Figure 1 illustrates how the basic calculator will connect with external peripherals. (Room for 4 peripherals is shown. However, an extender box will make room for more, up to 15 total).

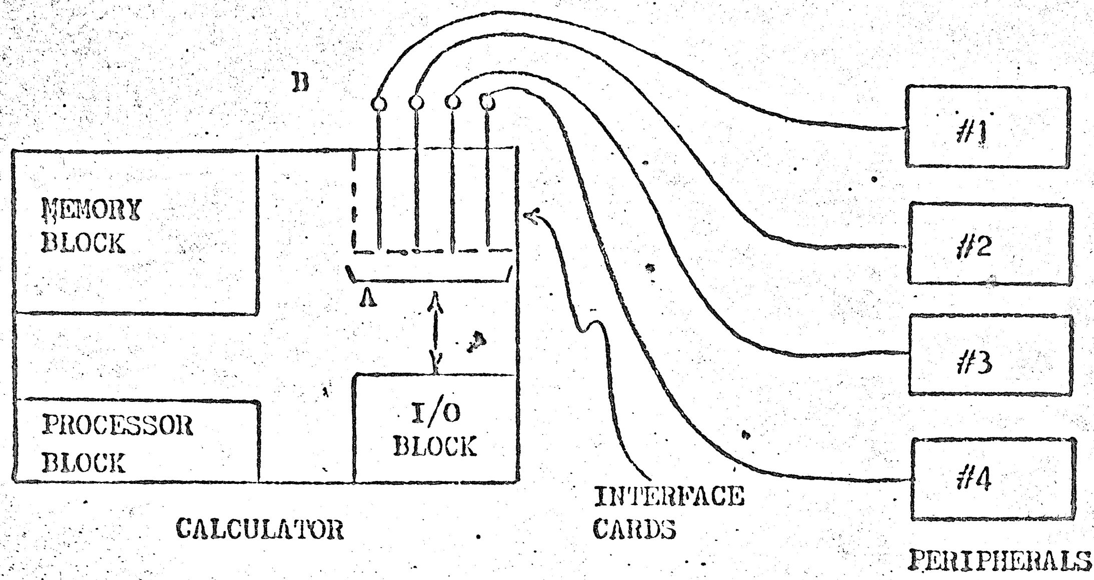

FIGURE 1. ILLUSTRATION OF CALCULATOR WITH CONNECTED PERIPHERALS.

The description of the input/output structure to follow will be given on the basis of describing the I/O lines connecting the I/O block and the interface card slots (See A Fig. 1).

The lines at point A , Fig.1, are divided into 7 catagories and are common to each I/O interface card slot. The 7 catagories are:

1. Output control lines
2. Output data lines
3. Output status lines
4. Output channel code lines
5. Input control lines
6. Input data lines
7. Input status lines

Figure 2, illustrates the lines at point A between the I/O block and the interface card slots. All output lines from the I/O block are sourced from TTL integrated circuits. Logic circuits on each interface card may place not more than 1 normal TTL load on any given output line or a maximum of 4 loads from the 4 interface slots per output line. The logic sense for all output lines except "Control Enable" and "Service Inhibit" is positive true ("1" = +5 V ; "0" = 0V). The Control Enable and Service Inhibit output lines are negative true ("1" = 0V; "0" = +5V).

All input lines to the I/O block are negative true and must be driven from open collector circuits on the interface cards. Pullup resistors are provided in the I/O block for each input line from the I/O interface slot. Input lines are used in a wired or configuration by each interface card.

Gary Egan

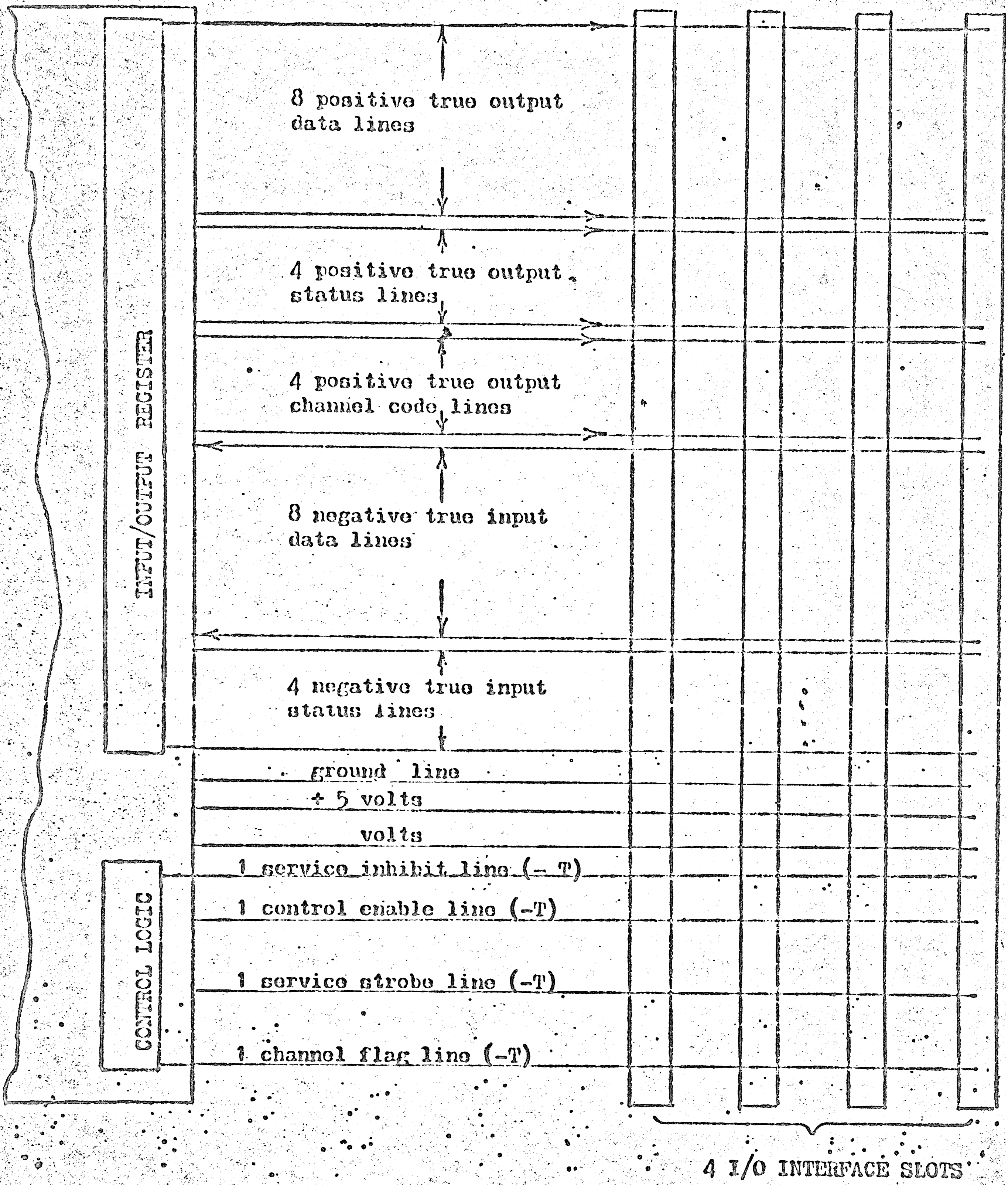

FIGURE 2. LINE ILLUSTRATION BETWEEN I/O BLOCK & INTERFACE SLOTS.

Input lines from an inactive interface card must be assured high (+V) so that they do not interfere with an active interface card's input lines. Figure 3 illustrates this.

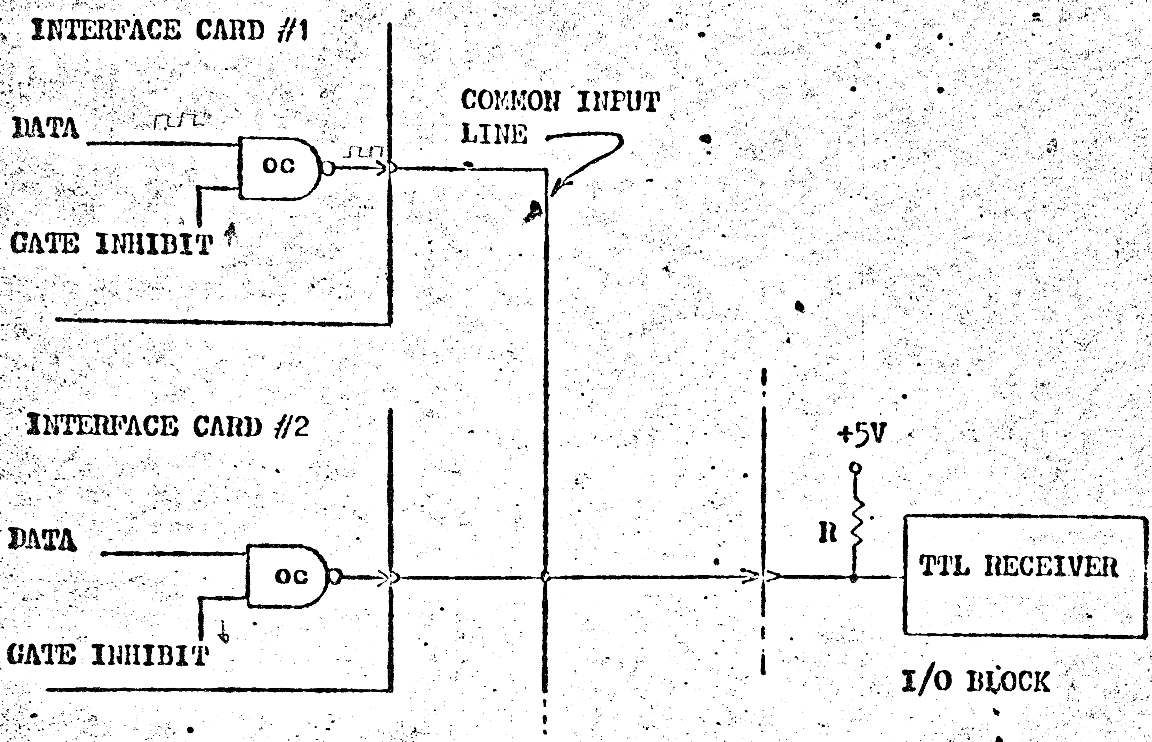

FIGURE 3. ILLUSTRATION SHOWING WIRED OR CONNECTION TO INPUT LINE.

If interface #1 is active the inhibit input to the open collector gate must be high (+V) so as to enable the gate to pass information. The inhibit input on interface #2 to the open collector gate must be low (OV) so that interface #2 does not interfer with active interface #1. In general only one interface will be active at a given time. Exceptions will be discussed later.

Output information is passed from the I/O block to the interface cards using the data out lines, the status out lines, the channel code lines and the control enable line. The channel code lines provide information to select a specific interface. These lines are decoded on the interface card which allows the card to be slot independent. Data and status out are passed to the interface card at the same time the channel code is. Approximately 10 us later control enable is passed to the interface card on the Control Enable line. By using the decoded channel code on the interface card and the control enable signal, the command signal is given to the peripheral to receive data. The I/O structure in this mode operates on a handshake method, therefore it is necessary for the peripheral to reply after it has received information with a strobe. The strobe is processed on the I/O interface card and passed to the I/O block through an open collector gate as the Channel Flag signal. The Channel Flag signal to the I/O block is negative true and must have a width of 0.3 us pw 10 us. The Channel Flag signal clears the Control Enable signal from the I/O block to the I/O interface card. Figure 4 illustrates the logic for a typical output interface.

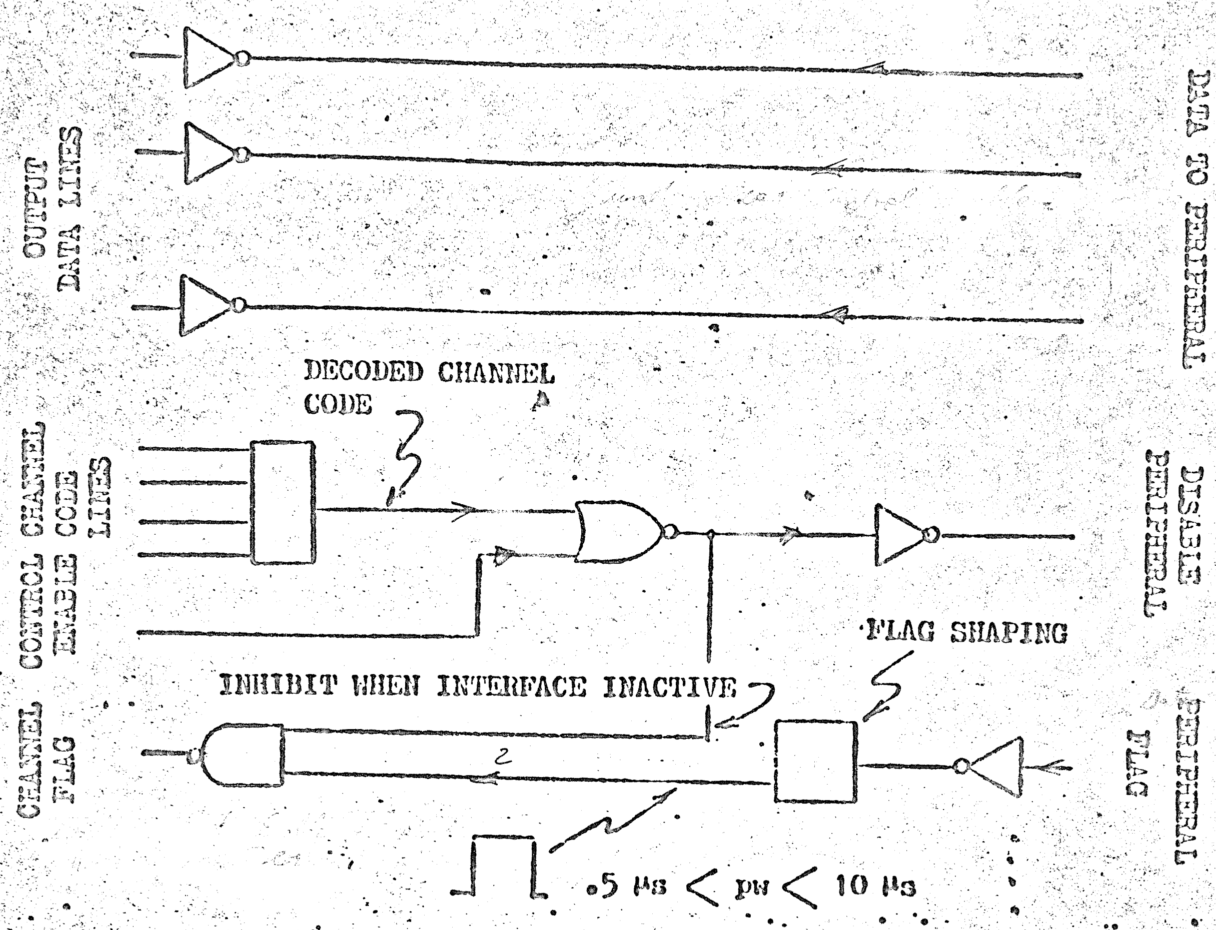

FIGURE 4. Some Typical Output Interface Logic.

Input information is passed to the I/O block from the interface by two methods. The first method may be called a control-handshake method and the second method may be called the keyboard-service method. A third method involves calling status from a peripheral by the calculator and will be discussed after the description of methods 1 and 2.

The control-handshake method requires the calculator to command the peripheral before the peripheral will respond. The sequence of events may be stated as follows. The channel code is sent from the I/O block to the interface card where it is decoded. The Control Enable signal is sent to the interface card and combined with the decoded channel code to form the command signal being sent to the peripheral. When the peripheral has responded and data is present at the interface the peripheral returns a strobe to the interface. The strobe is received by the interface card and sent to the I/O block as the Channel Flag input signal. The Channel Flag signal causes the data from the interface card to be stored in the I/O block and clears the Control Enable signal comming back to the interface card which causes the command signal to be cleared going to the peripheral. This holds the peripheral from another response until another command is given by the calculator. The Channel Flag signal must have the same characteristics as in the output mode described previously. Figure 5 illustrates typical logic for an input interface using the control-handshake method. When operating in either the output mode or the input mode only one interface is active at a time.

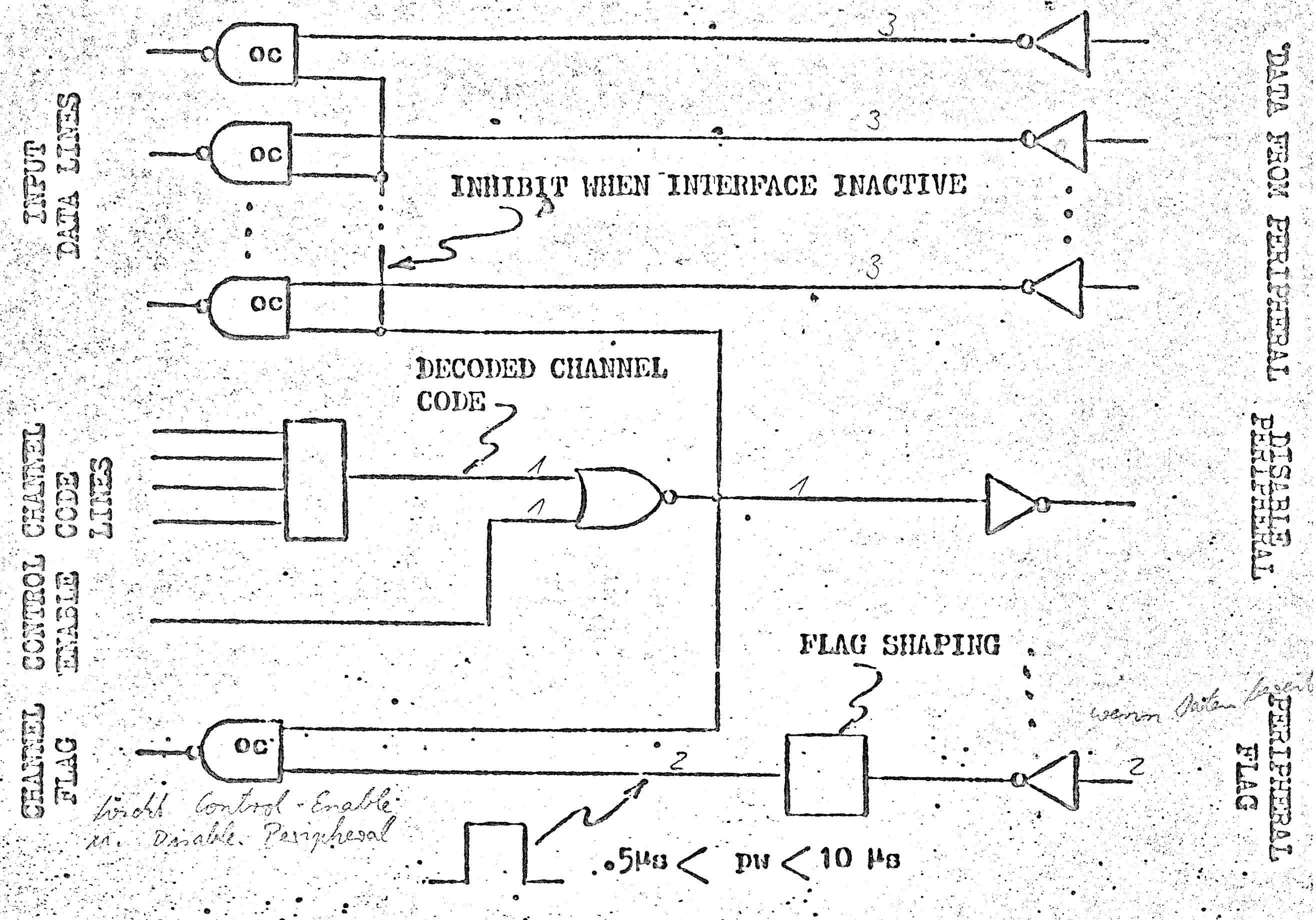

FIGURE 5. TYPICAL INPUT INTERFACE LOGIC

The keyboard-service method of input enters information in the same way as the calculator keyboard does. Data is sent to the calculator as keycodes from the peripheral with no command needed other than the peripheral be enabled just as the calculator keyboard is enabled.

The Marked Sense Card Reader or Paper Tape Reader are examples of peripherals which use the keyboard-service method of entry. The entry sequence of events may be described as follows. The peripheral is manually activated (card placed in card reader, key pushed, etc.). This places data at the interface card. Next a strobe to the interface card is received and sent to the I/O block as the service strobe input. The Service Strobe signal causes the data from the interface to be stored in the I/O block and requests the processor to service the data. This method also has a lockout. The Service Inhibit signal will inhibit entry under this method. Thus when the control-handshake methods of inputting and outputting data are used the Service Inhibit signal will be given to to inhibit the calculator keyboard and all other peripherals using the keyboard-service method of entry.

The Service Strobe signal is negative true and must last for at least 750 usec. and may last indefinately. The reason the upper limit is indefinite is due to the fact that the pulse or strobe may be interlocked with the depression of a key and remain as long as the key is depressed. This has no effect on the processor since as soon as the input is accepted the Service Strobe signal is ignored by the processor. However, this strobe must be terminated before more information can be entered. When data is accepted by the processor the Service Inhibit signal is given until that data has been processed.

Under the keyboard-service method of entry it is possible for more than one peripheral to be active at one time. If this happens invalid information will try to be entered. Therefore since the user is controlling this mode of entry he should allow only one keyboard-service type peripheral to operate at one time. Figure 6 illustrates the logic for a typical interface using the keyboard-service method of entry.

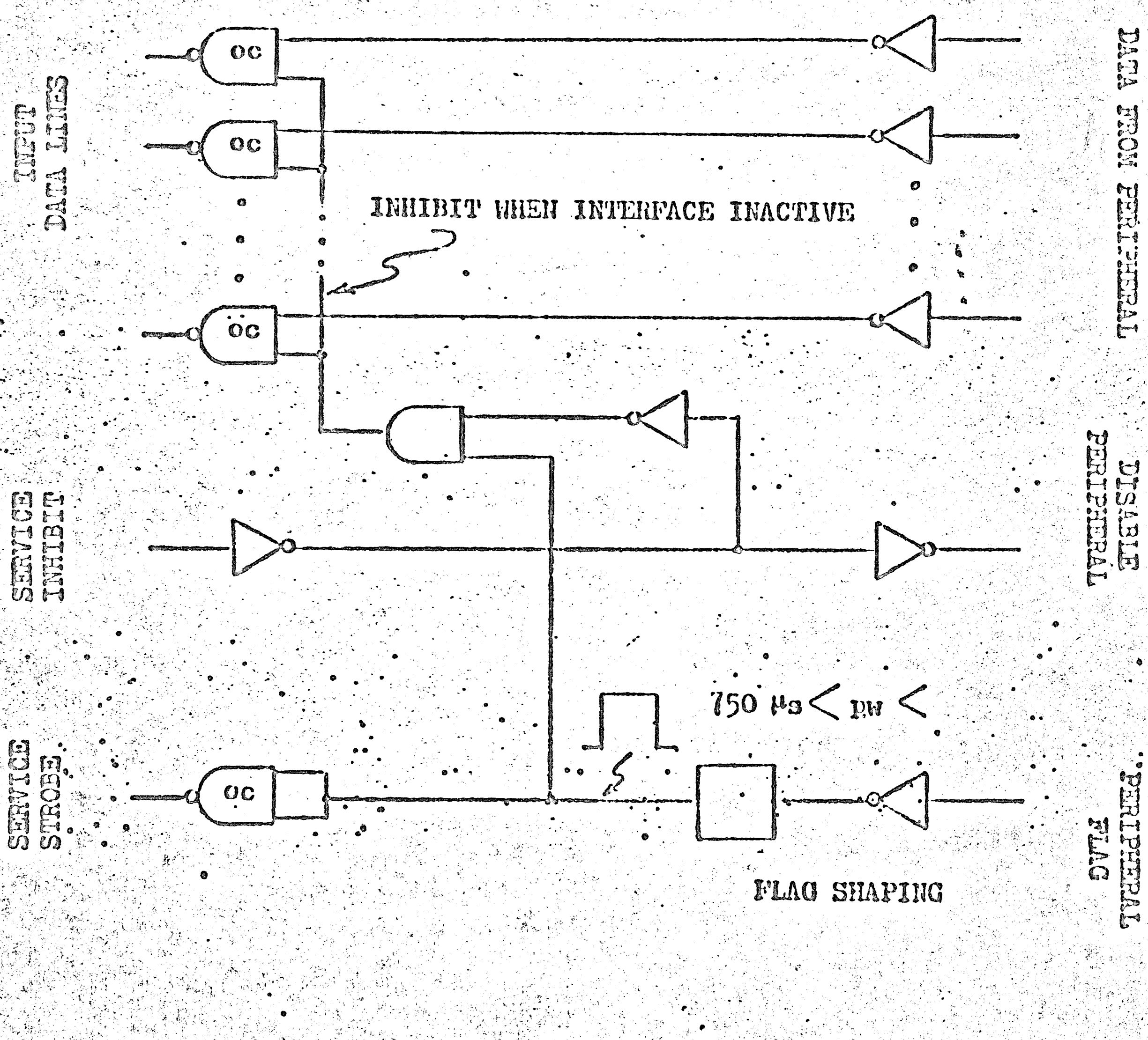

FIGURE 6. TYPICAL LOGIC FOR KEYBOARD-SERVICE METHOD OF INPUT.

As mentioned previously, a third method of input where the calculator requests status from a peripheral may be used. This is usually combined with the control method of entry. To enter status on the input lines the Channel Code is given and decoded on the interface card. This decoded channel code enables the status gates on the interface card and allows the I/O block to store this information with an instruction from the calculator. Figure 7 illustrates the logic for entering status. (NOTE: the 4 status input and the 8 data input lines are entered under this method).

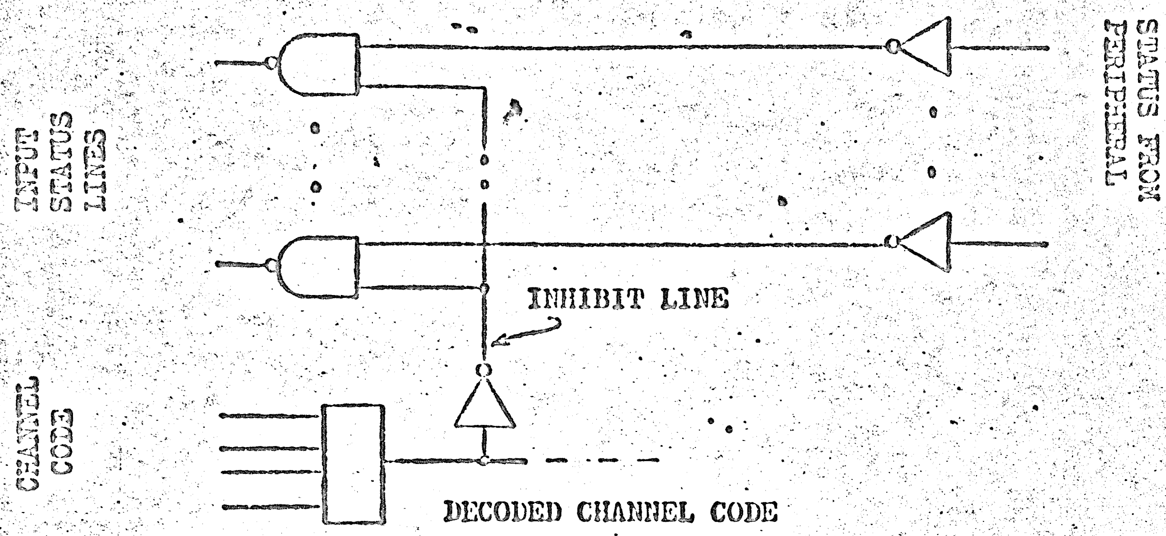

FIGURE 7. TYPICAL LOGIC FOR REQUESTING STATUS FROM A PERIPHERAL.

A peripheral which acts as a receiver and a transmitter and uses the control-handshake method will need an interface card that will pass data both directions. Therefore, an output status bit will be necessary to determine if the direction is input or output. This bit has been assigned to the most significant status bit. If the bit is high (+5) this implies input. If the bit is low it implies output.

At the present time the interface cards contain about 16 square inches of space each. A cable is connected to the extended end with a length of up to 6 ft. The present convention is to require all signals in the cable to be negative true. The drivers and receivers will have TTL levels.

As mentioned on page 3 the 4 channel code lines are decoded on each interface card to allow the interface card to be slot independent. There are many ways to do the decoding, however, Figure 8 illustrates a method where the 4 channel code lines may be decoded to one of sixteen channels.

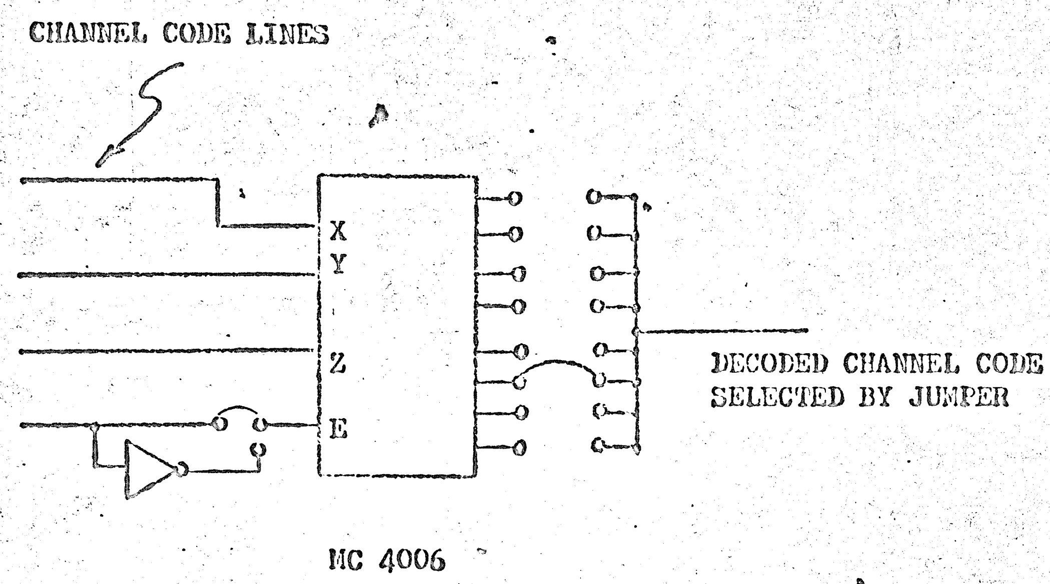

FIGURE 8. CHANNEL LINE DECODING METHOD.