THE X25/ECONET INTERFACE

PURPOSE OF THE DESIGN

An interface that will enable a single BBC machine or an Econet Network to communicate with other systems via a telephone line and the apropriate modems. The "target system" need not be a BBC based system; indeed any machine that can handle the X25 protocol can be attached to the far end of the telephone line.

It is intended that the circuit should provide a relatively low cost interface between the public telephone network and Econet. The prototype was built to communicate with systems in two possible ways: -

a) Over the Public Telephone Network - Any remote device or system can be "dialled up" either manually or automatically, the ensuing communication being through the standard X25 protocol.

b) Over a leased line - For an annual rent B:T will supply all interface equipment (modems, protection circuits etc) and a unique set of frequencies for communication with a specific device or system, although the physical connection is through the public exchanges they guarantee that there will be no interference from the rest of the network. It is not thought that a leased line will ever be needed for this particular application - indeed, the very idea goes against the philosophy of "global data communications" which , after all, is what we should be aiming for.

The X25/Econet Interface can easily be expanded merely by adding extra communication devices to the board. Applications for expansion could be access to systems or devices through an internal telephone system to possibly increase the current physical range of Econet. This could also be used as a bridge between Econets or, providing that the appropriate gateways existed, a bridge between Econet and another LAN -Cambridge or Fast Ring for example.

See Fig 1 for possible uses of the gateway.

Since this board uses much the same circuitry at the "telephone end" as the PRESTEL ADAPTOR the two designs could eventually become incorporated to provide a general purpose access to the public network from a users terminal.

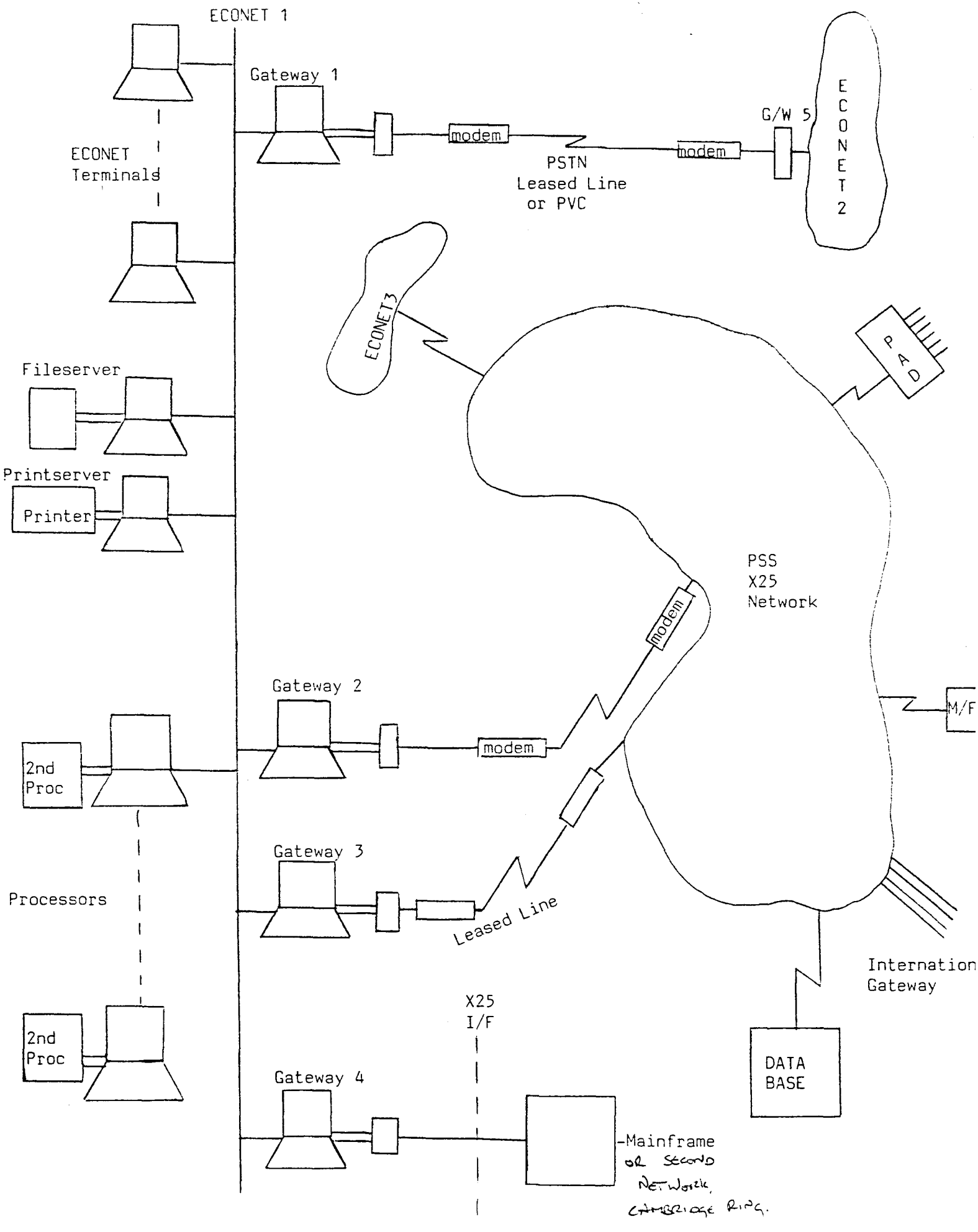


FIG. 1 Interconnection Possibilities

Elements of the design

As shown in the block diagram in Appendix A the interface consists of two Z80 processors working totally independantly of each other apart from when they are accessing the 32K of memory that is common to both of them.

The case of data being transmitted from the Econet to a target system is used to illustrate the functions of the various devices.

One of the Z80s, the "primary" processor reads the data as it is sent through the TUBE by the BBC whereupon it is converted to X25 format before being passed to the common area of memory. From there it is read by the "secondary" processor and passed to the SIO to be transmitted to the "target" device: On board is also a CTC that is used, not only for providing the timing signals for the SIO but can also be used to supply any timing or counting facilities that may be required by the software.

Obviously the reverse sequence of events holds for the case when characters are being received.

The way in which the secondary processor both reads and writes the data from the common area of memory is by DMA: It was decided to use a "second processor" for this purpose for the reason that the SIO is a fairly complex device and although the primary processor could have looked after it on its own it was felt that the overheads on the primary processor in looking after two,three or even four SIOs would be too much. Also Z80 DMA controllers are only single channel therefore two would have been needed for each SIO device. The AMID C controller was considered since it is a dual channel device but was not used on the grounds that both the software to operate it and the hardware necessary to interface it to a Z80 ware too complex to be realistic.

N:B It will be noticed from the circuit diagram that the primary processor operates at 6MHz, twice the speed of the secondary processor. This is necessary because the Z80 SIO cannot be operated at a speed of greater than 4MHz, whereas it was required that the primary processor be run as fast as possible since it has a lot of work to do in converting the very involved X25 protocol into that of Econet. In the final design it is hoped to operate the primary processor at 8MHz and the secondary at 4MHz.

Details of the circuit.

It is recommended that the reader refer to the circuit diagram for this section.

Memory access by the primary processor.

The memory for the primary side consists of 8K of EPROM residing between 0000h and 2000h: Fran here to the end of the memory space (FFFFh) is the RAM, the top 32K of which is shared with the secondary processor and is the medium through which the data is passed.

The RAM chips used are 8264 d-RAMs (8 off, 64K x 1): These are standard RAS and CAS operated chips and they are accesssed thus.

RAS is derived from the Z80 signal MREQ which becomes active during every memory access. MREQ is modified slightly so that there is a guaranteed precharge time of at least 100ns between successive MREQs. This is the function of IC17.

CAS is derived from sampling RAS with a delayed clock pulse, this is necessary to make sure that the CAS edge occurs after the WE edge since the data is latched into the memory chips on the falling edge of CAS: The generation of CAS is inhibited during a REFRESH cycle and also during a PROM access. Since CAS is generated from a delayed RAS there will be an appreciable delay between RAS and CAS returning to their high, inactive states. In some cases this caused problems because the time delay was too short between CAS going active and RAS going active again on the next cyle. It was for this reason that the generation of CAS was also inhibited when RAS was high - see gate 3, IC14.

Refreshing the memory chips is performed almost entirely by the Z80 processor itself which during every instruction goes through a refresh cycle where the refresh address is placed on the bus and a refresh signal is taken low. By enabling a RAS during this cycle means that the chips can be successfully refreshed as often as is necessary.

I/O space of the primary processor

By performing an I/O access at any address above and including 8000h will automatically access the TUBE. By performing an I/O access below this address means that the NMI line on the secondary processor will be kicked: This feature is included in case the protocol between the two processors needs such a facility.

The Second Processor

In terms of the actual devices that must be accessed by the second processor it is much more complex than the primary. The available address space is divided up into eight sections by an LS138 (a 3 to 8 line decoder) that has its three inputs connected to the top three address lines and its outputs connected to four devices: The devices accessed by the 138 are as shown below.

INPUT ODDE (A15, A14, A13)	OUTPUT SELECTED	DEVICE ACCESSED
0 0 0	0	EPROM
0 0 1	1	EPROM
0 1 0	2	RAM
0 1 1	3	CTC
1 0 0	4	SIO
1 0 1	5	NOT USED
1 1 0	6	NOT USED
1 1 1	7	NOT USED

Accessing the common memory from the secondary processor

Accessing this area of memory from the secondary processor is basically a very simple process made difficult by the fact the two processors are running at different speeds. In order to access the memory the secondary processor merely performs a memory access to a location above or including 8000h. When this occurs a BUSRQ is automatically sent to the primary processor and the secondary processor is put into a WAIT state until the corresponding BUSAK is received.

When the primary processor has issued its BUSAK several alternative signals are applied to the memory in order to effect the access. In order to run the memory at a slower speed the clock that helps generate RAS and CAS is changed from a 6MHz to a 3MHz. In addition to this RAS is brought artificially low and the combination of this and the slower clock produces a CAS that is a lot later than under normal circumstances: This is to accommodate the much later arrival of WE. In

addition to the memory access signals that are made available during BUSAK to address and data busses are latched in from the secondary processor via two LS244s and one LS245. The LS245 tends to get rather warm. This is because the "direction" pin is attached to the RD line of the secondary processor and hence is being activated many times during the course of "normal" operations: It is being operated within its tolerance however and should be able to stand this strain.

The SIO

The Z80 SIO is a highly versatile serial I/O device that can be used merely as a USART or, at the other end of the scale, a device capable of dealing with X25 protocol with automatic header stripping, zero insertion and CRC checking. It is obviously in this mode that the device will be used.

The SIO is not, contrary to popular belief, difficult to program but its high degree of versatility means that there is a fair amount to do to define its functional personality: Therefore the work lies in the amount of initialisation necessary as opposed to the actual complexity of it: Provided the user knows exactly what he wants of his SIO there should be no problem: Defining the operation of the SIO is performed by writing to eight control registers and thereafter its operation is monitored by reading its three read registers: There is a comprehensive range of error flags thus enabling the software to deduce exactly what has gone wrong in such an eventuality.

The data can be written to or read from the SIO in three main ways:

1) By interrupts - Interrupts can be generated under normal operations for things like Transmit Buffer Empty or Interrupt on Every Character: In addition to this there is a wide range of interrupts for special conditions which enable the software to act promptly in the case of errors occuring. Data can also be transferred between the SIO and the CPU by polling. In order to operate in the polling mode one merely disables the interrupts and tests the relevant status flags.

The third rrethod of data transfer is by use of the Wait/Ready pin on the SIO: This signal is designed to signal either the CPU or a DMA controller that it is ready for the next data transfer. If the Wait/Ready option was employed then it would mean that extra logic would be required to multiplex all of the Wait/Ready signals into one. Obviously the amount of extra logic required would depend on the degree of expansion but in any case the extra logic was considered excessive - especially considering that there are two alternative methods of transferring data: Provision has been made however for use of the Wait/Ready signal with the non-expanded i.e: the single SIO option.

If this option is used then the signal Est be programmed in the Ready state since the Wait condition produces a logic '1' on the pin and the Wait input of the Z80 is active low.

The CDC has four independant channnels. Two of them are used for the channels on the SIO leaving two spare: In the case of only one SIO being employed the extra outputs can be used for providing pulses to help operate the software - by providing a periodic interrupt for example.

Expansion

As it stands the design is capable of passing signals from the telephone line to Econet via the BBC: Obviously the number of sources that can be dealt with at one time is limited only by the space that the data from each source can successfully be squeezed into or how fast the primary processor can handle the data: Since there is, in theory, an infinite number of stations that the interface board will communicate with there seems little use for expansion - after all there is only one connection to the public service.

Expansion nay be required in the following cases:

1) A connection to an internal phone system which would effectively extend the range of Econet or possibly act as a bridge between Econets or an Econet and another network.

2) The direct connection of the Interface board to a serial device such as a terminal or printer. Although this application is unlikely it is not impossible and such a requirement would neccesitate an extra SIO for communication with such devices:

Since there will only be at most an extra two SIOs involved it is advised that they reside in the empty space of the LS138 assigned to the secondary processor.

As a footnote it is worth pointing out that the board could be used as a new Terminal Concentrator - the only difference between this and the standard X25 link being the software and the fact that many SIOs would be needed to "Concentrate the Terminals". It must be remembered that as more SIOs are used effective throughput on each channel will be reduced. It is therefore necessary to make a decision on the number of channels/throughput ratio. In this application more decoding will be needed for the extra SIOs and Appendix B shows how this can be done.

THE TELEPHONE INTERFACE

Obviously some circuitry will be needed to interface the Z80 SIO to the telephone: See Fig: 2 for details of the actual components needed.

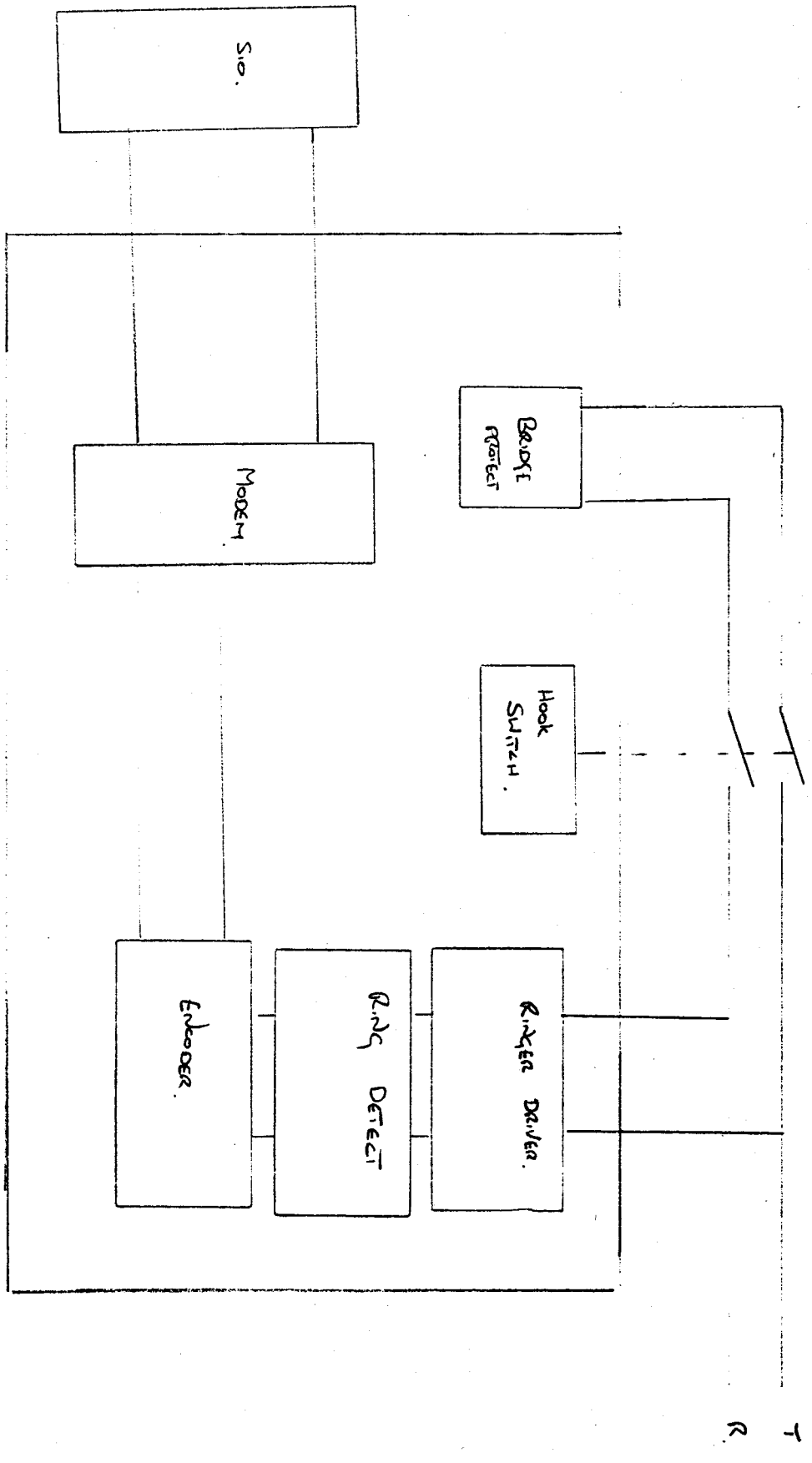
A Note on the Modem

Many different types of modem chip are available ranging from the primitive to the very sophisticated. The final choice of modem chip will depend, amongst other things, on the market that the gateway will be aimed at. A low cost, basic modem with only manual calling/answering will certainly suffice for the gateway aimed at the cheap end of the market where the user may want to transfer the odd file. However a more sophisticated device with auto calling/answering capability should be used if the interface is aimed at the type of person who would want to log in to several computers and/or filing systems during the course of a normal working day.

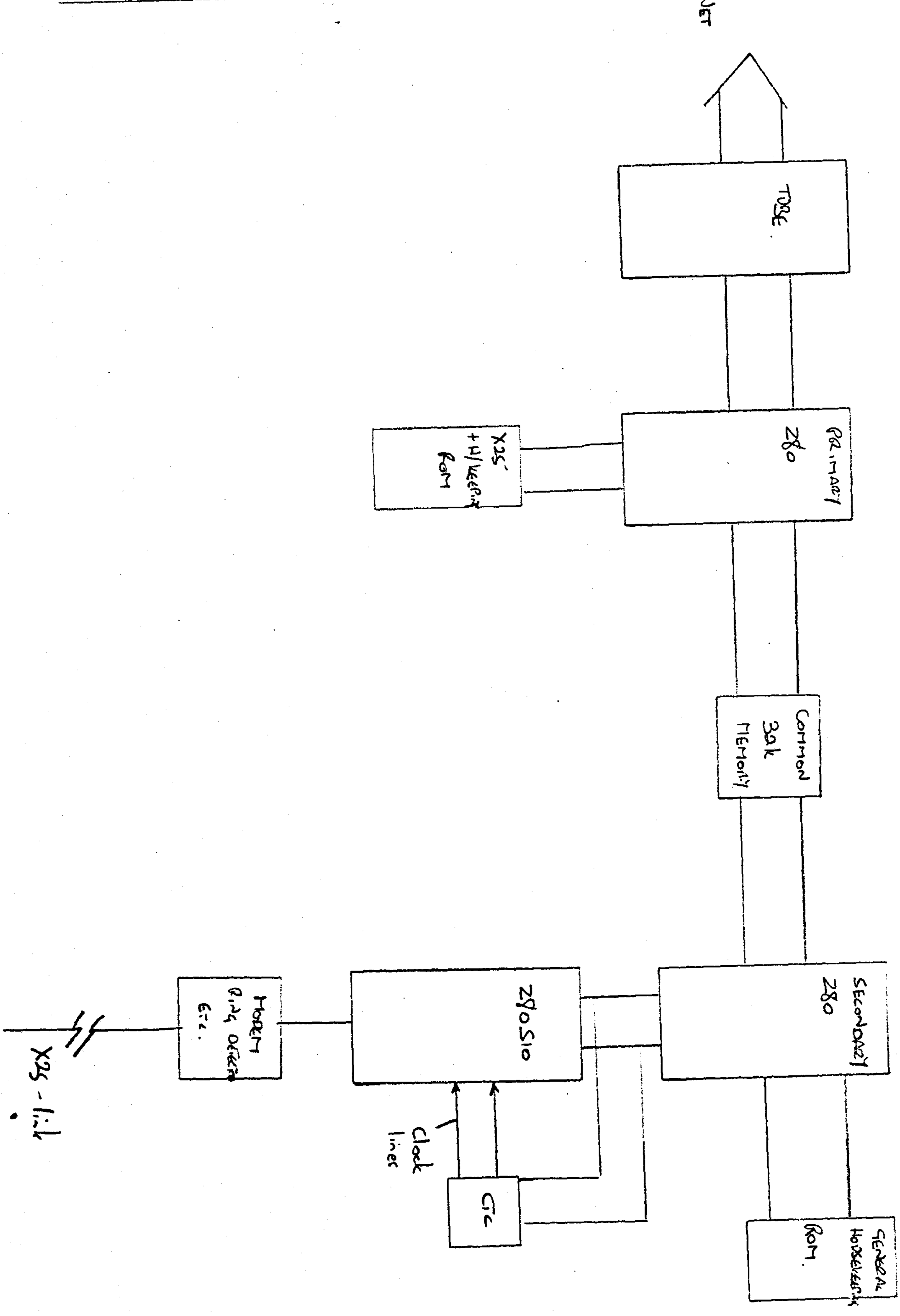
It is recommended that an auto call/answer facility is provided since integrated chip modems provide this facility for very little extra cost and also it is a good selling point.

The other components associated with the "telephone end" such as ring detectors, protection circuits and tone or pulse encoders are readily available and a design could be formulated fairly easily: This will be done when the actual modem is decided upon - this decision will be made when the actual application and market are defined.

FIG 2.

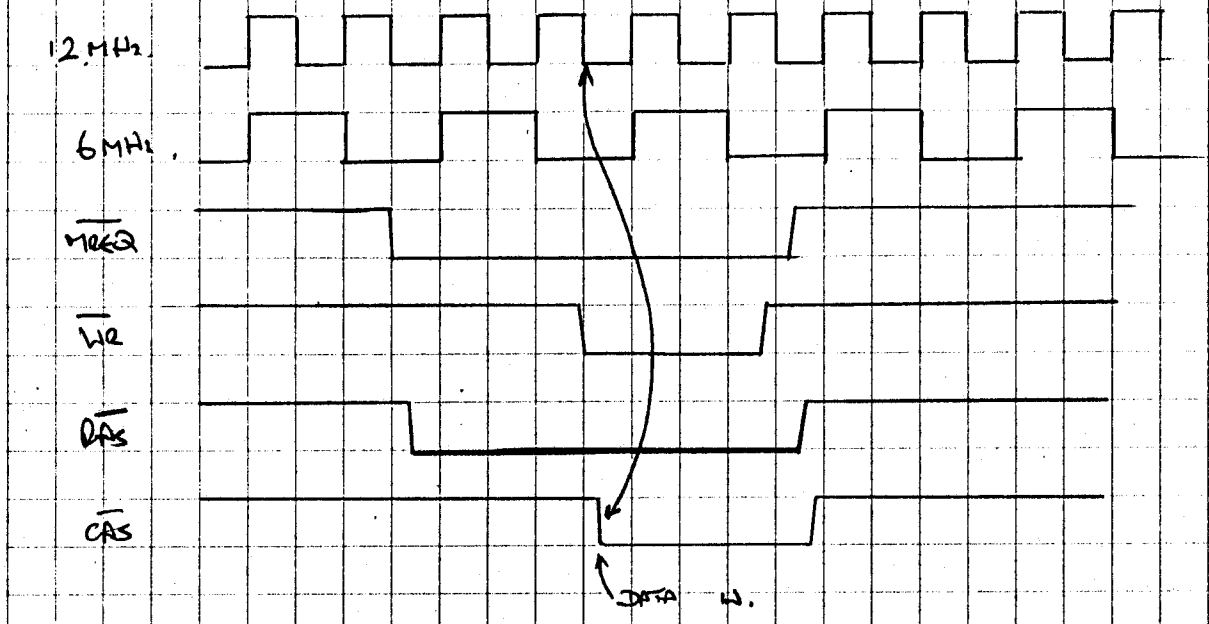

SIO.
MODEM.
BRIDGE PROTECT
HOOK SWITCH.
ENCODER.
R.ING DETECT
RINGER DRIVER.
T
R

APPENDIX A.

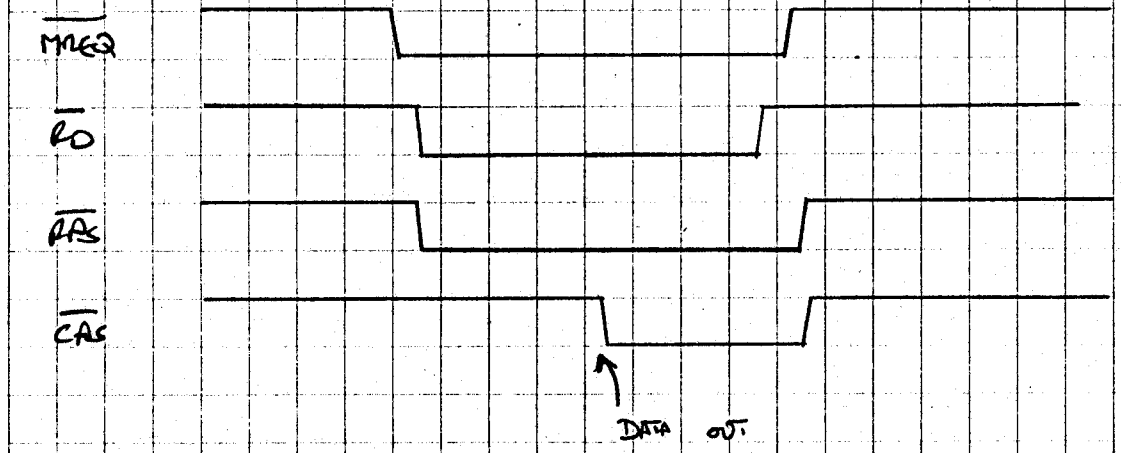

TO BBC + ECONET
TUBE
PRIMARY Z80
X25 + H/KEEPING ROM
COMMON 32k MEMORY
SECONDARY Z80
Z80 SIO
MODEM Ring detect etc.
X25 - link
Clock lines
CTC
GENERAL HOUSEKEEPING ROM.

1 sq = 40 ns

WRITE CYCLE



READ CYCLE



Timing Diagrams for RAM Access by Primary Processor.