6.3 Handshaking

The most difficult aspect of serial communication is being able to process characters as fast as they are received. At certain baud rates it may be necessary to suspend character transmission from the host device long enough for the receiving device to "catch up". The process of signalling between the devices when to transmit and when not to, has been coined "Handshaking". There are two basic methods of handshaking, hardware handshaking and software handshaking.

Hardware handshaking can be achieved by using pin4 and pin5 on the D-shell connector. From the EIA RS-232C specification, the functions of these two pins can be summarized as :

Pin4 : REQUEST TO SEND : This pin is used by the transmitting device to determine if the receiving device is able to receive data.

Pin5 : CLEAR TO SEND : This pin is used by the receiving device to inform the transmitting device that it is ready to receive data.

Therefore, to achieve hardware handshaking, pin4 on the PC is connected to pin5 on the CNC controller. Conversely, pin4 on the CNC controller must be connected to pin5 on the PC. The CNC controller could then suspend transmission by altering the voltage on it's pin5. This would therefore alter the voltage on pin4 on the PC, and the PC's serial interface would then know that it is "NOT CLEAR TO SEND" to the CNC controller.

Software handshaking is a much more elegant method of handshaking. Firstly it does not require an extra pin connections, and secondly, as the handshaking is controlled from within the software, it allows the status of the handshaking to be used in the control logic of the software. For example, while transmission is not possible, the software can perform other tasks such as updating screen displays etc., thus utilizing the processing power of the computer more efficiently. The BTR Interfaces on the CNC machines in this project all use software handshaking. The reason for this is that at the outset of the project, the serial tape readers connected to the CNC controllers all used software handshaking. In order to make use of the existing cabling, it was decided to develop serial interface software that incorporated software handshaking.

Software handshaking is a relatively simple procedure. It involves the transmission of two control characters from the receiving device to the transmitting device along the normal data transmission pins. (ie. along pin2 on the CNC controller which is connected to pin3 on the PC.) The two control characters are :

- XOFF (Character with ASCII value of 19)
- XON (Character with ASCII value of 17)

While the PC transmits on pin2, it continually monitors any incoming data from the CNC controller on pin3. As soon as the XOFF character is received by the PC on pin3, it suspends transmission on pin2. It then waits until it receives the XON character (also on pin3), upon which it resumes transmission on pin2.

The BTR software for CNC machines appears in Appendix F. As the standard serial interface (EIA RS-232C) is used on the PC's, high level functions are available in most programming languages to control the transmission/receipt of data. It is therefore not necessary to discuss memory locations and pin connections as was the case in section 5. Appendix E contains the structure and explanation of the BTR Interface program code for CNC machines. The high level functions used to transmit and receive data will be explained in that Appendix. For a sample session on using the BTR Interface for CNC machines, refer to Appendix G.

7. BTR DATABASE AND UTILITIES

The BTR Database is a simple system that stores all the NC part programs. The utilities allow the NC programmers to store and retrieve the NC part programs. The BTR Database is accessed by the BTR network whenever part programs are required on the shop floor. Figure 7.1 describes the interaction of the BTR Database and utilities with the rest of the system.

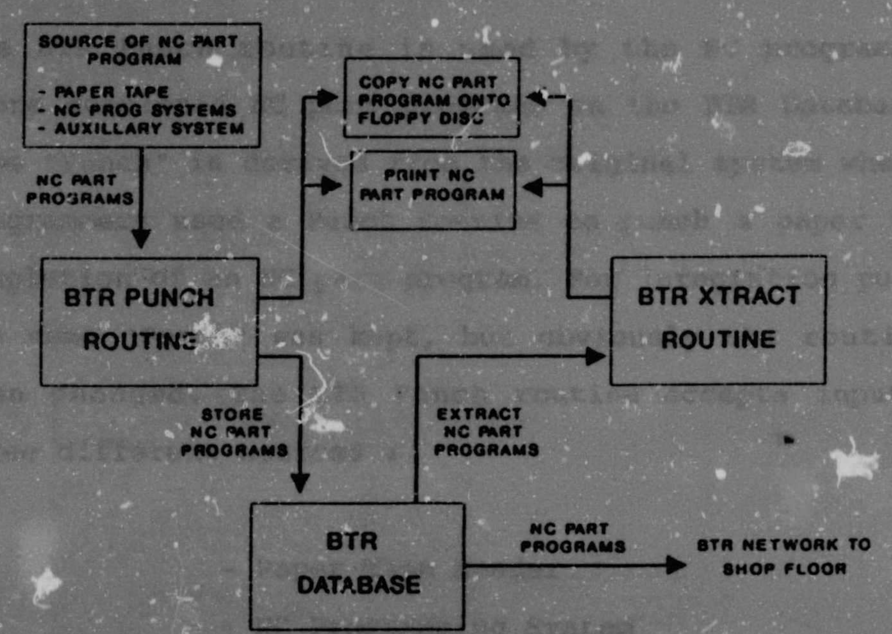

Fig 7.1 Interaction of BTR Database and Utilities with the rest of the BTR System

7.1 BTR Database

The BTR Database is physically an MS-DOS directory on a PC

7. BTR DATABASE AND UTILITIES

The BTR Database is a simple system that stores all the NC part programs. The utilities allow the NC programmers to store and retrieve the NC part programs. The BTR Database is accessed by the BTR network whenever part programs are required on the shop floor. Figure 7.1 describes the interaction of the BTR Database and utilities with the rest of the system.

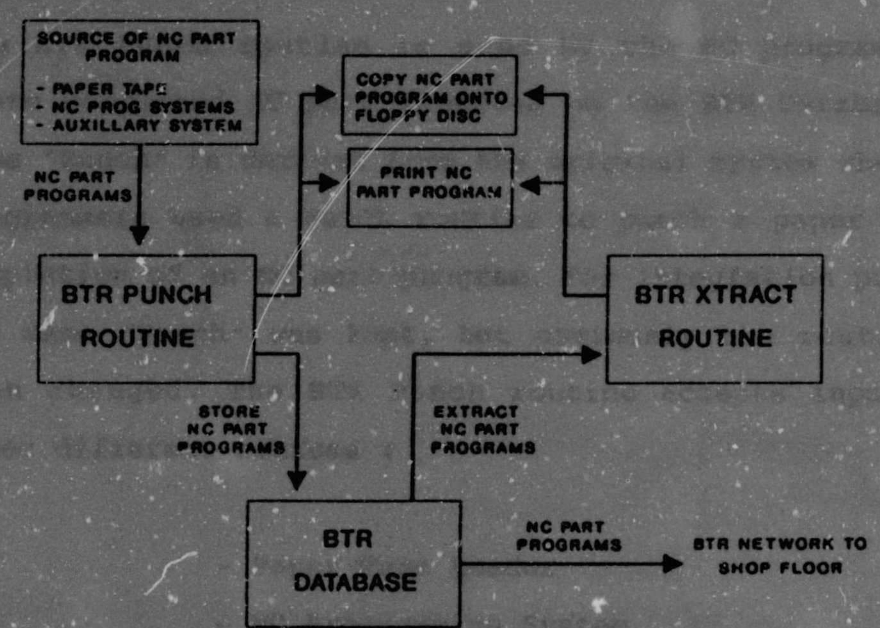

Fig 7.1 Interaction of BTR Database and Utilities with the rest of the BTR System

7.1 BTR Database

The BTR Database is physically an MS-DOS directory on a PC hard disk in the NC programming department.The directory is C:\BTR. All the NC part programs are placed in this directory by the NC programmers using the BTR Punch routine. The part programs have a 5 digit filename ranging from AA001 to ZZ999. The main function of the BTR Database is to make all the completed NC part programs available to the shop floor via the BTR Network.

7.2 BTR Punch Routine

The BTR Punch routine is used by the NC programmers to store completed NC part programs on the BTR Database. The name "Punch" is derived from the original system whereby NC programmers used a Punch routine to punch a paper tape on completion of an NC part program. For integration purposes, the name "Punch" was kept, but obviously the routine has been changed. The BTR Punch routine accepts inputs from three different sources :

- Paper Tape Reader
- NC Programming System
- Auxiliary System

Paper Tape Reader : This source is used when an NC programmer wishes to place an old NC part program on the BTR Database that has already been punched as a paper tape. A paper tape reader has been connected to the PC containing the database via RS-232 communications. The NC programmer inserts the paper tape into the paper tape reader and the Punch routine automatically reads the part program.

NC Programming System Source : The PC containing the BTR Database is also linked to the CAD/CAM mini computer via RS-232 communications. NC part programs are transferred to the PC and then post processed, (see Fig. 7.2), and placed in a file called "PUNOUT". When the NC programmer runs the BTR Punch routine and specifies the "Post Processor File" option, this "PUNOUT" file is accessed as the source.

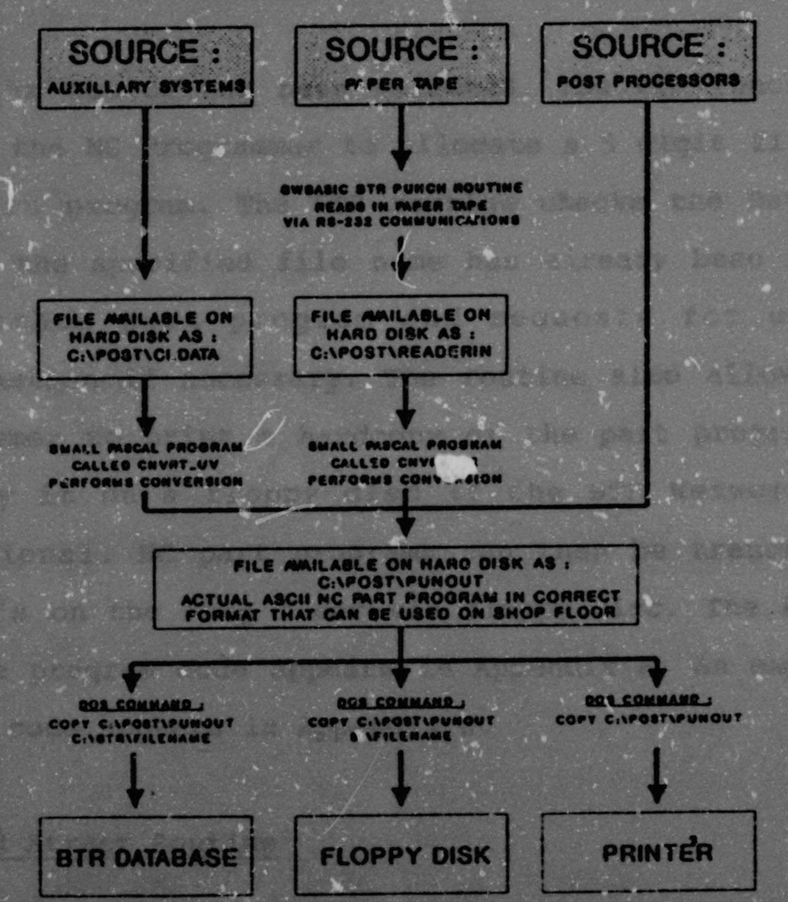

Fig 7.2 NC Part Program Sequence during BTR Punch Routine

Auxiliary System Source : Residing on the CAD/CAM system are a number of files that were created on an earlier NC system. This system is not used anymore, but the files are still available to the PC. These files can be brought across to the PC via the post processors and the RS-232 link.

The Punch routine also runs small conversion programs if the Paper Tape Reader or the Auxiliary source is selected. These programs delete the excess "space" characters that are inserted in the files during the transfer process.

Once the source has been selected, the BTR Punch routine allows the NC Programmer to allocate a 5 digit filename to the part program. The BTR routine checks the Database to see if the specified file name has already been allocated to another part program and requests for overwrite confirmation if necessary. The routine also allows the NC programmer to print a hardcopy of the part program and/or to copy it on a floppy disc if the BTR Network is not operational. NC part programs can then be transferred to the PC's on the shop floor via floppy disc. The BTR Punch routine program code appears in Appendix K. An explanation of the code appears in Appendix J.

7.3 BTR Xtract Routine

This routine is used by the NC programmers to extract or access an NC part program that has already been placed in the BTR Database. It allows the NC programmers to print the NC part program and/or to copy it onto a floppy disc. The BTR Xtract routine program code appears in Appendix M. An explanation of the code appears in Appendix L.

8. BTR NETWORK

The BTR Network is the medium across which the NC part programs are transferred from the BTR Database to the BTR interface on the shop floor. The network hardware and software are commercially available, and were not developed as a part of this project. Therefore this section will concentrate on the network system design in terms of functionality which is obviously important to this project.

The environment in which the network has to operate is an NC shop floor, which inherently is a harsh environment in terms of mechanical, electrical and electromagnetic interference. Forklift trucks, trolleys and general traffic provide the mechanical interference, while the starting and stopping of electrical motors in the machinery, welding and other electrical apparatus provide the electromagnetic interference. Apart from the mechanical interference, computer equipment and especially copper communication cabling is extremely prone to electrical and electromagnetic interference. For this reason the design of the network was based around a fibre optic communication medium.

The fibre optic cabling available on the market is extremely durable and not prone to electrical and electromagnetic interference. The silicon fibre itself is very delicate, but it is housed in PVC and has metal strengthening members that allow it flexibility and strength. The cable can tolerate a bend radius of 100 mm under tension. The cable is also physically lighter and smaller as opposed to it's copper alternatives. The method of data transfer through fibre optic cabling is by light pulses and not electrical pulses that are prone to electromagnetic interference as is the case with the copper cabling alternatives.

Many companies have developed proprietary copper and fibre optic networks. In many cases, only arbitrary standards exist, and there is no assurance that future technological developments can be assimilated, or that changing needs can be met. The network that will meet these requirements is one that is designed and conforms to internationally accepted standards. The network that was selected satisfies the full specification of the International Standards Committee, IEEE 802.3. The specification is commonly called "Ethernet". It has fast data transfer rates, (10 Mbits/second), coupled with considerable flexibility in both design and implementation due to the fact that it allows the mixing of copper and fibre optic cabling. The copper cabling is cost-effective in office environments, while the fibre optic cabling is available for harsher environments.

The Ethernet specification calls for a network design that is based around a common, logical "backbone", see figure

8.1. Devices (computers) are connected to this backbone by physically "tapping" into the cable. Devices can then "talk" across the backbone by placing "packets" of data on the backbone headed by a destination address. All other devices ignore the data apart from the device with the corresponding address.

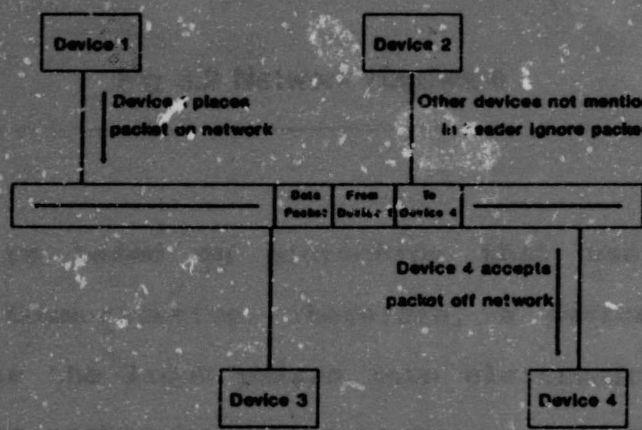

Fig 8.1 The Ethernet "Logical" Backbone

Unfortunately, fibre optic cabling cannot be "tapped" simply because a light pulse does not radiate any electromagnetic forces. Therefore, special devices, (called fibre optic hubs), are used to "split" the light pulse, see figure 8.2. This results in a physical network resembling a star type topology, but logically it remains a backbone because the fibre optic hubs distribute the light pulses along all connected cables, irrespective of the data's destination.