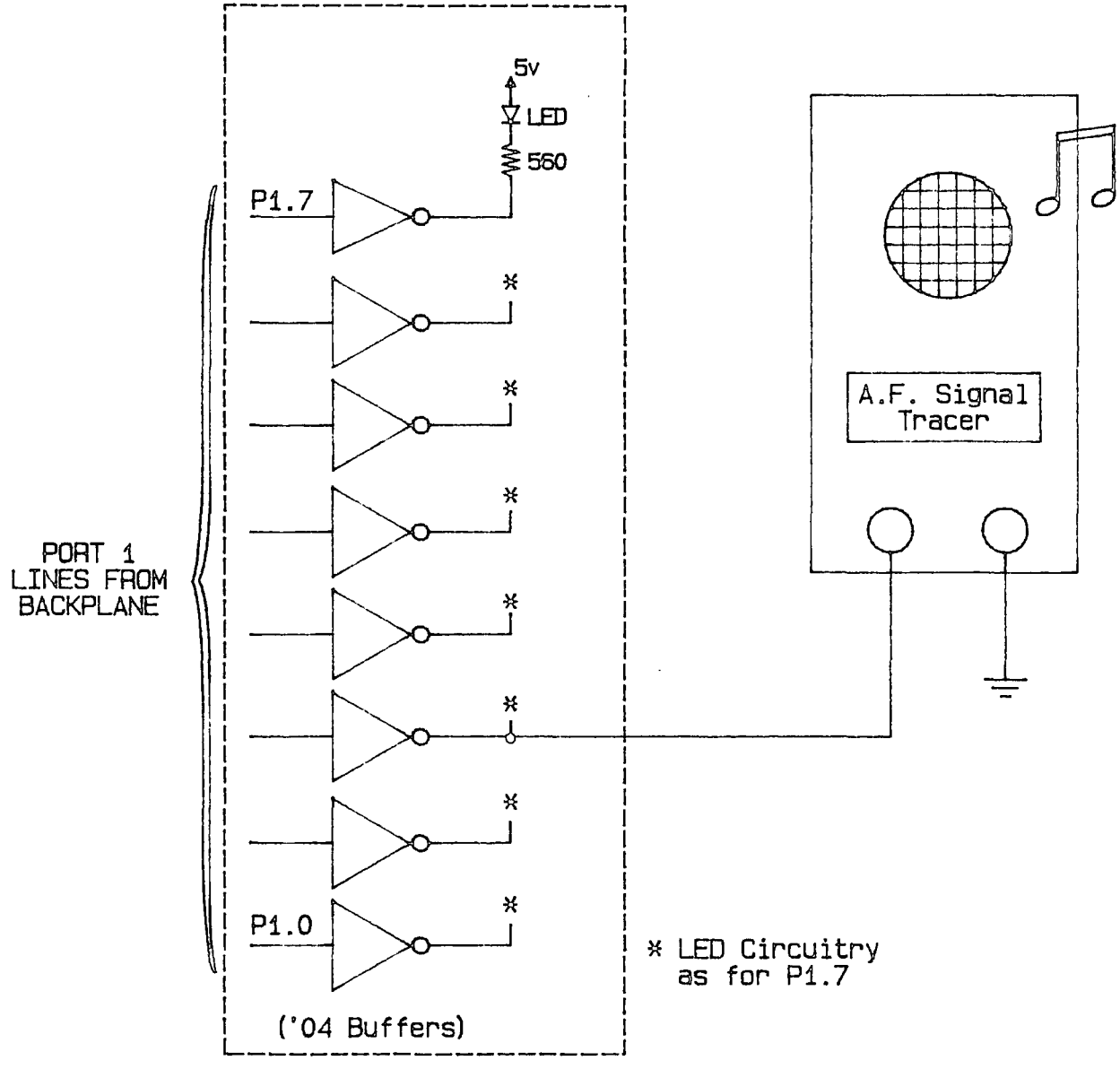


Figure 8.1 - Port1 Monitor Breadboard & Audio Monitor.

8.5. Review

The 8052AH-BASIC microprocessor appeared on the commercial market towards the end of the author's work on the system, when most of the design decisions had been taken. Had it entered the market a year or so earlier, it would have been feasible to have designed the system around the device's specialized external memory requirements (Appendix 8). However, despite the limited amount of testing the author was able to perform, the BASIC-52 system impressed in its ease of use and range of facilities.

Clearly it would be useful to have such a system as an alternative to an assembler-based system, as it requires no expensive support hardware and is easily reconfigured in the field by service engineers. It was, for example, a simple matter to add a series of 'IF' statements to the BASIC program described in Section 8.3. to produce a specific message on the printer log for each sensor input change detected - creating, in essence, an instant alarm logging system. To create an equivalent system using the standard cards in the system and assembler-based 8751 firmware requires considerably more equipment and effort.

Thus it may be concluded that the BASIC system has clear advantages for ease of use and rapid development. The penalty paid for these facilities is, of course, speed - an interpreted system runs much more slowly than the system based on compiled or assembled programs. So it can be seen that each processor, the 8751 and the 8052AH-BASIC has its own advantages and drawbacks.

The selection of P1.6 as the memory/peripheral control line of the system fits in well with any future integration of the 8052 chip into the system; BASIC-52 uses P1.6 as a DMA acknowledge line, as part of the 'fake DMA' offered by the system. This 'DMA' is in fact a low-speed interrupt-based system that is too slow to be compatible with some of the (relatively) high-speed data transfers required by some interfaces - e.g., HPIB - that the system supports . As all of the other port 1 lines (see Appendix 8) are of potential use, then this is probably the best line to retain for the purpose. BASIC-52's ability to communicate with assembler subroutines means that a small peripheral service routine may be written in assembler which could perform the same functions as the 'XBY' command, but additionally would manipulate the P1.6 signal line. However, special BASIC-52 CPU and RAM cards would also have to be designed, with careful thought given to the avoidance of address decode conflicts.

CHAPTER 9 - PROJECT REVIEW

Looking back on the work described in this thesis, it can be said that the project as a whole was of benefit to the Company, and provided a firm foundation for future developments. The pilot scheme described in Chapter 2 was in itself an interesting and rewarding project - a duplicate system was in fact installed in another of the Company's sites to meet an urgent requirement for an alarm monitoring system for remote areas. The RF coaxial line communications system used in this installation was found to perform well over distances of the order of five hundred metres.

Possibly it might be worth developing the original node system used in the pilot scheme into a small single-board device as a cheap alternative to the sub-miniature rack system, i.e., a device compatible with the rack system as regards the RF communication link, but offering a cheaper alternative for situations that require the monitoring of ON/OFF sensor devices. Clearly, the T.D.M. (Time Division Multiplexing) protocol used for the evaluation system would be unsuitable for more sophisticated applications involving a large number of nodes. The obvious solution would be to impress a standard protocol on the system using one of the commercially available chips-sets. Two common protocols that spring to mind are Ethernet and S.D.L.C. (Synchronous Data Link Control); despite the implications of acronym, S.D.L.C. is frequently used as the protocol for asynchronous systems - PLL's being employed to recover a clock from an asynchronous data stream, N.R.T.Z. (Non Return To Zero) encoding (see Reference 28) being used to guarantee the necessary level transitions on the incoming data signals.

A recently introduced variant of the 8751, the 8744, offers possibilities in this direction. This chip is in essence, on a single die, an S.D.L.C. protocol controller, buffer memory, and 8751; externally, the

device is pin-compatible with the 8751, except that the serial port is used purely for the reception/transmission of S.D.L.C. data. Thus the introduction of S.D.L.C., for example, need not require major modifications in hardware.

The 8744 is, of course, an ideal choice as an alternative CPU for the CPU board is the rack system itself, and with the addition of a protocol such as S.D.L.C., the system assumes a new power. The 8751 and 8744 are both relatively expensive chips, because of the EPROM content. If the user is prepared to hold programs in external EPROM, then the CPU chip price drops significantly. For example, the ROM-less equivalent of the 8751 is the 8031 microprocessor, which is identical in function to the former but relies on external EPROM for program storage. The chip may be fitted to the existing CPU board without the need for major design modifications.

Addition of external EPROM facilities is one way in which the system may be improved. The CPU architecture allows for the addressing of up to 64K of external EPROM; it is a simple matter to alter the existing RAM board design to allow for the fitting of EPROM chips - the pin-out of both RAM and EPROM chips conform to international (JEDEC) standards.

It was felt that the design of the RAM board was acceptable, although the design will almost certainly be rationalized in the future as higher density memories become available. The IRAM's, originally cost effective against other memory devices, were only marginally so in relation to some of the 8K SRAM's available on the market as the project work approached completion. Some of the Japanese 8Kx8K parts tested (see Appendix 11) were acceptable replacements, and offered faster operating speeds. It was felt that the IRAM was, in concept, a promising device - despite the unfavourable competition from the cheap SRAM's.

The DVI board performed well under test, and several site

installations using prototype boards are operating satisfactorily at the time of writing. The parallel bus handshake options have not, as yet, been utilized, but as there is no cost penalty in manufacturing the boards without the extra components, it is felt that the (potential) extra flexibility the options afford make their inclusion in the design worthwhile. As indicated in the previous chapter, a complementary 'DGO' Board has been designed as a future enhancement to the system, to allow eight digital outputs to be driven.

The bargraph display on the front panel of the DVI (and DGO) boards provided an elegant and efficient monitor of the digital inputs, but it had several drawbacks. Firstly, the ten-element modules were difficult to connect to the board, as standard right-angled DIL sockets were available only as sixteen-pin items; the thick, rectangular leads of these sockets caused problems with the pad sizes on the PCB artworks. Also, the front panel cut-out, hand cut on the prototypes, would have been expensive to die-press in production. Somewhat reluctantly, it was decided to switch to standard Schroff right-angled stacking LED's for final production. The hole matrix in the front panel could be easily produced by the Contractors using standard stock tools.

The Watchdog Board design was considered to be satisfactory, with the exception of the original Cannon connector,which was replaced by a SIL Weidmuller part (Type GZ8), as shown in Figure 3.6. The Cannon part was found to be satisfactorily for installations where the whole rack system was bolted to a backplate of a standard industrial enclosure, but it proved unsuitable to those applications of the system where the rack was housed in a free-standing instrument case - e.g. for bench use. In the latter example, the power supply was usually housed in the back of the instrument case, causing problems in cabling to the plug and socket arrangement on the front panel. The proposed production system features an extra board-mounted

Weidmuller (straight) socket, in addition to the Weidmuller (right-angled) socket for the front panel. This allows the cable to be inserted into a socket enclosed within the rack, thus avoiding an unsightly cable loop on the Watchdog Board front panel.

Of the communications interfaces covered in Chapter 5, the UART board was the most satisfactory design, as it featured the most acceptable component layout with a passable track density. Both the BCD and HPIB Board layouts were considered too cramped, and although the boards functioned correctly, it was felt that these two boards were the ones in the system with unacceptably high track densities. This should not be a problem for small-scale production runs, but would pose problems on on large volume manufacturing involving wave soldering methods. It is difficult to foresee how these problems might be eased, considering the size of the major chips on each board, and the high chip density.

In the case of the HPIB Board, the board supports only the Talker and Listener functions; this is a limitation of the 96LS488 chip, an otherwise excellent part. The implementation of the Controller function requires the employment of a chip-set from another source - Texas, Motorola and Intel, for example, all offer suitable devices, but, unlike the Fairchild part, they require external buffers to connect to the HPIB. This involves, typically, three major chips - making an acceptable board layout difficult to achieve.

The Analogue and Serial Boards were both fairly straight-forward design implementations, and both should be suitable for volume production. The RF Board caused problems during development, and a major re-design around a PLL with faster response characteristics is desirable, as explained in Chapter 6 and the related Appendices. The lessons learnt from

the development of the prototype board should prove useful in the future re-design work.

Each individual board in the system is at a different stage of its development at the time of writing this Review. For example, an early prototype of the BCD board had its own reset components on-board, as it has an active LOW reset requirement, and at that time the Watchdog Board provided only a single (active HIGH) signal on the backplane. The addition of the voltage supervisor chip to the watchdog board provided the active LOW signal, prompting the removal of the reset circuitry from the BCD Board. This has further implications for the Watchdog Board, whose timing components have to cater for the reset pulse specifications of all the boards in the system. The final development of the system in therefore seen to be cyclical in nature, as changes in any one board in the system are seen to impinge on the other boards. Changes to the CPU clock rate, for example, also affects the related backplane clock signals, which time the UART card and determine the scanning rate of the ADC on the Analog Board.

In summary, the development work described in this project has progressed a long way towards achieving the design goals set at the commencement of the work. A prototype distributed interfacing system has been developed, tested and installed on several of the Company's UK sites. The test systems installed perform useful tasks monitoring a variety of analog and digital signals on a number of projects. Although the work has still some way to progress before full-scale production can be contemplated, it can be said that the work performed throughout this project has laid the foundations for the future production of a flexible, reliable and useful production monitoring system.

REFERENCES

1. Intel Corp., The MCS-51 Family of Single Chip Microcomputers User's Manual, July 1981.

2. Signetics Corp., The Linear Components Databook, June 1980.

3. STC Electronic Services, 'A Watchdog For Microcomputer Systems', 'Distribution', Issue 21, p. 15, September 1984.

4. Parkinson, C.N., 'In-Laws and Outlaws', p. 102, John Murray, Oct. 1952.

5. Intel Corp., The Peripheral Design Handbook, August 1980.

6. Lesea, A. & Zaks, R., 'Microprocessor Interfacing Techniques', Sybex Inc., 2nd Edition, 1978.

7. Fisher, J.B., 'Report on the Characterization of the Crystal Oscillator On Board the Intel 8031/8051/8751 Microcontroller', Standard Crystal Corp. Design Note 2F, undated.

8. Williamson, T., 'Oscillators for Microcontrollers', Intel Application Note AP-155, June 1983.

9. Intel Corp., The Component Data Catalog, January 1981

10. Intel Corp., The Memory Components Handbook, 1983

11. Fallin, J.J., & Righter, W.H., 'Designing Memory Systems with the 8Kx8 iRAM', Intel Application Note AP-132, June 1982

12. Seyer M.D., 'RS232 Made Easy', Prentice Hall Inc., 1984.

13. Institute of Electrical and Electronic Engineers Inc., IEEE Standard Digital Interface for Programmable Instrumentation (ANSI/IEEE Std 488-1978), IEEE Inc., November 1978.

14. Fairchild Camera and Inst. Corp., 96LS488 Data Sheet, June 1981.

15. Summers, J., 'Microprocessor GPIB Interfacing With the 96LS488', Fairchild Application Note 351, June 1981.

16. Intersil Datel (UK), Ltd., Intersil Data Book, Ch. 8,1981.

17. Intel Corp., 'Designing Microcontroller Systems for Noisy Environments', Intel Application Note AFN-02131A, 1983.

18. Intel Corp., iPDS Personal Development System User's Guide, 1983.

19. Intel Corp., EMV-51 Emulation Vehicle User's Guide, 1982

20. Katausky, J., MCS BASIC-52 User's Manual, Intel Corp., Pub. No. 270010-001 (1984).

21. Intel Corp., 8251A/S2657 Programmable Communications Interface Data, Application Note AFN-01573A, 1980.

22. Intel Corp., 'Using the 8251A Universal Synchronous/Asynchronous Receiver-Transmitter', Application Note AFN-00600A, 1980.

23. Fairchild Camera and Inst. Corp., Linear Interface Data Book, , 1978.

24. Lancaster, D., 'The TTL Cookbook', The Howard W. Sams & Co. Inc.,1974.

25. Carter, G.W. & Richardson, A., 'Techniques of Circuit Analysis', p 461ff., Cambridge University Press, 1972.

26. Analog devices, AD7581 Data Sheet , Analog Devices Inc., (Undated).

27. Tucholski, H. & Wynne, J., 'DAS Packs Memory for Easy Microprocessor Interface', Electronic Design, September 13th, 1980.

28. IBM, SDLC General Information, GA27-3093-2, 3rd Ed., March 1979.

29. Schroff Equipment Catalogue, 1985.

30. Electronic Designs Inc., EDH8832C-12/15/20 Data Sheet, 1986.

31. Intel Corp., MCS 48 Family Single Chip User's Handbook, Sept. 1981.

32. Intel Corp., 8048 Family Applications Handbook, January 1980.

33. International Telecoms. Union,CCITT White Book, Vol. 8, 1969.

Appendix 1 - Crystals/Oscillator Components

The author is indebted to Intel Technical support for in-depth discussions concerning the MCS-51 Family oscillator components, and for providing the two excellent references (Fisher - Ref. 7 and Williamson - Ref. 8). The information given below is a summary of the practical experience gained from the experimental application of the theoretical data gleaned from the References.

The experimental circuit used by Fisher (Ref.7) gave values of 30pF for the X1, X2 shunt capacitors, but 22pf values were used in the CPU Board circuit implementation, as these are standard items from the Contractor's stock. These capacitors are needed to prevent the extremely small input capacitance of the internal F.E.T.'s of the 8751's Pierce Oscillator "pulling" the crystal up beyond its quoted frequency of oscillation. If required, the oscillator frequency may be "tweaked" by adding asymmetric (e.g. 5pF) trimming capacitors. Reference 7 contains all the details needed by the circuit designer in order to specify crystals to a manufacturer,including ESR (Equivalent Series Resistance) and Shunt Capacity graphs. Note that the oscillator tolerates quite large valued shunt capacitors - e.g. 100 pF, whilst still maintaining operation.

Best operation and stability (especially temperature stability) is obtained using the on-board oscillator. Reference 8 offers a number of alternative ways of obtaining a clock signal from the oscillator to drive external devices. Three of these options were tested ; two of the options (Ref. 8, p. 7, Fig. 9) used discrete components, notably npn transistors loosely coupled to X1 or X2. The circuit working off the X1 pin was found to operate reliably using a BC107 as the test transistor, whilst the alternative design malfunctioned under test. A third solution proposed in the Reference - a simple buffer used to provide the necessary drive for

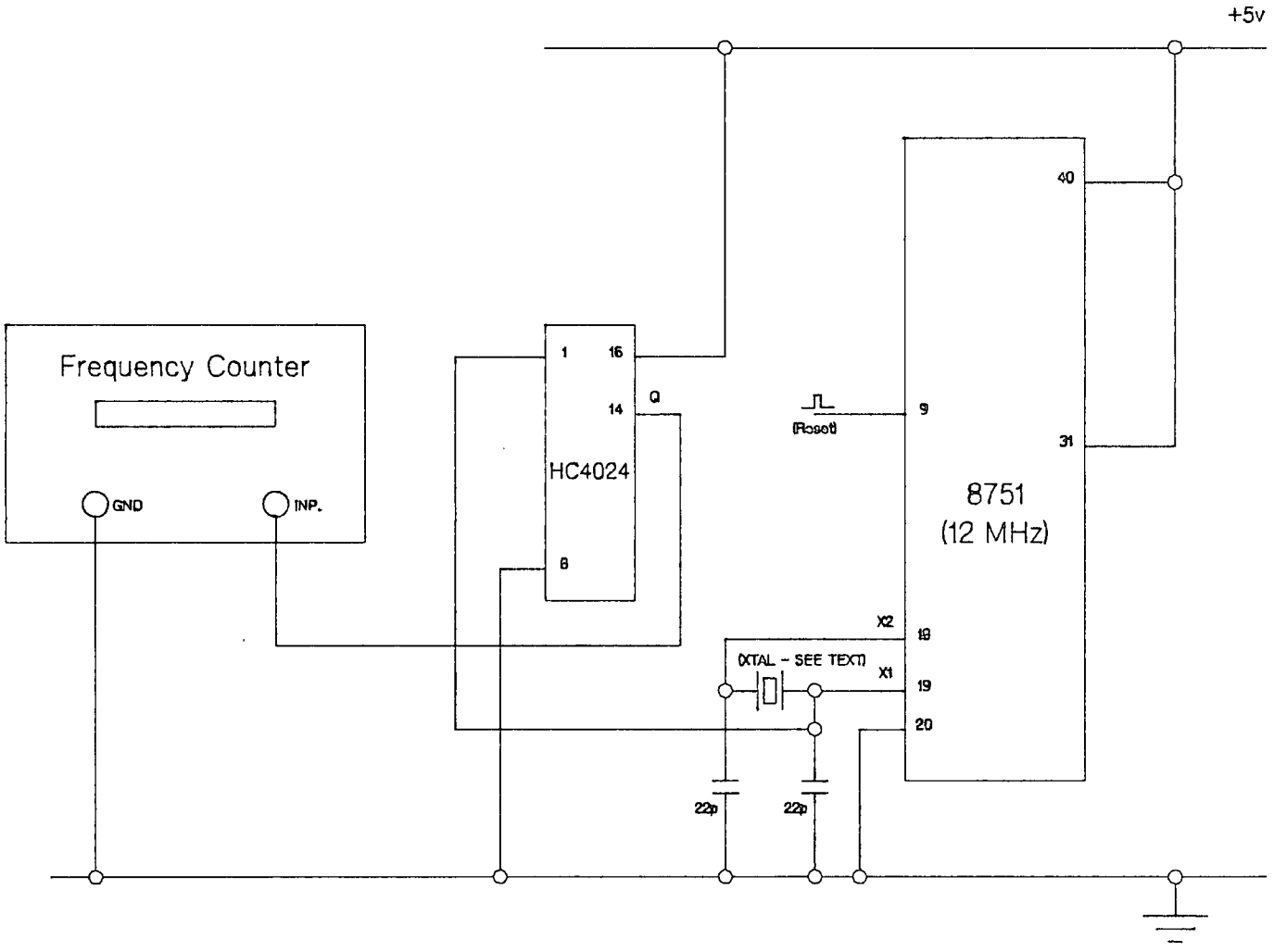
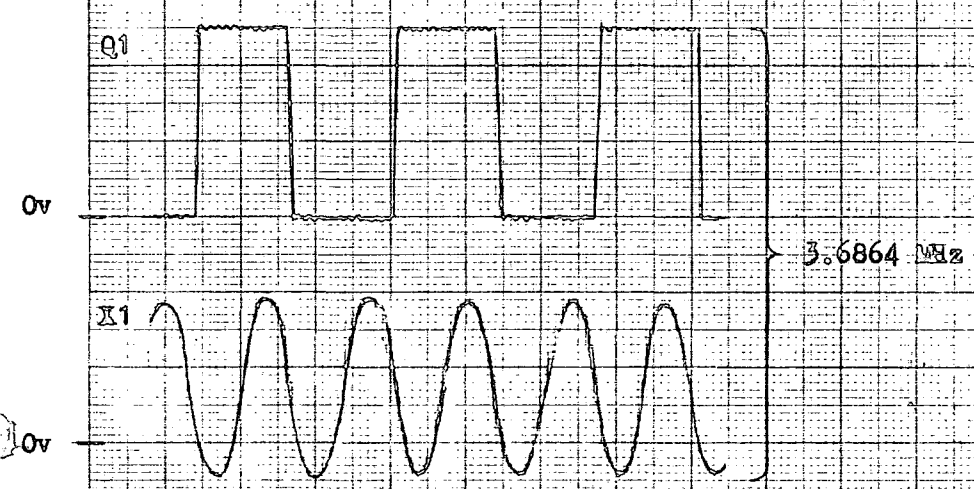
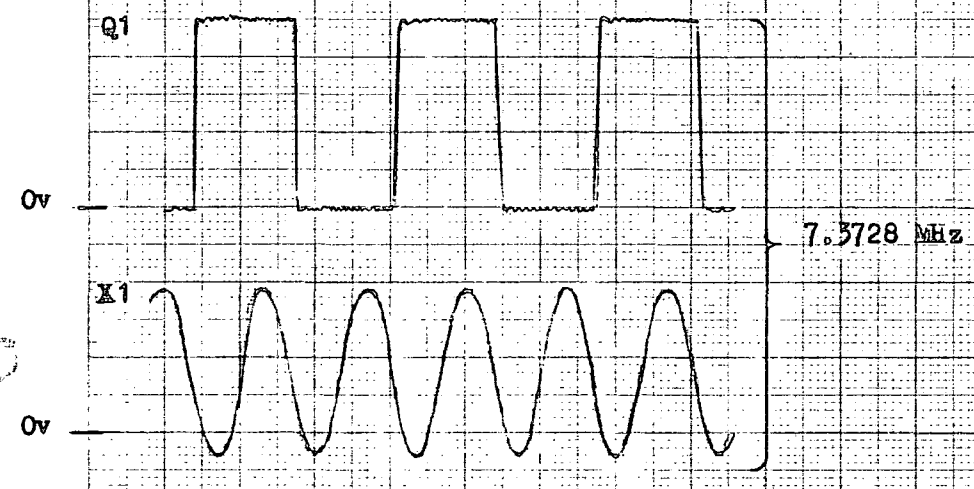


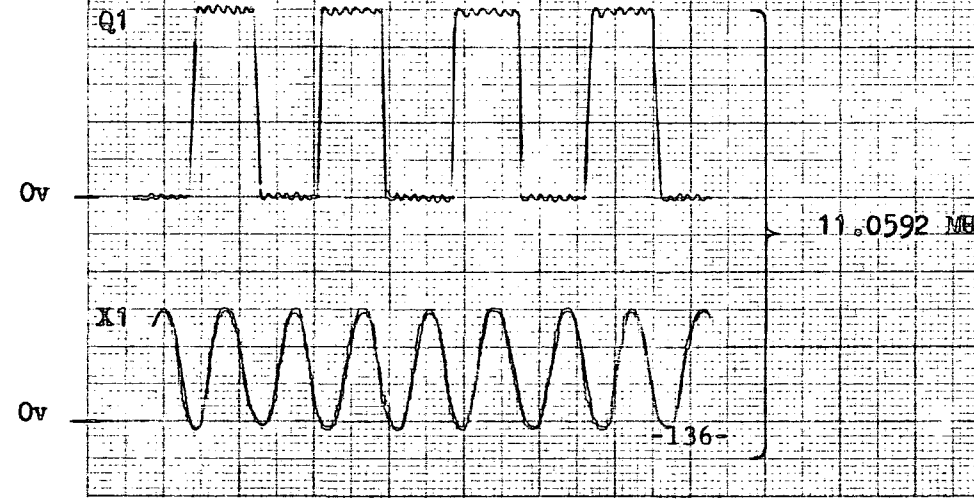
Figure A1.1 - CPU Oscillator Tests

Figure A1.2 – CPU Clock Waveforms for Three Selected Xtal Frequencies



Measurements taken with Phillips PM3215 oscilloscope
0.1uS/div horizontal. 2v/div vertical.





the backplane signal lines - is not without problems; TTL buffers provide excessive loading on X1 or X2 , whilst 'B' series CMOS parts are too slow and have appreciable input capacitance.

An ideal emerging logic family with the required parameters (high speed coupled with low loading properties) for this type of application is HCMOS. This logic family (the range of components is not, as yet , extensive) compares favourably with TTL as regards speed, and has the minimal current consumption and high noise immunity attributes associated with CMOS parts. Figure A1.1 shows the test circuit used to evaluate the operation of the device.

A precursory examination of the X1, X2 clock waveforms with an oscilloscope (50 MHz Phillips PM3215) showed that at 3.6864 MHz the X1 output was a pure sine wave slightly in excess of 5v peak-to-peak. The X2 output was seen as a signal of reduced amplitude 180 degrees out of phase with the X1 signal; an examination of each cycle of the waveform showed that the positive-going half-cycle was sinusoidal, whilst the negative-going half cycle was a linear ramp. (The signal resembled a distorted square wave). Both the X1 and X2 signals were attenuated at higher frequencies , and the X2 output was unable to drive the '4024 part (HCMOS clock divider) at 7.3728 MHz - the inputs require 3.15v minimum to guarantee recognition of the logic one state.

Figure A1.2 shows the observed outputs using the X1 signal. At all test frequencies the Q1 output of the '4024 was a clean 50% duty cycle square wave exhibiting minimal ringing. The 3.6864, 7.3728 MHz tests were performed for the 8751, whilst the 11.0592 MHz tests related to the 8052-BASIC chip (Chapter 8). It was obvious that at higher frequencies the X1 signal amplitude was affected was affected by the capacitance of the 'scope leads. Figure A1.2 shows the X1 amplitude as marginal for the '4024

input , but this is almost certainly a loading effect - the Q1 output was stable. Possibly a light pull-up (e.g. 100k) could be used on production boards operating at this frequency.

The 3.6864 MHz crystal was purchased off-the-shelf from a major component supplier and was quoted for parallel resonance. The other crystals were made to order by S.E.I. (Salford Electrical Instruments), to the following specifications:

Style:	ATQC1622
Frequency:	7.3728/11.0592 MHz
Accuracy:	+/- 50p.p.m. at 25deg. C.
Shunt C.:	22pF
Mode:	Parallel Resonance

The table below summarizes the results of frequency measurements taken with a Phillips PM6670 frequency meter.

Quoted Xtal Frequency	Q1 (Theoretical)	Q1 (Measured)
3.6864	1.8432	1.84363
7.3728	3.6864	3.68640
11.0592	5.5296	5.52960

(Measurements of the ALE frequency confirmed the inaccuracy of the 3.6864 MHz crystal frequency; possibly the part was misquoted for parallel resonance).

The oscillator frequency is one of the controlling parameters for the generation of the baud rate of the 8751's on-board UART. Enclosed with this Appendix is a list of theoretical baud rates obtainable using each of the (three) crystals mentioned previously; data concerning other useful or common crystal selections is also included. A listing of the HP86 program used to generate the data is enclosed with Appendix 3; the latter also provides details of the 8751 serial port.

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	9600	-65	148	-129	74	-193	50
-2	4800	-66	145	-130	74	-194	49
-3	3200	-67	143	-131	73	-195	49
-4	2400	-68	141	-132	73	-196	49
-5	1920	-69	139	-133	72	-197	49
-6	1600	-70	137	-134	72	-198	48
-7	1371	-71	135	-135	71	-199	48
-8	1200	-72	133	-136	71	-200	48
-9	1067	-73	132	-137	70	-201	48
-10	960	-74	130	-138	70	-202	48
-11	873	-75	128	-139	69	-203	47
-12	800	-76	126	-140	69	-204	47
-13	738	-77	125	-141	68	-205	47
-14	686	-78	123	-142	68	-206	47
-15	640	-79	122	-143	67	-207	46
-16	600	-80	120	-144	67	-208	46
-17	565	-81	119	-145	66	-209	46
-18	533	-82	117	-146	66	-210	46
-19	505	-83	116	-147	65	-211	45
-20	480	-84	114	-148	65	-212	45
-21	457	-85	113	-149	64	-213	45
-22	436	-86	112	-150	64	-214	45
-23	417	-87	110	-151	64	-215	45
-24	400	-88	109	-152	63	-216	44
-25	384	-89	108	-153	63	-217	44
-26	369	-90	107	-154	62	-218	44
-27	356	-91	105	-155	62	-219	44
-28	343	-92	104	-156	62	-220	44
-29	331	-93	103	-157	61	-221	43
-30	320	-94	102	-158	61	-222	43
-31	310	-95	101	-159	60	-223	43
-32	300	-96	100	-160	60	-224	43
-33	291	-97	99	-161	60	-225	43
-34	282	-98	98	-162	59	-226	42
-35	274	-99	97	-163	59	-227	42
-36	267	-100	96	-164	59	-228	42
-37	259	-101	95	-165	58	-229	42
-38	253	-102	94	-166	58	-230	42
-39	246	-103	93	-167	57	-231	42
-40	240	-104	92	-168	57	-232	41
-41	234	-105	91	-169	57	-233	41
-42	229	-106	91	-170	56	-234	41
-43	223	-107	90	-171	56	-235	41
-44	218	-108	89	-172	56	-236	41
-45	213	-109	88	-173	55	-237	41
-46	209	-110	87	-174	55	-238	40
-47	204	-111	86	-175	55	-239	40
-48	200	-112	86	-176	55	-240	40
-49	196	-113	85	-177	54	-241	40
-50	192	-114	84	-178	54	-242	40
-51	188	-115	83	-179	54	-243	40
-52	185	-116	83	-180	53	-244	39
-53	181	-117	82	-181	53	-245	39
-54	178	-118	81	-182	53	-246	39
-55	175	-119	81	-183	52	-247	39
-56	171	-120	80	-184	52	-248	39
-57	168	-121	79	-185	52	-249	39
-58	166	-122	79	-186	52	-250	38
-59	163	-123	78	-187	51	-251	38
-60	160	-124	77	-188	51	-252	38
-61	157	-125	77	-189	51	-253	38
-62	155	-126	76	-190	51	-254	38
-63	152	-127	76	-191	50	-255	38
-64	150	-128	75	-192	50	-256	38

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 3.6864MHz XTAL
****

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	14400	-65	222	-129	112	-193	75
-2	7200	-66	218	-130	111	-194	74
-3	4800	-67	215	-131	110	-195	74
-4	3600	-68	212	-132	109	-196	73
-5	2880	-69	209	-133	108	-197	73
-6	2400	-70	206	-134	107	-198	73
-7	2057	-71	203	-135	107	-199	72
-8	1800	-72	200	-136	106	-200	72
-9	1600	-73	197	-137	105	-201	72
-10	1440	-74	195	-138	104	-202	71
-11	1309	-75	192	-139	104	-203	71
-12	1200	-76	189	-140	103	-204	71
-13	1108	-77	187	-141	102	-205	70
-14	1029	-78	185	-142	101	-206	70
-15	960	-79	182	-143	101	-207	70
-16	900	-80	180	-144	100	-208	69
-17	847	-81	178	-145	99	-209	69
-18	800	-82	176	-146	99	-210	69
-19	758	-83	173	-147	98	-211	68
-20	720	-84	171	-148	97	-212	68
-21	686	-85	169	-149	97	-213	68
-22	655	-86	167	-150	96	-214	67
-23	626	-87	166	-151	95	-215	67
-24	600	-88	164	-152	95	-216	67
-25	576	-89	162	-153	94	-217	66
-26	554	-90	160	-154	94	-218	66
-27	533	-91	158	-155	93	-219	66
-28	514	-92	157	-156	92	-220	65
-29	497	-93	155	-157	92	-221	65
-30	480	-94	153	-158	91	-222	65
-31	465	-95	152	-159	91	-223	65
-32	450	-96	150	-160	90	-224	64
-33	436	-97	148	-161	89	-225	64
-34	424	-98	147	-162	89	-226	64
-35	411	-99	145	-163	88	-227	63
-36	400	-100	144	-164	88	-228	63
-37	389	-101	143	-165	87	-229	63
-38	379	-102	141	-166	87	-230	63
-39	369	-103	140	-167	86	-231	62
-40	360	-104	138	-168	86	-232	62
-41	351	-105	137	-169	85	-233	62
-42	343	-106	136	-170	85	-234	62
-43	335	-107	135	-171	84	-235	61
-44	327	-108	133	-172	84	-236	61
-45	320	-109	132	-173	83	-237	61
-46	313	-110	131	-174	83	-238	61
-47	306	-111	130	-175	82	-239	60
-48	300	-112	129	-176	82	-240	60
-49	294	-113	127	-177	81	-241	60
-50	288	-114	126	-178	81	-242	60
-51	282	-115	125	-179	80	-243	59
-52	277	-116	124	-180	80	-244	59
-53	272	-117	123	-181	80	-245	59
-54	267	-118	122	-182	79	-246	59
-55	262	-119	121	-183	79	-247	58
-56	257	-120	120	-184	78	-248	58
-57	253	-121	119	-185	78	-249	58
-58	248	-122	118	-186	77	-250	58
-59	244	-123	117	-187	77	-251	57
-60	240	-124	116	-188	77	-252	57
-61	236	-125	115	-189	76	-253	57
-62	232	-126	114	-190	76	-254	57
-63	229	-127	113	-191	75	-255	56
-64	225	-128	113	-192	75	-256	56

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 5.5296MHz XTAL
****

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	19200	-65	295	-129	149	-193	99
-2	9600	-66	291	-130	148	-194	99
-3	6400	-67	287	-131	147	-195	98
-4	4800	-68	282	-132	145	-196	98
-5	3840	-69	278	-133	144	-197	97
-6	3200	-70	274	-134	143	-198	97
-7	2743	-71	270	-135	142	-199	96
-8	2400	-72	267	-136	141	-200	96
-9	2133	-73	263	-137	140	-201	96
-10	1920	-74	259	-138	139	-202	95
-11	1745	-75	256	-139	138	-203	95
-12	1600	-76	253	-140	137	-204	94
-13	1477	-77	249	-141	136	-205	94
-14	1371	-78	246	-142	135	-206	93
-15	1280	-79	243	-143	134	-207	93
-16	1200	-80	240	-144	133	-208	92
-17	1129	-81	237	-145	132	-209	92
-18	1067	-82	234	-146	132	-210	91
-19	1011	-83	231	-147	131	-211	91
-20	960	-84	229	-148	130	-212	91
-21	914	-85	226	-149	129	-213	90
-22	873	-86	223	-150	128	-214	90
-23	835	-87	221	-151	127	-215	89
-24	800	-88	218	-152	126	-216	89
-25	768	-89	216	-153	125	-217	88
-26	738	-90	213	-154	125	-218	88
-27	711	-91	211	-155	124	-219	88
-28	686	-92	209	-156	123	-220	87
-29	662	-93	206	-157	122	-221	87
-30	640	-94	204	-158	122	-222	86
-31	619	-95	202	-159	121	-223	86
-32	600	-96	200	-160	120	-224	86
-33	582	-97	198	-161	119	-225	85
-34	565	-98	196	-162	119	-226	85
-35	549	-99	194	-163	118	-227	85
-36	533	-100	192	-164	117	-228	84
-37	519	-101	190	-165	116	-229	84
-38	505	-102	188	-166	116	-230	83
-39	492	-103	186	-167	115	-231	83
-40	480	-104	185	-168	114	-232	83
-41	468	-105	183	-169	114	-233	82
-42	457	-106	181	-170	113	-234	82
-43	447	-107	179	-171	112	-235	82
-44	436	-108	178	-172	112	-236	81
-45	427	-109	176	-173	111	-237	81
-46	417	-110	175	-174	110	-238	81
-47	409	-111	173	-175	110	-239	80
-48	400	-112	171	-176	109	-240	80
-49	392	-113	170	-177	108	-241	80
-50	384	-114	168	-178	108	-242	79
-51	376	-115	167	-179	107	-243	79
-52	369	-116	166	-180	107	-244	79
-53	362	-117	164	-181	106	-245	78
-54	356	-118	163	-182	105	-246	78
-55	349	-119	161	-183	105	-247	78
-56	343	-120	160	-184	104	-248	77
-57	337	-121	159	-185	104	-249	77
-58	331	-122	157	-186	103	-250	77
-59	325	-123	156	-187	103	-251	76
-60	320	-124	155	-188	102	-252	76
-61	315	-125	154	-189	102	-253	76
-62	310	-126	152	-190	101	-254	76
-63	305	-127	151	-191	101	-255	75
-64	300	-128	150	-192	100	-256	75

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 7.3728MHz XTAL
****

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	20833	-65	321	-129	161	-193	108
-2	10417	-66	316	-130	160	-194	107
-3	6944	-67	311	-131	159	-195	107
-4	5208	-68	306	-132	158	-196	106
-5	4167	-69	302	-133	157	-197	106
-6	3472	-70	298	-134	155	-198	105
-7	2976	-71	293	-135	154	-199	105
-8	2604	-72	289	-136	153	-200	104
-9	2315	-73	285	-137	152	-201	104
-10	2083	-74	282	-138	151	-202	103
-11	1894	-75	278	-139	150	-203	103
-12	1736	-76	274	-140	149	-204	102
-13	1603	-77	271	-141	148	-205	102
-14	1488	-78	267	-142	147	-206	101
-15	1389	-79	264	-143	146	-207	101
-16	1302	-80	260	-144	145	-208	100
-17	1225	-81	257	-145	144	-209	100
-18	1157	-82	254	-146	143	-210	99
-19	1096	-83	251	-147	142	-211	99
-20	1042	-84	248	-148	141	-212	98
-21	992	-85	245	-149	140	-213	98
-22	947	-86	242	-150	139	-214	97
-23	906	-87	239	-151	138	-215	97
-24	868	-88	237	-152	137	-216	96
-25	833	-89	234	-153	136	-217	96
-26	801	-90	231	-154	135	-218	96
-27	772	-91	229	-155	134	-219	95
-28	744	-92	226	-156	134	-220	95
-29	718	-93	224	-157	133	-221	94
-30	694	-94	222	-158	132	-222	94
-31	672	-95	219	-159	131	-223	93
-32	651	-96	217	-160	130	-224	93
-33	631	-97	215	-161	129	-225	93
-34	613	-98	213	-162	129	-226	92
-35	595	-99	210	-163	128	-227	92
-36	579	-100	208	-164	127	-228	91
-37	563	-101	206	-165	126	-229	91
-38	548	-102	204	-166	126	-230	91
-39	534	-103	202	-167	125	-231	90
-40	521	-104	200	-168	124	-232	90
-41	508	-105	198	-169	123	-233	89
-42	496	-106	197	-170	123	-234	89
-43	484	-107	195	-171	122	-235	89
-44	473	-108	193	-172	121	-236	88
-45	463	-109	191	-173	120	-237	88
-46	453	-110	189	-174	120	-238	88
-47	443	-111	188	-175	119	-239	87
-48	434	-112	186	-176	118	-240	87
-49	425	-113	184	-177	118	-241	86
-50	417	-114	183	-178	117	-242	86
-51	408	-115	181	-179	116	-243	86
-52	401	-116	180	-180	116	-244	85
-53	393	-117	178	-181	115	-245	85
-54	386	-118	177	-182	114	-246	85
-55	379	-119	175	-183	114	-247	84
-56	372	-120	174	-184	113	-248	84
-57	365	-121	172	-185	113	-249	84
-58	359	-122	171	-186	112	-250	83
-59	353	-123	169	-187	111	-251	83
-60	347	-124	168	-188	111	-252	83
-61	342	-125	167	-189	110	-253	82
-62	336	-126	165	-190	110	-254	82
-63	331	-127	164	-191	109	-255	82
-64	326	-128	163	-192	109	-256	81

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 8MHz XTAL
****

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	24000	-65	369	-129	186	-193	124
-2	12000	-66	364	-130	185	-194	124
-3	8000	-67	358	-131	183	-195	123
-4	6000	-68	353	-132	182	-196	122
-5	4800	-69	348	-133	180	-197	122
-6	4000	-70	343	-134	179	-198	121
-7	3429	-71	338	-135	178	-199	121
-8	3000	-72	333	-136	176	-200	120
-9	2667	-73	329	-137	175	-201	119
-10	2400	-74	324	-138	174	-202	119
-11	2182	-75	320	-139	173	-203	118
-12	2000	-76	316	-140	171	-204	118
-13	1846	-77	312	-141	170	-205	117
-14	1714	-78	308	-142	169	-206	117
-15	1600	-79	304	-143	168	-207	116
-16	1500	-80	300	-144	167	-208	115
-17	1412	-81	296	-145	166	-209	115
-18	1333	-82	293	-146	164	-210	114
-19	1263	-83	289	-147	163	-211	114
-20	1200	-84	286	-148	162	-212	113
-21	1143	-85	282	-149	161	-213	113
-22	1091	-86	279	-150	160	-214	112
-23	1043	-87	276	-151	159	-215	112
-24	1000	-88	273	-152	158	-216	111
-25	960	-89	270	-153	157	-217	111
-26	923	-90	267	-154	156	-218	110
-27	889	-91	264	-155	155	-219	110
-28	857	-92	261	-156	154	-220	109
-29	828	-93	258	-157	153	-221	109
-30	800	-94	255	-158	152	-222	108
-31	774	-95	253	-159	151	-223	108
-32	750	-96	250	-160	150	-224	107
-33	727	-97	247	-161	149	-225	107
-34	706	-98	245	-162	148	-226	106
-35	686	-99	242	-163	147	-227	106
-36	667	-100	240	-164	146	-228	105
-37	649	-101	238	-165	145	-229	105
-38	632	-102	235	-166	145	-230	104
-39	615	-103	233	-167	144	-231	104
-40	600	-104	231	-168	143	-232	103
-41	585	-105	229	-169	142	-233	103
-42	571	-106	226	-170	141	-234	103
-43	558	-107	224	-171	140	-235	102
-44	545	-108	222	-172	140	-236	102
-45	533	-109	220	-173	139	-237	101
-46	522	-110	218	-174	138	-238	101
-47	511	-111	216	-175	137	-239	100
-48	500	-112	214	-176	136	-240	100
-49	490	-113	212	-177	136	-241	100
-50	480	-114	211	-178	135	-242	99
-51	471	-115	209	-179	134	-243	99
-52	462	-116	207	-180	133	-244	98
-53	453	-117	205	-181	133	-245	98
-54	444	-118	203	-182	132	-246	98
-55	436	-119	202	-183	131	-247	97
-56	429	-120	200	-184	130	-248	97
-57	421	-121	198	-185	130	-249	96
-58	414	-122	197	-186	129	-250	96
-59	407	-123	195	-187	128	-251	96
-60	400	-124	194	-188	128	-252	95
-61	393	-125	192	-189	127	-253	95
-62	387	-126	190	-190	126	-254	94
-63	381	-127	189	-191	126	-255	94
-64	375	-128	188	-192	125	-256	94

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 9.216MHz XTAL
****

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	28646	-65	441	-129	222	-193	148
-2	14323	-66	434	-130	220	-194	148
-3	9549	-67	428	-131	219	-195	147
-4	7161	-68	421	-132	217	-196	146
-5	5729	-69	415	-133	215	-197	145
-6	4774	-70	409	-134	214	-198	145
-7	4092	-71	403	-135	212	-199	144
-8	3581	-72	398	-136	211	-200	143
-9	3183	-73	392	-137	209	-201	143
-10	2865	-74	387	-138	208	-202	142
-11	2604	-75	382	-139	206	-203	141
-12	2387	-76	377	-140	205	-204	140
-13	2204	-77	372	-141	203	-205	140
-14	2046	-78	367	-142	202	-206	139
-15	1910	-79	363	-143	200	-207	138
-16	1790	-80	358	-144	199	-208	138
-17	1685	-81	354	-145	198	-209	137
-18	1591	-82	349	-146	196	-210	136
-19	1508	-83	345	-147	195	-211	136
-20	1432	-84	341	-148	194	-212	135
-21	1364	-85	337	-149	192	-213	134
-22	1302	-86	333	-150	191	-214	134
-23	1245	-87	329	-151	190	-215	133
-24	1194	-88	326	-152	188	-216	133
-25	1146	-89	322	-153	187	-217	132
-26	1102	-90	318	-154	186	-218	131
-27	1061	-91	315	-155	185	-219	131
-28	1023	-92	311	-156	184	-220	130
-29	988	-93	308	-157	182	-221	130
-30	955	-94	305	-158	181	-222	129
-31	924	-95	302	-159	180	-223	128
-32	895	-96	298	-160	179	-224	128
-33	868	-97	295	-161	178	-225	127
-34	843	-98	292	-162	177	-226	127
-35	818	-99	289	-163	176	-227	126
-36	796	-100	286	-164	175	-228	126
-37	774	-101	284	-165	174	-229	125
-38	754	-102	281	-166	173	-230	125
-39	735	-103	278	-167	172	-231	124
-40	716	-104	275	-168	171	-232	123
-41	699	-105	273	-169	170	-233	123
-42	682	-106	270	-170	169	-234	122
-43	666	-107	268	-171	168	-235	122
-44	651	-108	265	-172	167	-236	121
-45	637	-109	263	-173	166	-237	121
-46	623	-110	260	-174	165	-238	120
-47	609	-111	258	-175	164	-239	120
-48	597	-112	256	-176	163	-240	119
-49	585	-113	254	-177	162	-241	119
-50	573	-114	251	-178	161	-242	118
-51	562	-115	249	-179	160	-243	118
-52	551	-116	247	-180	159	-244	117
-53	540	-117	245	-181	158	-245	117
-54	530	-118	243	-182	157	-246	116
-55	521	-119	241	-183	157	-247	116
-56	512	-120	239	-184	156	-248	116
-57	503	-121	237	-185	155	-249	115
-58	494	-122	235	-186	154	-250	115
-59	486	-123	233	-187	153	-251	114
-60	477	-124	231	-188	152	-252	114
-61	470	-125	229	-189	152	-253	113
-62	462	-126	227	-190	151	-254	113
-63	455	-127	226	-191	150	-255	112
-64	448	-128	224	-192	149	-256	112

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 11MHz XTAL
****

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	28800	-65	443	-129	223	-193	149
-2	14400	-66	436	-130	222	-194	148
-3	9600	-67	430	-131	220	-195	148
-4	7200	-68	424	-132	218	-196	147
-5	5760	-69	417	-133	217	-197	146
-6	4800	-70	411	-134	215	-198	145
-7	4114	-71	406	-135	213	-199	145
-8	3600	-72	400	-136	212	-200	144
-9	3200	-73	395	-137	210	-201	143
-10	2880	-74	389	-138	209	-202	143
-11	2618	-75	384	-139	207	-203	142
-12	2400	-76	379	-140	206	-204	141
-13	2215	-77	374	-141	204	-205	140
-14	2057	-78	369	-142	203	-206	140
-15	1920	-79	365	-143	201	-207	139
-16	1800	-80	360	-144	200	-208	138
-17	1694	-81	356	-145	199	-209	138
-18	1600	-82	351	-146	197	-210	137
-19	1516	-83	347	-147	196	-211	136
-20	1440	-84	343	-148	195	-212	136
-21	1371	-85	339	-149	193	-213	135
-22	1309	-86	335	-150	192	-214	135
-23	1252	-87	331	-151	191	-215	134
-24	1200	-88	327	-152	189	-216	133
-25	1152	-89	324	-153	188	-217	133
-26	1108	-90	320	-154	187	-218	132
-27	1067	-91	316	-155	186	-219	132
-28	1029	-92	313	-156	185	-220	131
-29	993	-93	310	-157	183	-221	130
-30	960	-94	306	-158	182	-222	130
-31	929	-95	303	-159	181	-223	129
-32	900	-96	300	-160	180	-224	129
-33	873	-97	297	-161	179	-225	128
-34	847	-98	294	-162	178	-226	127
-35	823	-99	291	-163	177	-227	127
-36	800	-100	288	-164	176	-228	126
-37	778	-101	285	-165	175	-229	126
-38	758	-102	282	-166	173	-230	125
-39	738	-103	280	-167	172	-231	125
-40	720	-104	277	-168	171	-232	124
-41	702	-105	274	-169	170	-233	124
-42	686	-106	272	-170	169	-234	123
-43	670	-107	269	-171	168	-235	123
-44	655	-108	267	-172	167	-236	122
-45	640	-109	264	-173	166	-237	122
-46	626	-110	262	-174	166	-238	121
-47	613	-111	259	-175	165	-239	121
-48	600	-112	257	-176	164	-240	120
-49	588	-113	255	-177	163	-241	120
-50	576	-114	253	-178	162	-242	119
-51	565	-115	250	-179	161	-243	119
-52	554	-116	248	-180	160	-244	118
-53	543	-117	246	-181	159	-245	118
-54	533	-118	244	-182	158	-246	117
-55	524	-119	242	-183	157	-247	117
-56	514	-120	240	-184	157	-248	116
-57	505	-121	238	-185	156	-249	116
-58	497	-122	236	-186	155	-250	115
-59	488	-123	234	-187	154	-251	115
-60	480	-124	232	-188	153	-252	114
-61	472	-125	230	-189	152	-253	114
-62	465	-126	229	-190	152	-254	113
-63	457	-127	227	-191	151	-255	113
-64	450	-128	225	-192	150	-256	113

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 11.0592MHz XTAL
****

CONST.	BAUD	CONST.	BAUD	CONST.	BAUD	CONST.	BAUD
-1	31250	-65	481	-129	242	-193	162
-2	15625	-66	473	-130	240	-194	161
-3	10417	-67	466	-131	239	-195	160
-4	7813	-68	460	-132	237	-196	159
-5	6250	-69	453	-133	235	-197	159
-6	5208	-70	446	-134	233	-198	158
-7	4464	-71	440	-135	231	-199	157
-8	3906	-72	434	-136	230	-200	156
-9	3472	-73	428	-137	228	-201	155
-10	3125	-74	422	-138	226	-202	155
-11	2841	-75	417	-139	225	-203	154
-12	2604	-76	411	-140	223	-204	153
-13	2404	-77	406	-141	222	-205	152
-14	2232	-78	401	-142	220	-206	152
-15	2083	-79	396	-143	219	-207	151
-16	1953	-80	391	-144	217	-208	150
-17	1838	-81	386	-145	216	-209	150
-18	1736	-82	381	-146	214	-210	149
-19	1645	-83	377	-147	213	-211	148
-20	1563	-84	372	-148	211	-212	147
-21	1488	-85	368	-149	210	-213	147
-22	1420	-86	363	-150	208	-214	146
-23	1359	-87	359	-151	207	-215	145
-24	1302	-88	355	-152	206	-216	145
-25	1250	-89	351	-153	204	-217	144
-26	1202	-90	347	-154	203	-218	143
-27	1157	-91	343	-155	202	-219	143
-28	1116	-92	340	-156	200	-220	142
-29	1078	-93	336	-157	199	-221	141
-30	1042	-94	332	-158	198	-222	141
-31	1008	-95	329	-159	197	-223	140
-32	977	-96	326	-160	195	-224	140
-33	947	-97	322	-161	194	-225	139
-34	919	-98	319	-162	193	-226	138
-35	893	-99	316	-163	192	-227	138
-36	868	-100	313	-164	191	-228	137
-37	845	-101	309	-165	189	-229	136
-38	822	-102	306	-166	188	-230	136
-39	801	-103	303	-167	187	-231	135
-40	781	-104	300	-168	186	-232	135
-41	762	-105	298	-169	185	-233	134
-42	744	-106	295	-170	184	-234	134
-43	727	-107	292	-171	183	-235	133
-44	710	-108	289	-172	182	-236	132
-45	694	-109	287	-173	181	-237	132
-46	679	-110	284	-174	180	-238	131
-47	665	-111	282	-175	179	-239	131
-48	651	-112	279	-176	178	-240	130
-49	638	-113	277	-177	177	-241	130
-50	625	-114	274	-178	176	-242	129
-51	613	-115	272	-179	175	-243	129
-52	601	-116	269	-180	174	-244	128
-53	590	-117	267	-181	173	-245	128
-54	579	-118	265	-182	172	-246	127
-55	568	-119	263	-183	171	-247	127
-56	558	-120	260	-184	170	-248	126
-57	548	-121	258	-185	169	-249	126
-58	539	-122	256	-186	168	-250	125
-59	530	-123	254	-187	167	-251	125
-60	521	-124	252	-188	166	-252	124
-61	512	-125	250	-189	165	-253	124
-62	504	-126	248	-190	164	-254	123
-63	496	-127	246	-191	164	-255	123
-64	488	-128	244	-192	163	-256	122

TABLE OF RELOAD CONSTANTS/BAUD RATES FOR 12MHz XTAL
****

Appendix 2 - Pilot Scheme Data

This Appendix includes further details of performances of RF circuits relating to both the Pilot Scheme (Chapter 2) and the RF Board (Section 6.2), and also provides details of the firmware used in the Pilot Scheme.

VCO Performance

Figure A2.1 shows a plot of output frequency versus modulation voltage for the VCO circuit used in the Pilot Scheme, obtained from the experimental results listed in Table A2.1 below.

Vmod(volts)	Freq.(KHz)	Vmod(volts)	Freq(KHz)
0.0	0.0	2.6	187.1
0.2	0.0	2.8	187.1
0.4	0.0	3.0	187.1
0.6	0.0	3.2	187.1
0.8	185.0	3.4	187.1
1.0	187.0	3.6	187.1
1.2	187.1	3.8	180.2
1.4	187.2	4.0	166.7
1.6	187.2	4.2	165.1
1.8	187.2	4.4	165.0
2.0	187.1	4.6	165.0
2.2	187.1	4.8	165.0
2.4	187.1	5.0	165.0

Table A2.1 - VCO Frequency variation with modulation voltage.

The readings were obtained from a CSC Model 5001 frequency meter which was connected to the output of the VCO (see Figures 2.5 and 2.6), while the modulation voltage was applied to the input using a variable output power supply. It can be seen from Figure A2.1 that there is a relatively stable output (187.1 KHz) around the Voh point (typically 2.4v for the 'LS14), with large frequency variations at voltages on either side. The cut-off at the 0.6v mark indicates that the 'LS14 quenches the oscillator when in the

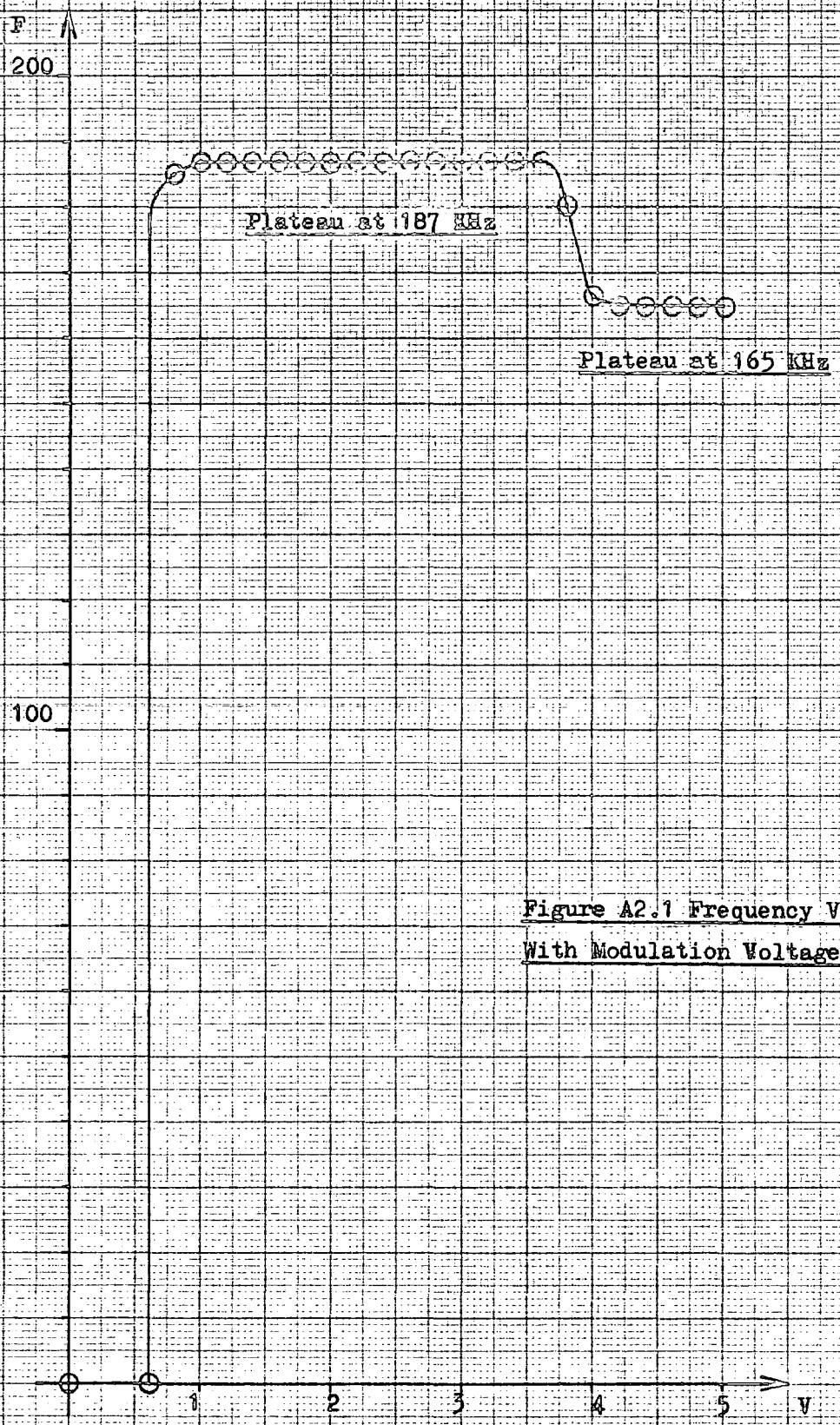


Figure A2.1 Frequency Variation With Modulation Voltage

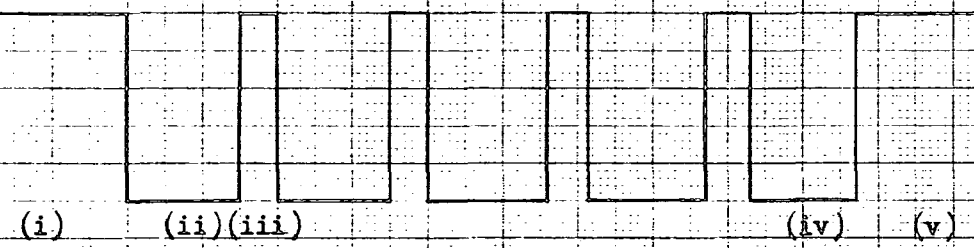
Vol state, theoretically making the making the line gating signal for the 75123 line driver redundant; in practice, the absence of a gating signal was found to affect the accuracy of the data transmission - an effect probably attributable to turn-off transients.

Other factors which affect the VCO output frequency are temperature and supply voltage. Several components in the VCO circuit (as well as the '567 itself), such as trim potentiometers, timing components etc., are also temperature sensitive, and changes to the nominal values of these components due to temperature fluctuations also affect operation.

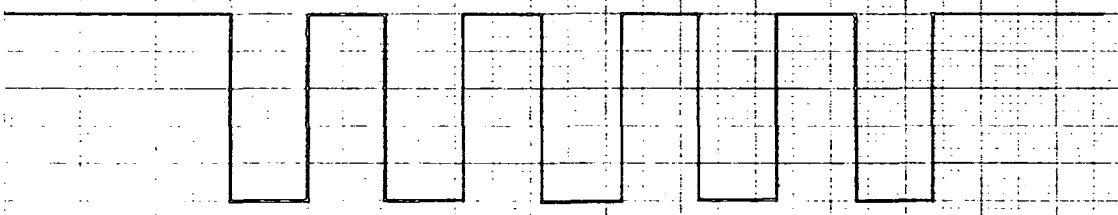
PLL Performance

Figure A4.2 shows the observed and theoretical waveforms for the serial data stream recovered by the PLL's on the RF Board. The observations were taken using a Phillips PM3215 oscilloscope during testing of the RF Board circuit (see Chapter 6). Clearly the PLL's were too slow in responding to the termination of the tone bursts, and this is reflected in the narrowing of the logic '1' pulses in the recovered serial data stream . This is probably attributable to the use of feedback capacitors to speed up loop lock, causing the loop to retain lock for a short period after the termination of the tone burst.

The timing of serial data reception is, with devices such as UART's, in relation to the occurrence of the start bit. In Figure A2.2 the start bit (ii) terminates the line idle state (i) and precedes the first data bit (iii) . UARTs usually time the reception of data bits from the centre of the start bit, which, in theory, in this case predicts problems with the reception of serial data, as the rising edges of the logic '1' pulses are close to the theoretical pulse centre positions, which correspond to the centres of the data sampling periods. (The data sampling period for the 8751 UART, for example, is defined by three samples taken at clock



Observed Waveform



Theoretical Waveform

Notes

(i) Line idle
(ii) Start bit
(iii) LSB of 'U' character
(iv) MSB " " "
(v) Stop bit/line idle

Measurements taken with
Phillips PM3215 oscilloscope
0.1mS/div horizontal
2v/div vertical
Jitter 0.01mS observed on pulses

Figure A2.2 - Serial Data Waveforms

intervals 7, 8 and 9 of a clock running at sixteen times the baud rate . The data is accepted on the basis of a best from three poll).

The 8751's UART was found to be more sensitive to problems associated with pulse narrowing and pulse jitter than the HP85's serial interface UART. Clearly further development work will be necessary before a production board can be manufactured; the choice originally of the '567 as the PLL was determined by the single-rail power supply constraint (for the Pilot Scheme) and by the specification given for an earlier phase of the project - originally the Pilot Scheme was conceived as a 300 Baud single-node system. (The '567 is well-suited for utilization in relatively slow-speed data transfer systems). With the power supply restriction removed, the choice of PLL's is greatly increased, and it is felt that if further development work on the board design is to be undertaken, then it would be prudent to substitute the '567 with a PLL with a faster lock characteristic and better high frequency specification. It can be seen from the information given in Appendix 3 that the 8751 serial port has some useful high data rate options.

Firmware.

Included with this Appendix is a set of flow diagrams relating to the firmware for the Pilot Scheme. Each flow diagram is described separately below.

Polling Firmware.

This firmware consists of s simple closed loop which transmits the 'polling sequence'. The polling sequence comprises the sequence 8,7,6,....1 repeated indefinitely. Before entering the loop the program initializes the 8751's serial port (see Appendix 3). The program polls eight devices, to allow for future expansion beyond the original three-node system.

To begin the polling sequence the program selects the first device by loading the constant eight into the accumulator. The subroutine 'OUCH' is used to output the binary character; after calling 'OUCH' the program pauses for twelve milliseconds (this delay is to allow the polled device to respond and transmit two bytes of sensor data) before decrementing the accumulator and jumping back to repeat the loop for the other seven characters in the polling sequence. The sequence is then repeated.

Monitoring firmware.

After initializing the serial port the monitoring node program inputs a polling character by calling subroutine 'INCH'. The polling character is compared with the node address (held as a constant in EPROM). If the polling character does not match the node address, then the next character is input from the serial port via 'INCH'.

On detecting equality the sensor data (eight bits indicating the state of eight sensors) is read from Port 0 and output to the coaxial line using subroutine 'NOUT'. This process is repeated for Port 1, after which the program jumps to the start of the loop and reads the next polling character.

Subroutine NOUT

This subroutine splits the eight-bit sensor data into two nibbles and adds the constant forty-eight to each nibble. This allows the HP85 (which 'eavesdrops' on the line via a non-intelligent node - see Chapter 2) to differentiate between polling characters (pure binary) and data characters (printing ASCII characters). Each byte (modified nibble) is output to the line using subroutine 'OUCH'.

N.b. It was found necessary to write one's (hexadecimal FF) to either port before reading the data. (Nowhere in the firmware is port 0 or 1 used for the outputting of data).

Subroutines OUCH and INCH

These two subroutines output or input characters respectively to or from the serial port. The buffer 'SBUF' is used to transfer data bytes, and is common to both processes, but the ready (or interrupt) flags - RI and TI - are individual to each function. The serial port sets the appropriate flag when the I/O transfer is complete and the flag is reset by software before the next byte is input or output. Both subroutines incorporate wait loops which test the state of the relevant flag line while the I/O function is in progress.

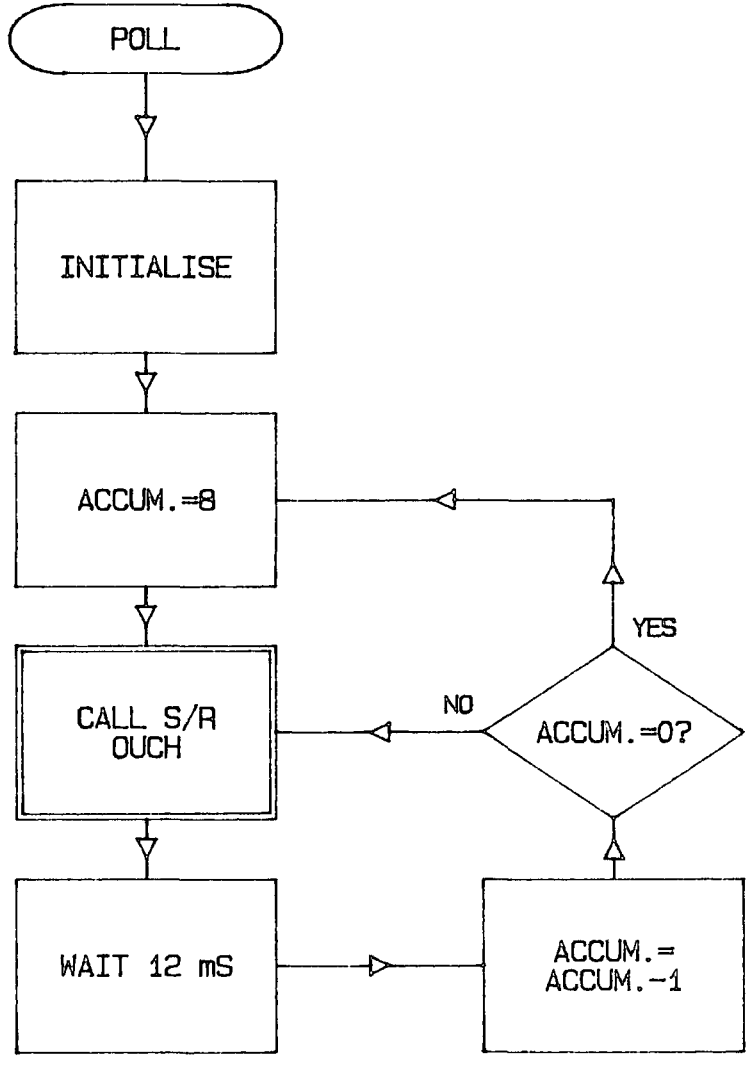


Figure A2.3 - Flowchart of Polling Firmware (Controller Node)

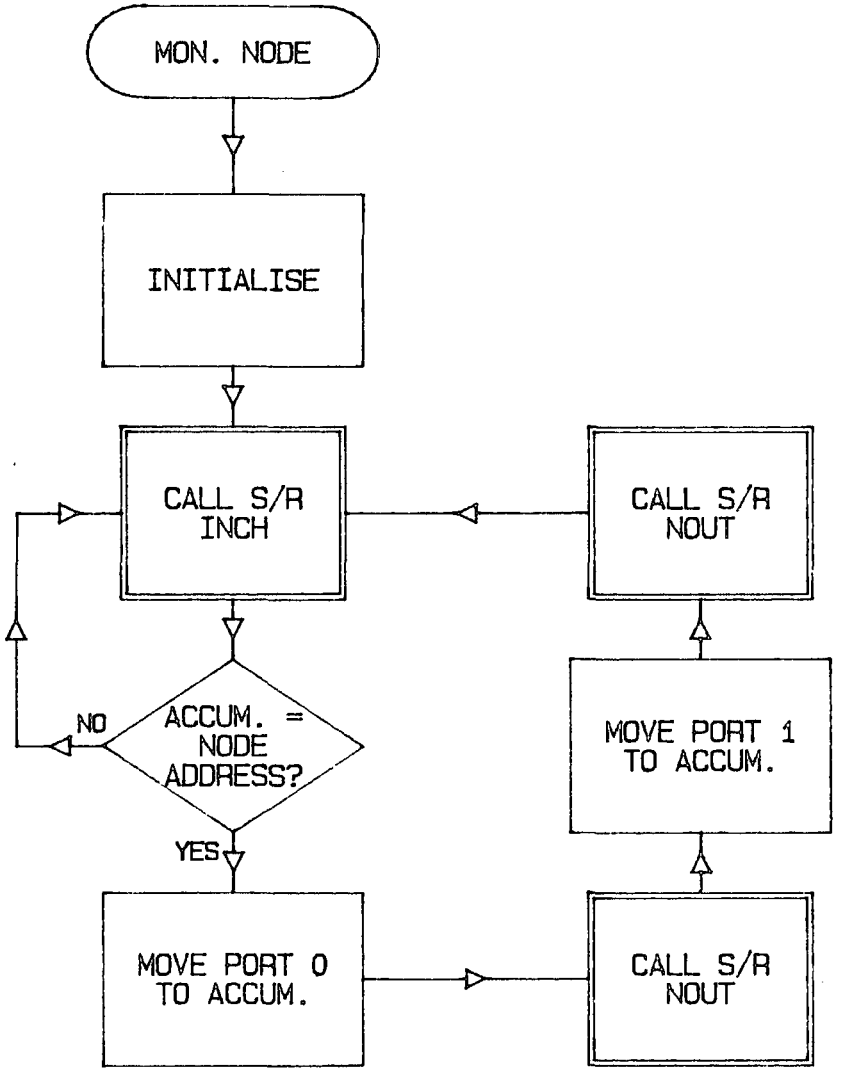


Figure A2.4 - Flowchart of (Monitoring) Node Firmware

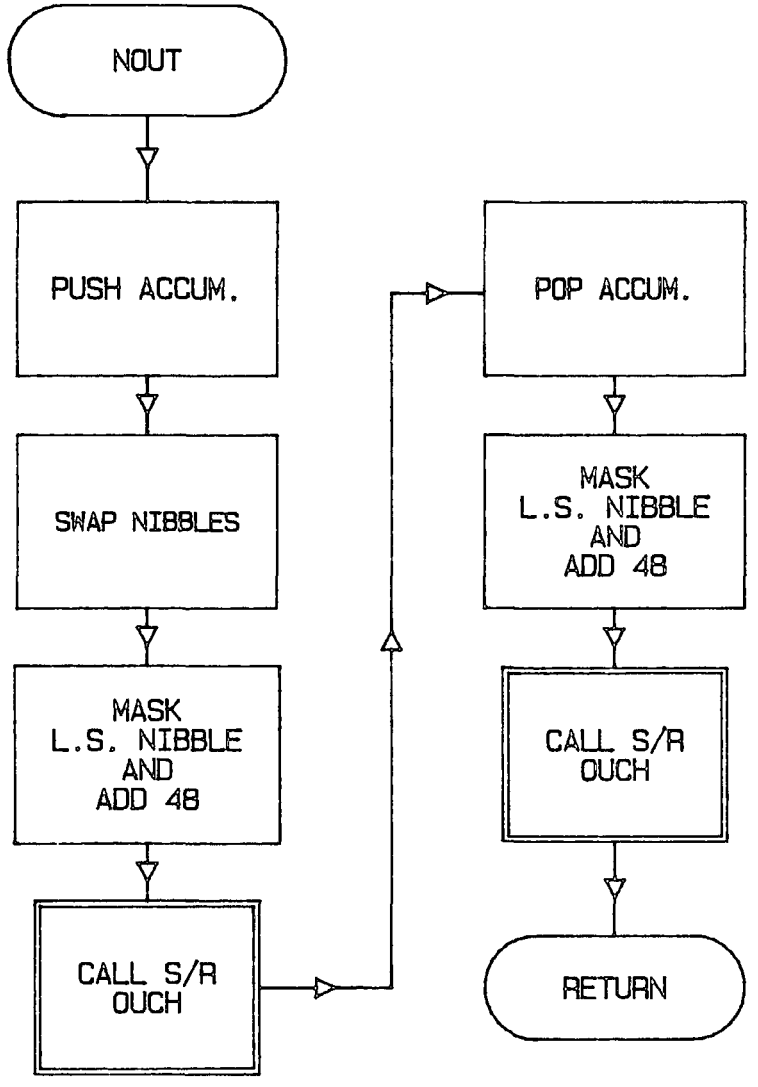
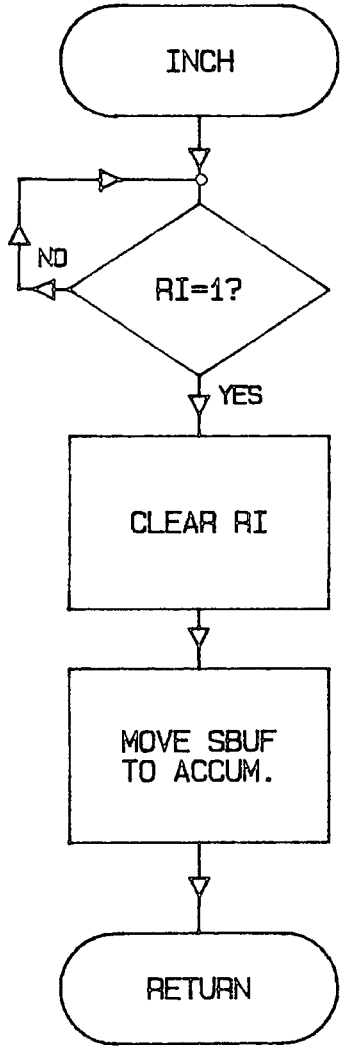


Figure A2.5 - Subroutine "NOUT"



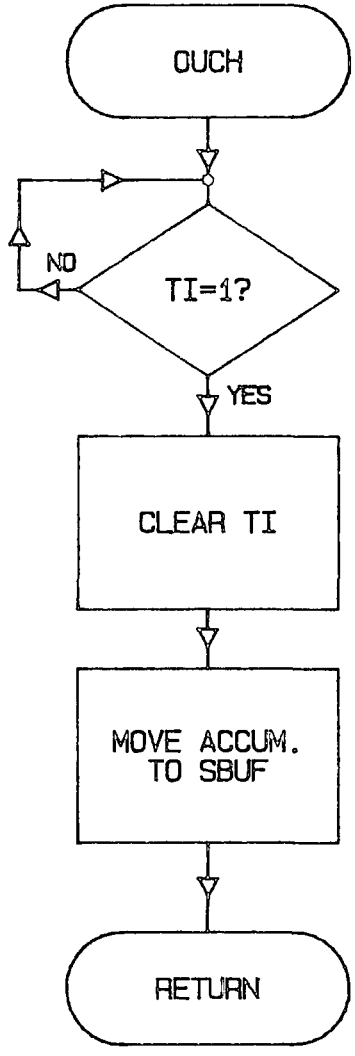


Fig. A2.6a - Subroutine "INCH" Fig. A2.6b - Subroutine "OUCH"

Appendix 3 - The 8751 Serial Port.

The 8751's serial port is a firmware-configurable device capable of supporting full-duplex data transfer . The device can operate in one of four alternative modes, selected by bits six and seven of SCON, the serial port control register.

Mode 0

Mode 0 configures the port as a synchronous data transfer device. The data rate is fixed at one twelfth of the crystal's operating frequency, i.e. 1 MBaud for a 12 MHz crystal. The Rxd pin is used for bidirectional data transfer and the Txd Pin is used as a clock output for synchronization. Although this mode of operation clearly has applications - e.g. I/O expansion using shifts registers, point-to-point interprocessor communications, high speed datalink implementation etc., etc. - it is not directly relevant to the work outlined in this thesis.

Mode 1

In this mode the serial port functions as a conventional eight-bit UART, with the Rxd, Txd pins being used for full-duplex data transfer. The receiver/transmitter ready/interrupt flags (RI, TI) are located in bits zero and one of SCON, and are used in conjunction with the serial port buffer (SBUF) to receive/transmit data. To utilize this mode of operation the user must configure Timer 1 as an auto-reload timer(i.e. set TMOD the timer mode register, equal to hexadecimal twenty) and load the appropriate constant into the high-order byte (TH1) of the timer's register. Baud rate generation commences when the Timer Run control bit (TR1) is set in TCON, the Timer Control Register. Enclosed with this Appendix is a listing of the Assembly language program used for the soak-testing of RF Boards - serial port dependent interfaces. This listing shows the reader how the serial port may be configured to transmit at 9600 Baud.

Also enclosed with this Appendix is a listing of the HP86 program used to generate lists of timer constants/baud rates (Appendix 1). The baud rate generated is a function of the TH1 constant and the oscillator frequency, and is given by the equation

$$\text{Baud Rate} = -(\text{Oscillator frequency})/(12 \times 32 \times R)$$

where R is the (negative) reload constant.

Thus, the highest baud rates are obtained using the reload constants with the smallest modulii - e.g. a reload constant of minus one gives the highest baud rate (9600 baud for the 3.6864 MHz) as it causes timer register overflow on the initial incrementation. The HP86 program lists the reload constants as negative two's complement integers; the constant minus one, for example, would be loaded into Th1 as hexadecimal FF.

Mode 2

In this mode the port is configured as a 9-bit UART operating at a fixed baud rate given by

$$\text{Baud rate} = (\text{Oscillator frequency})/64$$

At 12 MHz the calculated baud rate is 187.5 KB, which is fast enough for efficient asynchronous interprocessor communications - e.g. for datalinks or multi-CPU configurations using bussed serial ports. Because the baud rate is fixed, Timer 1 is available to the user for general-purpose use. The ninth bit of the serial frame is used as part of a useful master/slave protocol implemented by hardware within the 8751. The protocol has much to recommend it, mainly because it allows the user to implement a simple processor-to-processor communications system with a minimised software overhead. A full description (which is beyond the scope of this thesis) of the master/slave protocol can be found in Reference 1.

Mode 3

Mode 3 operation is similar to mode 1 operation in that Timer 1 is again utilized as a baud rate generator, but a nine-bit frame is employed, as for mode 2. This mode of operation has much to recommend it for general-purpose multi-processor communications, although data rates are of course limited compared with, say, mode 2.

```
;*************************************************************************
;
;                PROGRAM FOR SOAK TEST OF RF BOARDS
;
;THIS PROGRAM TRANSMITS THE CHARACTER 55H (U) VIA THE SERIAL PORT
;HARDWARE REQUIRED:
;       RACK
;       POWER (WATCHDOG) CARD
;       2 X RF CARDS
;       CPU CARD
;       HP85/86 WITH OPTION 1 SERIAL I/F CONFIG'D FOR TTL
;       ADAPTOR (FEM. 25-WAY DEE TO BOARD PLUG)
;*************************************************************************
;
;               DEFINITIONS
;
TX      EQU     42H             ;TRANSMITTER ENABLE
AUTO    EQU     20H             ;AUTO RELOAD TIMER
B9600   EQU     0FFH            ;9600 BAUD
ENABLE  EQU     0FFH            ;ENABLE (FOR P3)
U       EQU     55H             ;01010101 FOR BEST TEST
DTIME   EQU     0FH             ;CONSTANT FOR DELAY TIME
ORG     100H
;
;       CONFIG SER. PT. FOR 9600 BAUD
;
        MOV     P3,£ENABLE      ;WRITE ONE'S
        MOV     SCON,£TX        ;ENABLE TX
        MOV     TMOD,£AUTO      ;AUTO-RELOAD MODE
        MOV     TH1,£B9600      ;9600 BAUD
;
;START BAUD RATE GEN.
;
        SETB    TR1
;
;LOAD CHARACTER 'U' INTO ACC.
;
        MOV     A,£U
SELF:   JNB     TI,SELF         ;OUTPUT CHAR.
        CLR     TI
        MOV     SBUF,A
DELAY:  MOV     R0,£DTIME       ;WAIT - (HP85 IS SLOW)
DEL:    DJNZ    R0,DEL
        DJNZ    R1,DELAY
        JMP     SELF            ;NEXT CHAR.
        END
```

```
10 ! ***************************************************
20 ! PROGRAM TO LIST BAUD RATES FOR 8751 MICROPROCESSOR
30 ! AUTHOR P.GRAY SEPT. 1984
40 ! USES HP86 WITH P.P. AND A.P. ROMS
50 ! AND 82905B PRINTER
60 ! ***************************************************
70 !
80 ! DEFINE 82905B PRINTER AS OUTPUT DEVICE
90 PRINTER IS 702
100 ! REQUEST XTAL. FREQUENCY
110 LINPUT "XTAL FREQ. IN MHz?",X$
120 ! CONVERT TO Hz
130 X=VAL (X$)*1000000
140 ! PRINT HEADER
150 PRINT "    CONST.   BAUD     CONST.   BAUD     CONST.   BAUD     CONST.   BA
UD "
160 PRINT
170 ! PRINT IN 8 COLUMN FORMAT ON A4
180 FOR I=1 TO 64
190 ! CALULATE RELOAD CONSTANTS
200 R0=-I
210 R1=-(I+64)
220 R2=-(I+128)
230 R3=-(I+192)
240 ! CALCULATE BAUD RATES
250 B0=X/(12*32*I)
260 B1=X/(12*32*(I+64))
270 B2=X/(12*32*(I+128))
280 B3=X/(12*32*(I+192))
290 PRINT USING 300 ; R0,B0,R1,B1,R2,B2,R3,B3
300 IMAGE 4X,S3D,4X,5D,5X,S3D,4X,5D,5X,S3D,4X,5D,5X,S3D,4X,5D
310 NEXT I
320 ! PRINT CAPTION
330 PRINT
340 PRINT
350 PRINT "           TABLE OF RELOAD CONSTANTS/BAUD RATES FOR ";X$;"MHz XTAL"
360 ! UNDERSCORE WITH ASTERISKS
370 FOR I=1 TO 60+LEN (X$)
380 IF I<12 THEN PRINT " ";
390 IF I>11 THEN PRINT "*";
400 NEXT I
410 PRINT
420 ! RELEASE PRINTER
430 PRINTER IS 1
440 END
```

Appendix 4 - HPIB Overview

This brief overview of HPIB (Hewlett-Packard Interface Bus) is included for the reader who is unfamiliar with HPIB and requires an explanation of some of the technical terms used in Chapter 5. In 1978 the protocol was formally adopted by ANSI/IEEE as the Standard IEEE-488, a standard which is also referred to as GPIB (General Purpose Interface Bus). The protocol is referenced throughout this thesis as HPIB rather than IEEE-488 or GPIB; this is appropriate, as the Company does use exclusively HP computers, and therefore the equipment will be interfacing to HP's implementation of IEEE-488 rather than any other.

HPIB is a parallel bus system originally developed by HP to interlink laboratory equipment; a typical modern-day example of it's application might be, say, a system in which a desk-top computer is used to monitor the output of a complex piece of chemical analysis equipment and log the results on a dot-matrix printer in the form of a report. The three pieces of equipment - computer, analyser and printer would be linked together by a series of cables in a simple 'daisy-chain' manner. HPIB cables feature convenient stacking connectors to make the daisy-chaining a fool-proof operation. In this example the computer would (almost certainly) assume the role of the HPIB Controller (see below), but this would probably be a function performed transparent to the user. Despite its depth of specification, the beauty of HPIB is its simplicity of use compared with, say, a relatively crude protocol such as RS232, which is notorious for causing connector pin-out misunderstandings. This simplicity of application possibly accounts for HPIB's lasting popularity in Industry.

Figure A4-1 offers an illustration of the HPIB concept . An HP86 desk-top computer is shown as the Bus Controller which controls the transfer of data between the other devices which may be either Talkers,

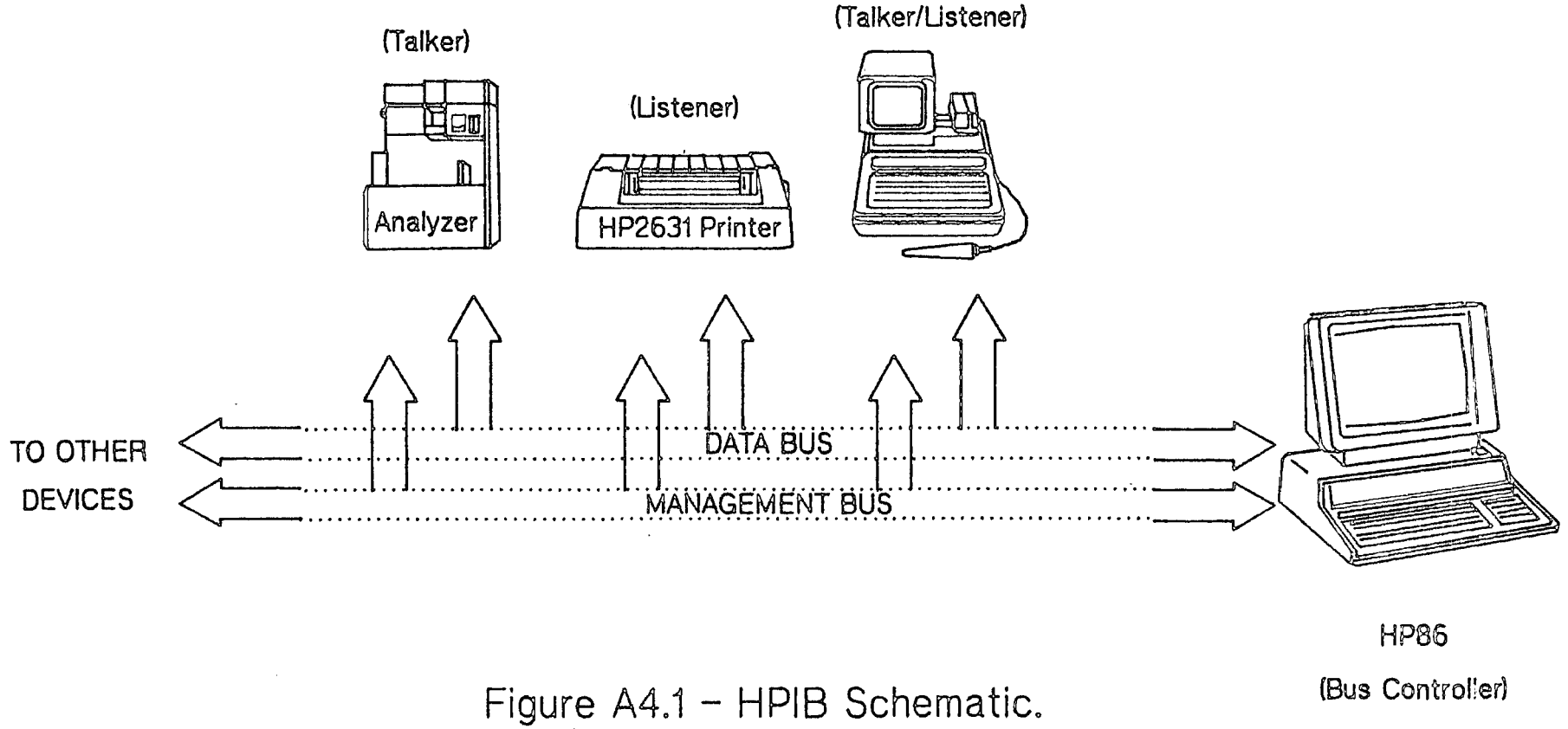


Figure A4.1 - HPIB Schematic.

Listeners, or both (Talker/Listeners). A typical Talker device (shown) might be the analyser quoted as an example previously. A Talker can transfer data via the Bus (HPIB) to one or more Listeners - in this case, perhaps the 2361 printer (illustrated) or, even the HP86 (a device which acts as a Bus Controller may also, additionally, perform Talker/Listener functions; this applies to the HP86 with its built-in HPIB interface). Another example of a Talker/Listener device (shown) might be a data capture terminal; such a device could have its own keyboard and display in addition to a slot reader and light-pen wand (devices for reading bar codes). Bar codes are used increasingly, for example, in the Pharmaceutical Industry for identifying batches of bulk material. The terminal would, typically, transfer data before logging it on the printer (the HP86 now acting as a Talker while the printer Listens) and the terminal's screen (the terminal is now a Listener).

To achieve this flexibility the HPIB Bus is organized into two sets of eight lines (each) known as the 'Data Bus' and the 'Management Bus'. The remaining lines of the HPIB (there are 24 in total) are earth lines associated with screening or grounds for signal logic. The Data Bus is simply eight lines (DIO1 - DIO8) used, as the name implies, to transfer data bytes (and also command bytes) between devices in a bit-parallel byte-serial manner. The Management Bus consists of eight control lines used for handshaking or other purposes.

The single most important control line is ATN which is asserted (HPIB uses negative-true logic) by the controlling device when it wishes to transfer commmand bytes onto the Data Bus. Talkers and Listeners respond to ATN by ceasing data transfer and responding to subsequent commands bytes from the controller.

A typical sequence of events is:

ATN = LOW

UNT (Untalk)

UNL (Unlisten)

LISTEN 1

TALK 2

ATN = HIGH

The UNT, UNL commands are used to remove current Talkers and Listeners from the Data Bus. The result of the sequence above is to cause data to be transmitted from device Two to device One. There may be up to sixteen devices on the Bus, each with its own unique address.

When a Listener is ready to accept data it indicates this to the active Talker via the handshake line NRFD (Not Ready For Data). (All Listeners will take NRFD HIGH when they are ready; thus the slowest device governs the speed of data exchange). On seeing NRFD HIGH the Talker takes DAV (Data Valid) LOW to indicate that a data byte is on the Data Bus. (Only one Talker may be active on the HPIB, so this signal condition is of immediate effect). The Listener takes the data byte and sets NDAC high (Not Data Accepted) - again the slowest Listener determines when the signal condition is established. The Talking device sets DAV high on seeing NDAC HIGH , and the Listeners respond by taking NDAC back to the LOW state. This completes the handshake sequence.

The NRFD/DAV/NDAC triple-wire handshake ensures that the data is reliably transferred at the speed of the slowest Listener. Other single-wire control lines not covered so far are SRQ, REN, IFC, and EOI.

SRQ (Service Request) is an active LOW signal that may be asserted by any device to request attention from the controller - which then polls the devices (see below) to determine the source of the request. REN (Remote

Enable) is a special signal that is used by the controller to switch all the Listeners into remote mode, while IFC is used as a general reset signal for Bus devices. EOI (End Or Identify) is a signal used by the controller to elicit a parallel poll response (see later).

For simplicity, the signal wires have been grouped into two data buses as shown in Figure A4.1. A more strict convention for the signal groupings is the following:

(i) An eight-line 'Data Bus' (DIO1 - DIO8).
(ii) A 3-line 'Data Byte Transfer Control Bus' (NRFD, DAV, NDAC).
(iii) A 5-line 'I/F Management Bus' (ATN, SRQ, REN, IFC, EOI

The Bus adapter chip (96LS488) used in the HPIB Board design responds to REN with a LOW output on the R/L pin; the chip also responds to both serial and parallel poll. Serial poll is performed by the controller sending the SPE (Serial Poll Enable) command on the data Bus followed by the talk address of each device in turn. In the implementation chosen by the author, the RQS (Requested Service Output) will drive DIO7 LOW if an SRQ request has been made via the interface.

The IST pin of the 96LS488 is the parallel poll input pin; the chip responds to a parallel poll request (initiated by the controller taking EOI and ATN LOW simultaneously) by comparing a defined Data Bus line with the current IST logic level (see Ref. 14), and driving another (defined) Bus line LOW. For the prototype HPIB Board design, the IST pin has been wired LOW, as certain facilities have been dropped to improve the track layout - the board is heavily populated. It is true to say that many HPIB devices that are manufactured are designed to respond to only a subset of the HPIB commands, and it is the author's experience that parallel poll is a feature that is rarely implemented. There is little point in implementing the feature if the majority of the devices in common use (printers, balances, etc.) do not support parallel poll.

Appendix 5 - Features of the 8748

The 8748 single-chip microprocessor (as used on the BCD Board - see Chapter 5) is a member of the MCS-48 family (Refs. 31,32), the precursor to the MCS-51 family. The major features of the 8748 are:

* Eight-bit CPU
* Three eight-bit I/O ports
* 1K Bytes of EPROM
* 64 Bytes of RAM
* Eight-bit timer
* Multiplexed Twelve-bit Address/Data Bus

Although, if compared with the 8751's facilities (Chapter 2), these features appear to make the 8748 to be the 'poor relation', the 8748 has several useful 'extras' not implemented by Intel within the 8751's hardware. For example, the user may debug small programs held in the 8748's EPROM by using the single step pin (pin 5); only a small amount of external hardware is needed to provide the function (see Reference 31 Figure 2-13). Two test inputs are provided (but only one interrupt pin) which may be directly tested by dedicated instructions (JT0, JNT0, JT1, JNT1). Probably the biggest 'extra' the 8748 offers, however, is its ability to expand its I/O using 8243 port expander chips. A timing signal (PROG) and 4 bits of port 2 are used for this purpose. The 8748's instruction set contains instructions allowing the programmer to address the 8243's as though they were an integral part of the 8748. By bussing several 8243's and using spare 8748 (port 2) pins to drive their chip selects, a very powerful I/O structure can be quickly designed. Thus the 8748 is an ideal choice of microprocessor for the BCD Board application described in Chapter 5.

It is beyond the scope of this thesis to include full details of the 8748's instruction set, which is fully described in Reference 31. This reference, and Reference 32, offers much application detail for the 8748.

Appendix 6 - BCD Board (Testing/Firmware)

As indicated in Chapter 5, testing of firmware for the 8748 was strictly limited by the lack of development facilities (the PDS system used for testing the 8751 supported neither an MCS-48 family assembler, nor an 8748 emulator). The testing, therefore, consisted of writing small blocks of 8748 machine code to check sections of the BCD board circuitry. This testing was to some extent trivial, as the author has had previous experience of the MCS-48 family in similar work unrelated to this project; also, the BCD interface design is not complex.

Figure A6.1 shows a simple schematic arrangement for data exchange between the BCD Board (8748) and the CPU Board (8751). The 8748 firmware accesses data from the 8243 I/O expanders (see Figure A6.2), writes the data to the data latch, and sets the status to 'ready' within the status latch. The 8751 then accepts the data and status bytes (reading the status register causes an 8748 interrupt, which indicates that the status has been read), stores the valid data byte and loops while testing the status register until the next byte is valid . The sequence then repeats. The simple sequence shown in Figure A6.1 will be replaced by more sophisticated software routines written by software specialists within the Company's Computer Service Department.

The BCD Board has the facility to interrupt the 8751 to initiate serial poll; thus the Board will be a member of the set of boards capable of taking part in the interrupt/status handshake system.

Testing of the 8748-8243 circuitry was achieved using switches (switching to ground with an open-circuit pull-up) to simulate the BCD data lines from a BCD device. Further switches were used to switch the two flag lines, and the status of the control line was monitored with a logic probe.

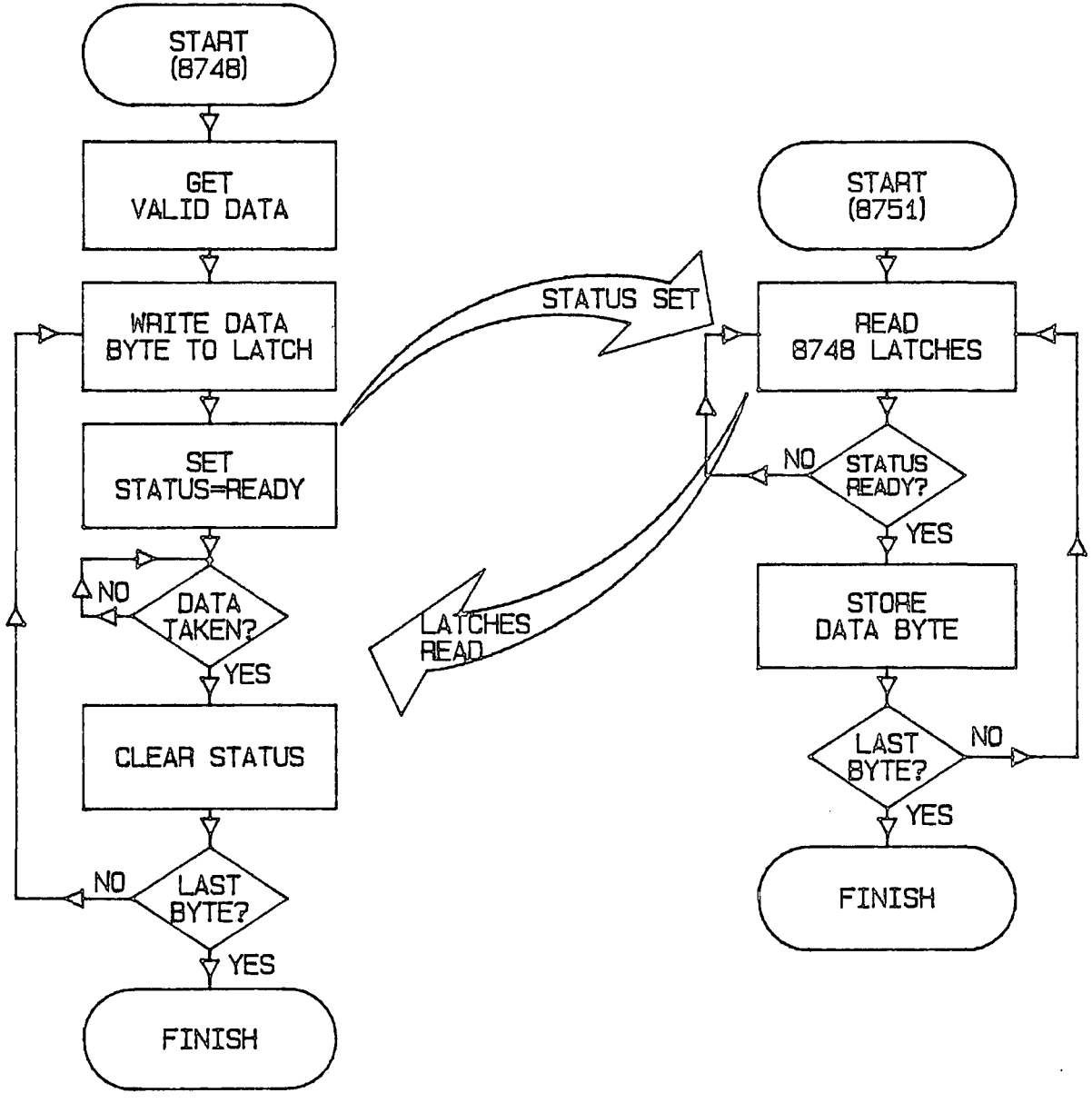


Figure A6.1 - Schematic of 8748/8751 Interaction

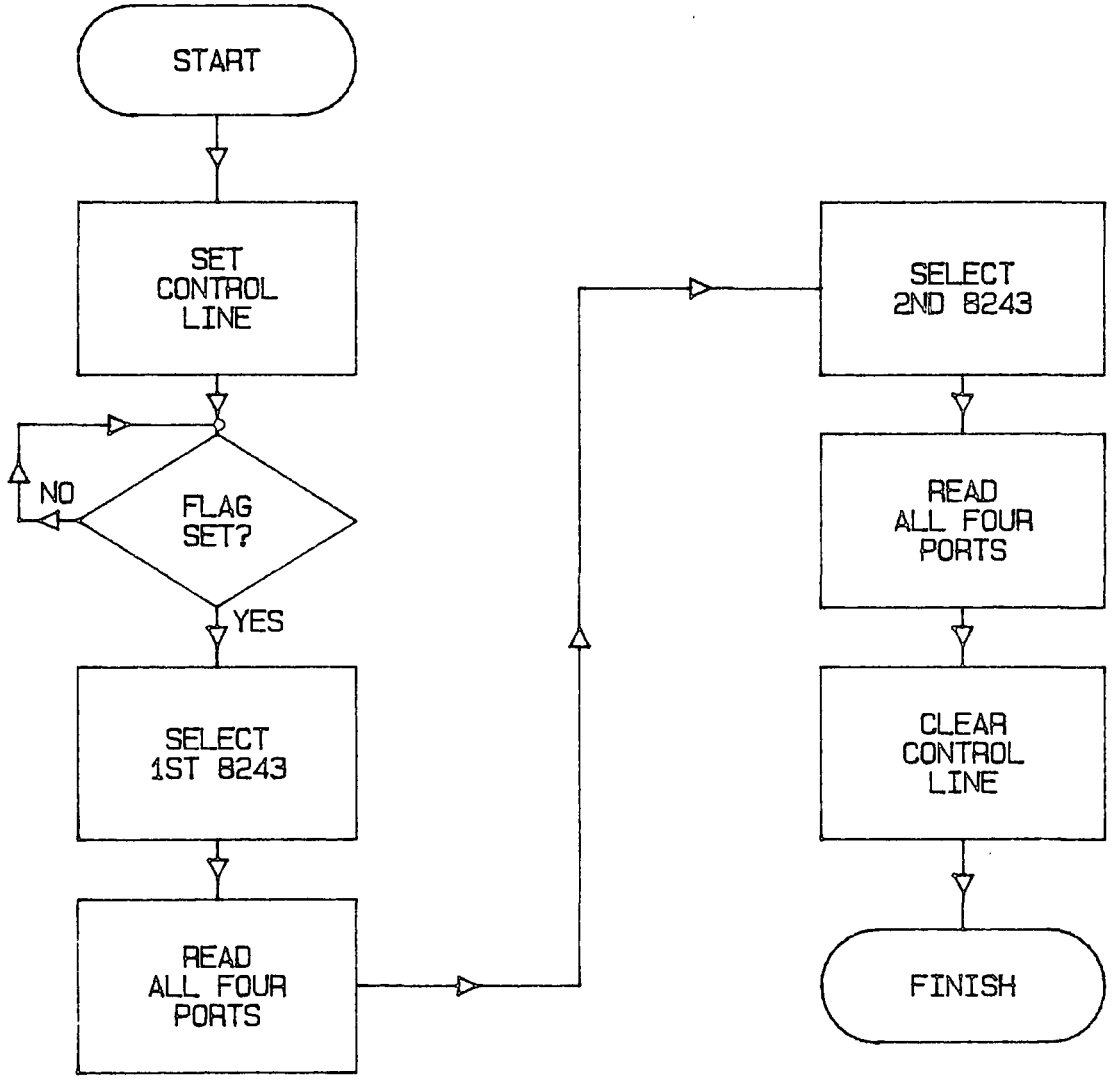


Figure A6.2 - Schematic of 8748/8243 Firmware.

A small modifiable machine-code routine was written to read data from adjacent pairs of selected expander ports (specified by the state of the flag lines 00, 01, 10, 11) and latch the data byte to the Bus Port. A further machine code routine was written to latch bytes into the two '574 latches from the 8748, and the PDS was used to check that the 8751 could read the data bytes via the backplane lines, thus the confirming the validity of the address decode logic on the BCD Board.

Appendix 7 - HPIB - RS232 Converter

In Chapter 5 the testing of the HPIB Board and the UART Board were described in detail, while Chapter 6 gave details of the Serial Board check-out. During later stages of the tests, the various boards of the system were checked in combination - some of these combinations suggesting an immediate use as working systems for specific applications.

An example of of a potentially useful combination of boards is the set:

Board	Use
CPU	Control
Watchdog	Power input/monitor
HPIB	Data Input
Serial/UART	Data Output

which, in the example shown in Figure A7.1, can be used to create a basic HPIB to RS232 converter.

The convertor shown schematically in the Figure A7.1 flow diagram can work with either a Serial or UART board as the output interface. The initialization of the serial device (see flowchart) in the case of the Serial Board consists of setting up the 8751's on-board UART registers as described in Appendix 3 and elsewhere, while the UART Board (if used) is set up by initializing the 8251 UART as discussed in Appendix 9. The HPIB interface requires a dummy read to set up its status registers correctly, as indicated in Chapter 5.

Once both devices are set up, the HPIB status register is examined to see if a byte has been received; the interface will not accept another byte from the HPIB until the 8751 takes the current one. As soon as the 8751 has taken the byte it sets the output handshake (RTS) line to the remote serial device acting as the data sink. This tells the remote receiver that a byte is pending, and the receiver responds by taking the input handshake

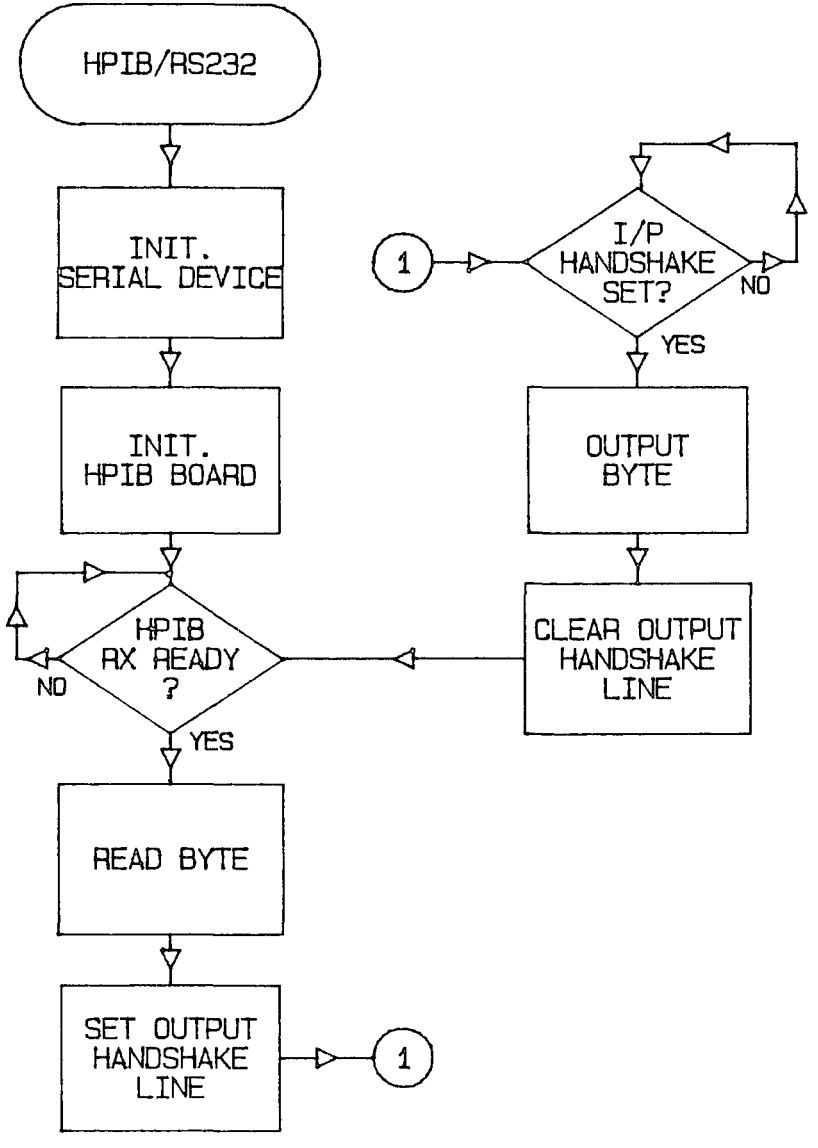


Figure A7.1 - Schematic of HPIB to RS232 Converter

line (CTS) high, prompting the 8751 to output a character via its serial channel. Once the character has been transmitted, the 8751 clears the output handshake latch and returns to the commencement of the main loop and awaits the arrival of the next character from the (HPIB device) data source. This is in theory, the complete firmware requirement for a simple HPIB to RS232 converter. In practice, the program probably would be enhanced in a number of ways - e.g. on-board buffering (RAM Board) would give the unit a 'spooler' feature which would enable the transmitting device to dump blocks on data into the unit, without being limited by the speed of the (slow) serial device. Also, there is no reason why the firmware should not be written in a general purpose way to support bidirectional data transfer.

Appendix 8 - The 8052-BASIC CPU

This Appendix gives further details of the hardware configuration required to support the 8052-BASIC chip described in Chapter 8. Figure A8.1 shows a typical minimum configuration for a system supporting RAM and EPROM.

Memory Configuration

The BASIC chip is programmed to expect a specific configuration of the external memory; the required partitioning is as follows:

0-32K	RAM	Memory (Controlled by RD and WR)
0-32K	EPROM	" (" " " PSEN)
32K-64K	RAM/EPROM	" (" " " [PSEN ANDed with RD] and WR)

The system allows for the programming of EPROM's resident in the upper address range. Because PSEN and RD are ANDed during access to this memory area, both BASIC and assembler programs stored in these locations can be executed.

During EPROM programming the system uses the ALE DISABLE (P1.3) line to override ALE, thus allowing the lower address byte to be held in the ('573) latch for a prolonged period; by using this technique the CPU is able to use port 0 as an I/O port (this port is open-drain in structure - it is necessary to use external 10k pull-ups) and use the port latches to hold data bytes for EPROM programming data. (A similar double-latch system is utilized in the BCD Board design). The programming pulses (PULSE AND ENABLE) provided to control the programming are TTL level signals, hence the user must provide the necessary level shifters to increase the voltages to those specified by the EPROM manufacturer. The system supports a

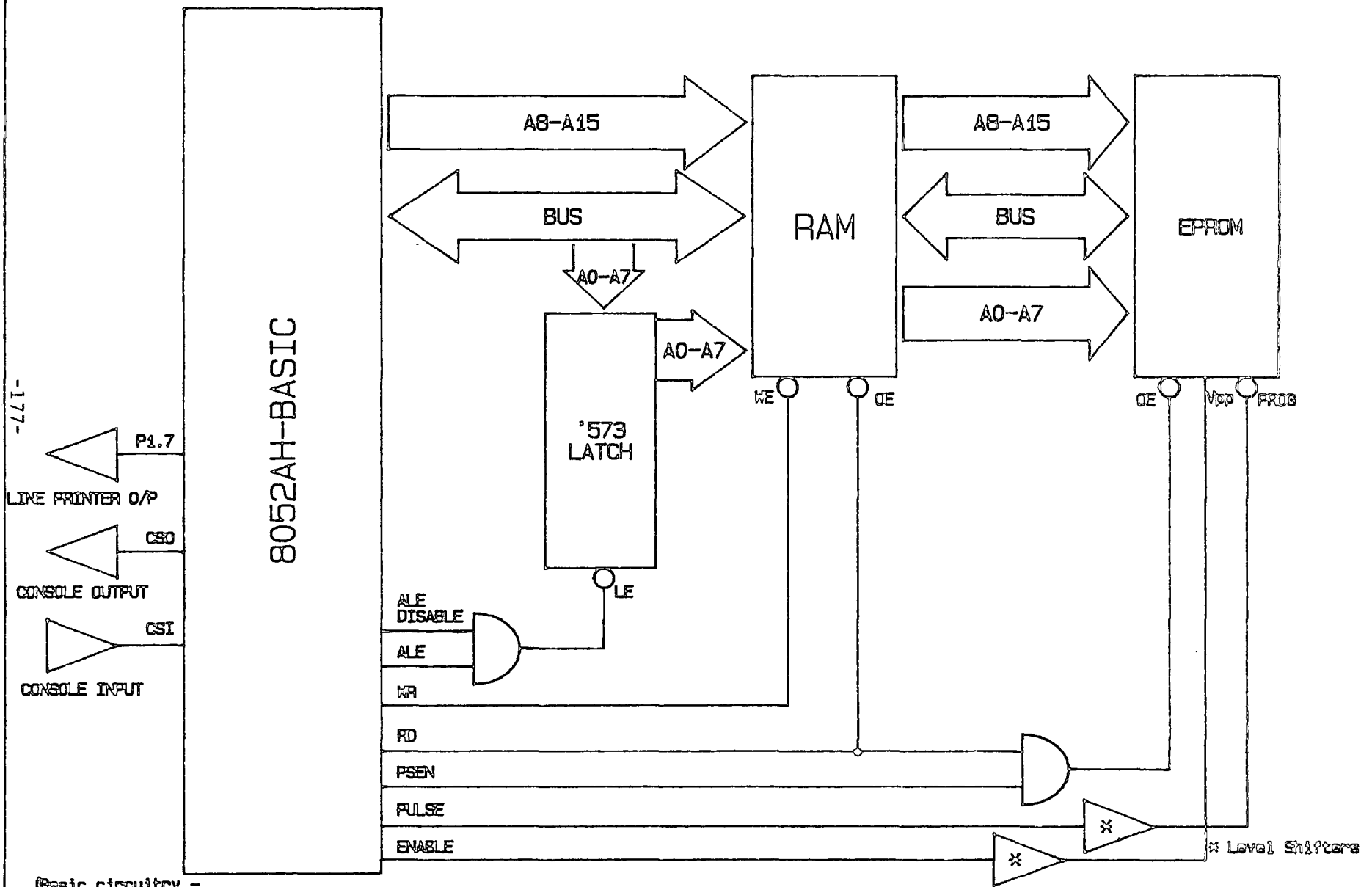


Fig A8.1 - Schematic of a Typical BASIC-52 System.

conventional programming algorithm (via the 'PROG' command) and also Intel's fast programming algorithm (FPROG). To use the latter it is mandatory to increase Vcc for the EPROM to 6v throughout the programming interval.

The MCS BASIC-52 system has a unique EPROM filing system for storing BASIC programs. Issuing the 'PROG' command, for example, causes the currently selected BASIC program, resident in RAM, to be stored in an EPROM file. The system maintains a file index automatically, allowing the user to recall programs back into RAM memory at will, via the 'XFER' command. Alternatively, the programs may be executed directly from EPROM memory. 'RAM' and 'ROM' commands are provided to allow the user to switch between the two memory areas.

It should be born in mind that 8K of the usable object code memory space is already occupied by the 8052's on-board ROM (used to implement the BASIC interpreter and other elements of the system). Also, the system needs 1K of external RAM for system variables, so the first 1K of external data memory is also unavailable to the user.

Statements

Reference 20 lists (Section 1.4) commands/statements/operators that are supported by BASIC-52. Typical commands that may be executed via the console are RUN, LIST, LIST=, etc. There are fourteen defined commands, forty-four statements such as DO-WHILE, DO-UNTIL, FOR-TO-STEP, IF-THEN-ELSE, etc.,etc., and forty-five operators. The operators available include the standard set of mathematical instructions associated with BASIC, as well as many other useful and powerful operators unique to BASIC-52.

Assembly Language Interface

MCS BASIC-52 allows the user to interface to an object program (held in EPROM) by means of the 'CALL' instruction. 'PUSH' and 'POP' instructions allow programs, subroutines or assembly language routines to exchange parameters via the stack. For example,

```
PUSH A,B
CALL 5000H
```

calls the assembly language subroutine at location 5000H. The parameters A and B are passed via the argument stack; assembly language programs can retrieve the parameters with calls to defined BASIC subroutines. These same subroutines (with specific parameter values) allow the user to make use of the BASIC facilities from assembly language programs. The user can thus write a program in assembly language - which is faster running than interpreted BASIC - and make use of the on-chip 'library' of powerful functions when required. Full details of the functions available are given in Reference 20.

Port 1 Functions

BASIC-52 provides the 'PORT1' operator (see Chapter 8) to read/write data to port 1. All eight port lines have special functions associated with them, as listed below:

Port 1 Pin	Function	Meaning
0	T2	Timer 2 Trigger Input
1	T2EX	Timer 2 Input
2	PWM O/P	Pulse Width Mod. O/P
3	ALE DISABLE	(used during EPROMing)
4	PROGRAM PULSE	(" " ")
5	PROGRAM ENABLE	(" " ")
6	DMA ACK	Fake DMA Acknowledge
7	L.P. O/P	Line Printer Output

The timer functions (P1.0, P1.1)are standard 8032/8052 functions, and the reader is referred to the respective data sheets. The PWM (Pulse Width Modulate) signal, whose use is described in Chapter 8, is output on pin P1.2. The ALE DISABLE, PROGRAM PULSE and PROGRAM ENABLE are all EPROM-related signals, as discussed earlier in this Appendix. The DMA facilities that use pin P1.6 as an acknowledge output are reviewed in detail below; the L.P. output (pin 1.7) has been mentioned already in Chapter 8 - it is a TTL compatible output that is used as a channel for listings and other outputs invoked by the LIST= and PRINT= commands. A statement (BAUD) is provided to allow the user to tailor the transmission rate to match the speed of the printer employed.

Fake DMA

Although the MCS-51 family architecture does not support DMA with dedicated on-chip hardware , the BASIC variant offers a type of 'DMA' known as 'Fake DMA', which is implemented using the INT0 pin as a DMA Request input and P1.6 as a DMA Acknowledge signal. To use the Fake DMA facilities the user must:

(i) Provide tri-state buffers for port 0 and port 2 and also a tri-state address latch, and control the tri-stating of these chips with the DMA Acknowledge signal.

(ii) Using BASIC statements, enable Fake DMA by setting three bits associated with the facility (one of these bits is located in the 8052's internal memory, the other two are held in the Special Function Register).

Although at first sight the functioning of the Fake DMA system appears similar to a conventional DMA system, operation is much slower, as the system is interrupt driven. During the operation of the DMA access, the

8052 idles, testing the state of INT0. The full Fake DMA operation sequence is:

(Initially, DMA Request (INT0) and DMA ACK (P1.6) are HIGH).

1. The requesting device takes INT0 LOW and waits for DMA ACK to go LOW.

2. The 8052 interrupts its processing, takes DMA ACK LOW and waits for INT0 to return HIGH.

3. The requesting device performs the memory access and then takes INT0 HIGH. (The device waits for DMA ACK to return HIGH before making further DMA requests).

4. The 8052 sees INT0 return HIGH, takes DMA ACK HIGH, and continues normal execution.

Obviously, this method of memory access is inefficient compared with 'cycle stealing' systems, but is a useful feature for those users who require a DMA-like memory access system, and are not constrained by high-speed requirements.

Appendix 9 - Serial/UART Board Functions & Options

Current Loop Configurations

The active/passive current loop configurations shown in Figure A9.1 apply to both the Serial and UART Boards, as the current loop circuitry is identical in design for both boards. By means of the eight-way DIL switch that is incorporated in the prototype designs, the system user may select active/passive combinations for both the receiver and transmitter circuits. For example, the passive options shown in Figure A9.1 may be selected may by closing switches one through four and opening switches five through eight, while the active options may be configured by utilizing the complementary switch settings.

The instruments with current loop interfaces that the Company have used in the past have, although possessing full duplex facilities, not been used for simultaneous bidirectional data transfers. Hence the prototype units described in this thesis are designed to support a three-wire cabling system that is frequently used within the Company to connect to small industrial balances. This is not good practice for long cable runs which should feature twin twisted-pair cables with separate signal returns; it would be better therefore, on production boards, if the passive loop options featured independent return lines, to minimize cross-talk and offer improved noise immunity.

UART Board - Configuration

The 8251A USART is a software-configured device. Two sections of the USART, the synchronous and asynchronous sections, can be thought of as two independent devices sharing the same package. As there is no foreseen requirement for the synchronous facilities the part offers, this option has been ignored in the development work. Obviously, should the need arise in

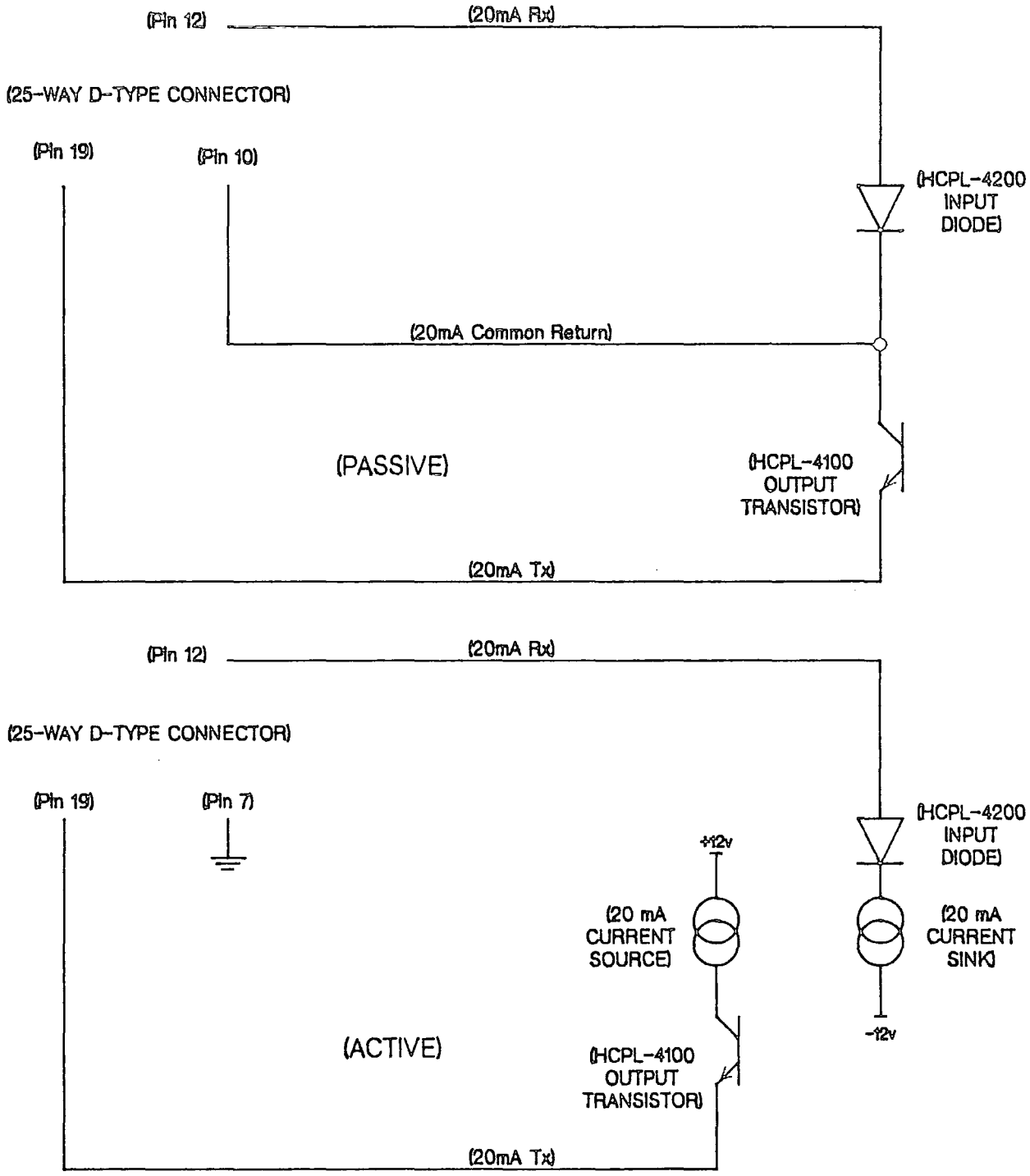


Figure A9.1 - Current Loop Configurations.

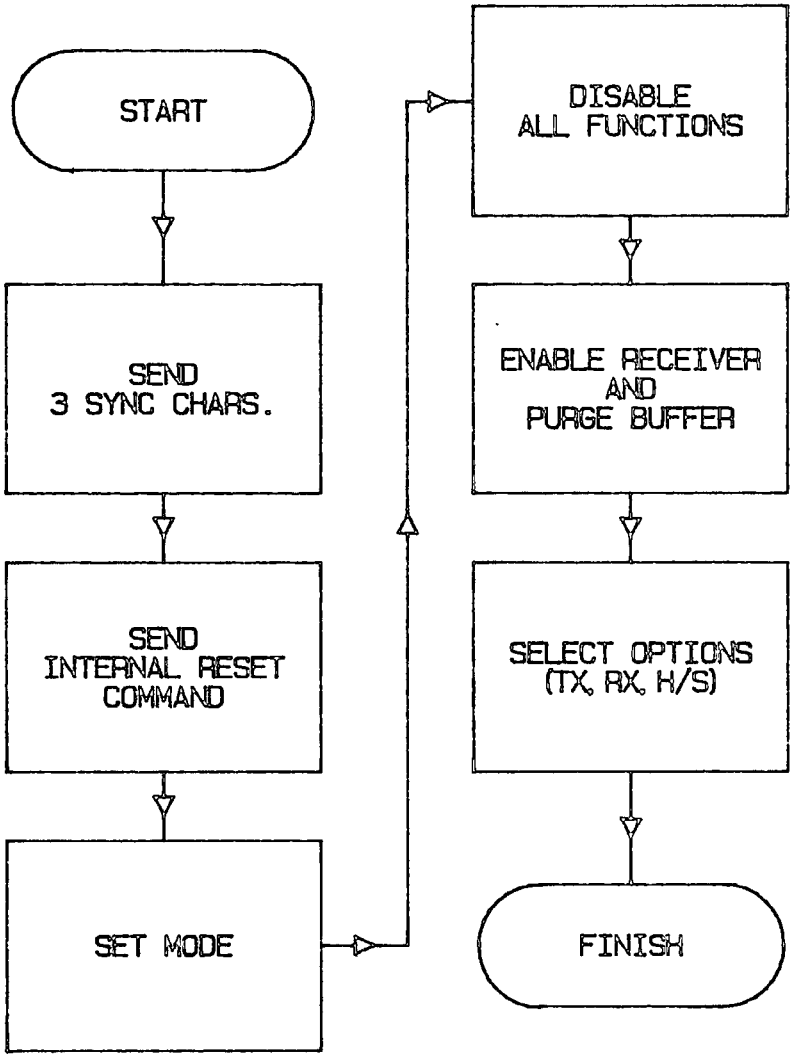


Figure A9.2 - UART Initialisation Sequence.

the future, then the system could easily support synchronous devices with the minimum of circuit modification.

To use the USART correctly, it must be initialized in accordance with the manufacturer's specification; the reader is referred for full details to the 8251A data sheets and to Reference 5, which contains much useful application data. After correct initialization, the user passes/accepts data to/from the USART via its internal data buffers. Internal registers are also provided for command/status bytes, which the 8751 accesses via the data bus as described in Chapter 5.

Figure A9.2 shows schematically a typical system for initializing the USART, which is a simplification of the diagrams/notes contained in the Intel application data. The initialization scheme starts with the transmission of three dummy SYNC characters (binary zero), which is a necessary precursor to issuing an internal reset command (hexadecimal forty) to the USART, in order to put the chip into a known state. The dummy SYNC characters are required as a result of the USART's dual function, explained earlier. (The reader should note that delays are needed in the initialization firmware between writes to the USART in order to meet the various recovery/response times specified in the manufacturer's data sheets).

After the reset state has been successfully achieved, the user writes the mode byte to the USART, which selects the various options - parity, character length, etc.. After these mandatory operations, the user may perform a number of different commands according to his specific requirements. Typically, for a full duplex data transfer, the user will inhibit all options by writing zeros into the command register, before purging the 8251's input buffer - as required by the manufacturer's specifications, as the 8251 can supply erroneous data upon start-up. (This

is a similar condition to the one encountered during the HPIB interface initialization). The user can now safely set-up any required options by writing the appropriate command word to the 8251. The command word options, mode instruction and status register formats are all detailed in the 8251A data sheets to which the user is referred for further information.

Appendix 10 - Power Supplies

Several different power supplies were used during the course of the Project; the specifications of the various units are listed below.

Farnell D5-05A

This was the sub-miniature 500mA supply used for the Pilot Scheme. It was a mass produced economically priced unit, primarily intended for powering a digital panel meter. It had a regulated 5v output, and also an unregulated 12v output which proved useful in the Pilot Scheme for powering instrumentation external to the nodes.

Parameter	Specification
Mains Input	214-255v and 108-128v, 50-400Hz
O/P Voltages	5v +/- 0.2v (3-terminal regulator)
	12v +/- 4v (unregulated)
Load Regulation	Less than 50mV
Line Regulation	" "
Ripple	" " pk to pk

Some use was also made of the Farnell 6/500P sub-miniature supply (a single output supply with a superior specification to the D5-05), and the related 15/15/100P unit, which featured split supplies suitable for powering RS232 buffers, OP amps, etc. All units were enclosed in robust cases formed from extruded aluminium sections, and had clip on covers to protect the connections.

VMS 2000

This switch-mode power supply was used for powering the rack system and proved to be a most reliable and economical unit incorporating an integral mains filter.

Parameter	Specification
Total Maximum O/P	60 Watts
Input Voltage Range	175 to 265v AC @ 50/60Hz
	90 to 130v AC @ 50/60Hz
DC Output	+5v DC @ 6.0 Amps (adjustable)
	+12v DC @ 1.0 Amps
	-12v DC @ 1.0 Amps
	-5v DC @ 1.0 Amps
Line Regulation	Less than 0.25% over the input voltage range
Load Regulation	" " " for a 20% to 80% load change
Ripple	" " 50mV pk to pk

The above regulation figures apply to the 5v line only. The other supply lines are regulated by standard three terminal regulators (see the D5-05 figures). The unit was of open-frame construction, and was, naturally, considerably bulkier than the supplies used for the Pilot Scheme.

Appendix 11 - Details of RAM Memory Devices

This Appendix gives further information on the various RAM chips mentioned in Chapter 5, especially the 2186 IRAM used on the RAM Board.

The 2186 IRAM

The 2186 and 2187 pseudostatic (or 'integrated') RAM's are two 'byte-wide' memories introduced by Intel in an attempt to fill the gap between small-scale static RAM memories and complex DRAM systems. The 2186/2187 RAM's have the following features:

* On Chip Refresh
* External Refresh Option (2187)
* Microprocessor Handshake Option(2186)
* TTL Pin Compatibility

The 2187 is a synchronous device requiring more complex external circuitry than the 2186 asynchronous part, the latter being a natural choice for the RAM Board design, resembling a conventional SRAM in its external electrical characteristics. The 2186 has a 'Ready' output on pin one, which can, optionally, be used as a microprocessor handshake line to time memory accesses, thereby avoiding contention delays and achieving faster response. This feature was not, however, utilized in the RAM Board design.

The simple 'R-M-W' treatment used by way of illustration in Chapter 4 is not strictly correct for the 2186; it is certainly an accurate overview for the some modes of operation of DRAM memories such as the 2118, but the 2186 is a much more sophisticated chip embodying not only DRAM arrays but also refresh timers, counters, arbiters, address multiplexers, etc., etc.. The various modes of operation of the 2186 are reviewed briefly below, but for

a fuller treatment of the 2186 architecture the reader should consult References 10 and 11.

There are 'pulsed' and 'long' mode options for both read and write operations. Both operations are subject to contention delays, but otherwise the contention resolution processes are transparent to the device's functioning.

Pulsed Mode Read

If CE (Chip Enable - active LOW) returns HIGH before OE (Output Enable) returns HIGH, then a pulsed mode read occurs, provided the 2186 timing specifications are observed. This is a useful mode of operation for designs where the chip enable is not gated for, say, the duration of ALE or a similar long pulse. In the case of the RAM Board design, ALE is used as the main enable pulse, so this mode of operation is not relevant to the project application.

Long Mode Read

This mode applies to designs with a protracted CE pulse; the minimum time for CE to be held LOW after OE goes LOW is given in the 2186 data sheet. Failure to observe the timing specifications results in an FMC (see below).

Pulsed Mode Write

A pulse mode write occurs if CE returns high before WE (Write Enable), within the device's timing specifications.

Long mode Write

As for long mode read, this mode of operation is applicable to the RAM Board design in that it requires a protracted CE pulse; failure to observe the timing specifications results in an FMC.

False Memory Cycle

An FMC (False Memory Cycle) occurs when CE goes LOW without either WE or OE being active, or when a read/write violation takes place. This is in itself a valid mode of operation, causing a refresh of the row within the 2186 which is selected by the seven external row addresses.

Other Memory Chips

Both the Hitachi HM6264 and the Toshiba TC5564P 8Kx8 SRAM's were sampled for evaluation (see main text). Pin 26 on either device is used as an extra chip select (active HIGH), which is a useful facility for designers wishing to minimize select logic. The parts are available with a variety of options, with access times as fast as 100nS. Other memories of interest include the family of Hybrid RAM's from Electronic Designs Inc. (Ref. 30), whose family of components include static RAM-based parts from 32Kx8 bits upwards, and hybrid components - 80C31 (CMOS version of the 8031) CPU's combined with SRAM arrays, surface-mounted on 40-pin headers - obviating the need for further external RAM.