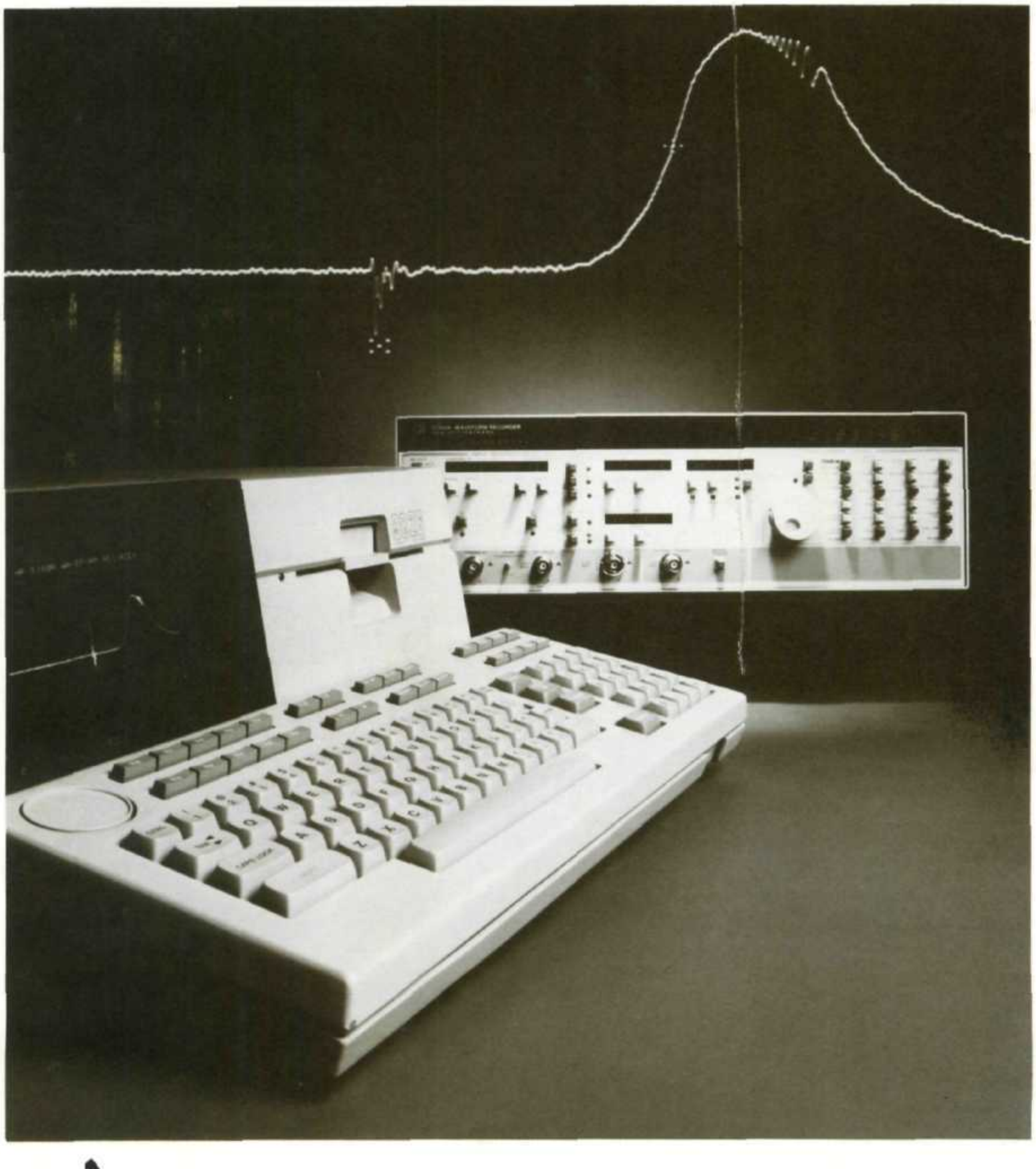
Application Note 313-8

Using the Direct Memory Access Capability of the HP 5180A Waveform Recorder with the HP 9826 Desktop Computer.



HEWLETT PACKARD

Please Note:

This application note is a revision of an earlier note bearing the same title which was released in February 1983 (HP PN 5952-7676). The principal difference between the older note and this revision is that this version describes the "FSO" DMA transfer which was not available at the time the first note was published.

5180 Waveform Recorders with serial numbers 2324A00671 and above are capable of performing "FSO" transfers. Instruments with lower serial numbers may have this capability retrofitted. To check if your 5180 can perform "FSO" transfers, set the HP-IB address switch to 94, press <SHIFT> <BUS ADDR>, then press <SHIFT> <OFFSET>. If the number that appears in the 5180 display area is below 1.4, your instrument will not be able to perform "FSO" transfers. It will, however, perform "Fast O" and "Fast I" transfers using the software described in this note. If you are working with a 5180 which cannot perform "FSO" DMA transfers, this application note can still be useful to you. Simply skip over the sections describing "FSO" and the auto stop switch and don't press the "FSO READ" special function key on the 9826 when you run the demonstration program.

INTRODUCTION

Direct memory access (DMA) provides a way to move data very quickly between the 5180 and a controller such as the 9826. This application note shows how the DMA feature can be implemented in BASIC with a 9826 (or 9836) as the controller. Although the focus is on the 9826 and BASIC, this note should be helpful to you even if you are working with a different controller and another language.

This note begins with a general description of DMA on the 5180, a comparison of DMA with the HP-IB*, and a description of two practical DMA applications. Next, the subroutines used to perform DMA are shown along with a line by line description of how they work. After that, a DMA demonstration program is presented and instructions are given for its use. The first five appendices cover error recovery, DMA hardware troubleshooting, the DMA cable, the theory of operation for 5180 DMA transfers, and the 5180 "auto-stop" switch. The sixth appendix gives a listing of the demonstration program.

"Fast O" DMA Transfers (5180 to 9826):

Three modes of DMA are offered on the 5180. "Fast O" (Fast Output) transfers a record from the memory of the 5180 to the memory of the controller, in this case, a 9826. With "Fast O", it is typically possible to transfer a 1K record (1024 data points) from the 5180 to the 9826 in about 80 milliseconds. A 16K record takes about 120 milliseconds. These figures assume that the 5180 displays are disabled during the DMA transfer by use of an "EN0" HP-IB command which will be discussed later in the "DMA subroutines" section of this note. If this is not done, the DMA transfers will take considerably longer.

"Fast I" DMA Transfers (9826 to 5180):

The second mode of DMA is called "Fast I" (Fast Input). With "Fast I", data is transferred from the controller to the 5180. (From the 5180's point of view, this is an input operation. For the 9826, of course, it is an output.) With "Fast I", it is possible to transfer a 1K record from the 9826 to the 5180 in about 85 milliseconds. A 16K record takes about 140 milliseconds. If the display outputs are not disabled during the transfer, the transfer times will be more than twice as long.

"FSO" DMA Transfers (measure, then 5180 to 9826):

The third DMA mode, "FSO" (Fast Sweep/Output), allows you to transfer a 1K record from the 5180 to the 9826 in less than 5 milliseconds. A 16K record can be transferred in about 45 milliseconds. "FSO" differs from "FAST O", however, in the fact that the DMA data transfer must immediately follow a measurement. The "FSO" feature is designed to minimize the measure-transfer-rearm cycle time. Using "FSO", it is possible to measure, transfer, and rearm more than 200 times per second.

In applications where it is not desirable to perform the DMA data transfer immediately after the measurement is completed, however, the "FAST O" DMA mode may be preferred. One such application is where the controller is not available to receive the DMA transfer after a measurement is taken. In this situation, it may be desirable to accumulate a number of measurements using the auto advance feature and transfer them to the controller at a later point in time. In cases such as this, the greater flexibility of the "FAST O" mode makes it preferable to the "FSO" mode despite its relatively high overhead time.

Comparison With HP-IB:

Although it is possible to transfer data between the 5180 and the 9826 through the HP-IB, the "Fast O" and "Fast I" features are more than ten times as fast, and "FSO" is several hundred times faster.

On the other hand, since you can transfer a 1K record (1024 data points) over the HP-IB in about one second, for many applications, the transfer speed offered by the HP-IB is quite satisfactory. If that is the case, the HP-IB method is probably preferable, since with the HP-IB you don't have to bother with the DMA board, the GPIO interface board, or the DMA cable. In other situations, however, speed is more important.

*HP-IB (Hewlett-Packard Interface Bus) is HP's implementation of IEEE 488-1978.

In system applications where a great deal of data must be transferred to a controller in a short period of time, DMA gives you capabilities that the HP-IB cannot provide. If, for example, you want to record 100 1K records in less than one minute, the HP-IB is too slow. The data rate is not fast enough to allow the 9826 to accumulate data as quickly as it is being generated by the 5180. On the other hand, the 16K internal memory of the 5180 is not large enough to record all of the data that is being stored. Using DMA, the data can be transferred quickly to the controller, leaving time for the controller to store it on a floppy disc or perhaps transform or manipulate the data as it is being recorded.

Other Considerations:

One important fact to remember about the 5180 is that it cannot measure and transfer data at the same time (See Appendix D). When a 5180 DMA transfer is taking place, no measurements can be taken and the 5180 cannot trigger. In measurements involving random events, this means that a certain amount of dead time will be introduced by the use of DMA or any other data transfer method. This can be a problem with "Fast O" transfers which can take between 80ms and 120ms to complete. However, the 5ms dead time associated with the "FSO" mode is comparable to that found with the auto advance feature, and should not present a serious problem.

Typical DMA Applications:

DMA is useful in many applications. The most obvious situation where fast data transfers are important is where the events which must be measured occur in rapid succession, but must all be recorded. For example, an automobile manufacturer may want to monitor the pressure inside the cylinders of an engine running at 6000 RPM. Data must be taken during the first 5ms of each engine stroke. After this measurement is complete, the 5180 has another 5ms to prepare for the next measurement. Since it takes less than 5ms to transfer a 1K record using the "FSO" feature, it is possible to perform a separate DMA transfer for each engine revolution. Using the subroutines described in this note, it would be possible to write a program which would allow measurements of 1K points each to be recorded for up to 750 consecutive engine cycles.

There are many other possible applications. A manufacturer of videotape recorders, for example, may want to test his products using a 5180 and a controller such as the 9826. Measurements taken on the 5180 by a test technician could be transferred by DMA to the 9826. The controller could then compare the measured waveform with a reference waveform to see if whether the unit under test was functioning correctly. Then, using the "Fast I" DMA feature, the measured and the reference waveforms could be transferred back to the 5180 where the technician could observe the two on a CRT display and check to see whether the discrepancy is due to an error in the test procedure, or due to a defective unit.

These two examples illustrate the wide variety of applications for which DMA can be useful. In almost any application involving the 5180 and a controller, DMA offers increased convenience, speed, and capability.

DMA SUBROUTINES

This section describes the "Fast_o_read", "Fast_i_write", and "Fso_read" subroutines which use DMA to transfer records between the 5180 and the 9826. A few additional main program lines are included that relate to the subroutines under discussion. Later in this note, the DMA demonstration program section shows how these subroutines can be used in practical applications.

DMA Set-Up Lines:

The set-up functions listed here must be carried out before any DMA transfers can be performed.

```
10     INTEGER Buffer(16384) BUFFER
20     Gpio=12                              ! Give select code of GPIO board here.
30     Hp_ib=704                            ! Give 5180 HP-IB address here.
40     ASSIGN @Gpio TO Gpio;WORD            ! Give I/O path name to interface.
50     ASSIGN @Buffer TO BUFFER Buffer(*);WORD ! Give I/O path name to 'Buffer'.
60     OUTPUT Hp_ib;"EN0"                   ! Turn off 5180 display to cut overhead.
```

The individual program lines are listed below. Along with each line there is a short explanation of why it is there and how it works.

```
10     INTEGER Buffer(16384) BUFFER
```

This line dimensions an integer array and indicates that it will be used as a buffer. The array shown here is rather large — it will hold a full 16K record from the 5180. If there is no need for a buffer of that length, a smaller one can be used instead.

String buffers and unnamed buffers are also available on the 9826. They are not as useful as array buffers, however, since data must be (slowly) loaded into an array before it can be plotted or manipulated arithmetically. With array buffers, data arrives at the 9826 in an immediately useful format.

One possible use for unnamed buffers is for situations where a variable length buffer is required. In such a situation, the "ASSIGN" statement can be used to create an unnamed buffer which requires no more memory than is necessary. An array or string buffer, on the other hand, must always be dimensioned to its maximum possible length. The line, "ASSIGN @Buffer TO Buffer [N*2]" creates an unnamed buffer which holds N data points and requires approximately N*2 bytes of memory when the program is run. "ENTER" and "OUTPUT" statements can be used to transfer data between this buffer and arrays on the 9826.

```
20     Gpio=12
```

If the select code for the GPIO interface you are using is not "12", substitute that value for the "12" in this line. For example, if it is "13", replace this line with "Gpio=13". That is the only change that you must make to the demo program for it to run properly with a select code other than "12".

```
30     Hp_ib=704
```

If you want to use a different HP-IB address, substitute it for "704" in this line. That is the only modification that you must make to change the HP-IB address in the demo program.

```
40     ASSIGN @Gpio TO Gpio;WORD
```

Here, an I/O path name ("@Gpio") is assigned to the GPIO interface card, which has a select code of "Gpio" (a numerical variable which probably equals 12). All I/O path names must start with "@". The "WORD" modifier indicates that 16 bits of data are used for each array integer. The "BYTE" (default) modifier uses 8 bits per array integer and results in errors if used for 5180 DMA transfers.

```
50     ASSIGN @Buffer TO BUFFER Buffer(*);WORD
```

This line assigns the I/O path name, "@Buffer" to "Buffer(*)" which was declared in line 10. As before, the modifier, "WORD" indicates that 2 bytes (16 bits) of data are combined to form each integer array element.

```
60     OUTPUT Hp_ib;"EN0"
```

This line disables the XYZ display outputs from the 5180 and sharply reduces the overhead associated with a DMA transfer. (For a 1K record, "Fast O" goes from 115ms to 80ms and "Fast I" goes from 290ms to 85ms.) This line is not included in the DMA demo program because for a demonstration you probably want to see the display. You can turn the display back on by sending "EN1" over the HP-IB.

At first glance, it might seem desirable to put an "EN0" statement at the beginning of "Fast_i_write" and an "EN1" at the end. This would reduce the DMA transfer overhead by 205ms while retaining the display. The problem with this is that the time required for the 5180 to respond to the two HP-IB commands far exceeds the reduction in overhead. For that reason, "EN0" should be part of the set up, and not part of the "Fast_o_read" or "Fast_i_write" subroutines.

"Fast_o_read" DMA Subroutine:

This subroutine performs a "Fast_O" DMA, transferring data from the 5180 to the 9826.

```
10 Fast_o_read:  GOSUB Find_length       ! Find length of record before transfer.
20     CONTROL @Buffer,4;0               ! Set # of bytes in buffer to zero.
30     CONTROL Gpio,0;2                  ! Reset GPIO interface.
40     CONTROL Gpio,3;0                  ! Writing zero clears data registers.
50     STATUS Gpio,3;A                   ! Set IN/OUT to 'in'.
60     CONTROL Gpio,2;1                  ! Set TREQ low.
70     TRANSFER @Gpio TO @Buffer;COUNT Length*2,WAIT ! Transfer record to 9826.
80     CONTROL Gpio,2;0                  ! Set TREQ high.
90     CONTROL Gpio,1;1                  ! Send final handshake.
100    RETURN
```

In order to make clear how this subroutine functions, individual program lines are given below along with explanations of what they do and how they work.

```
10 Fast_o_read:  GOSUB Find_length
```

The "Length" variable tells the 9826 what the current 5180 record length is. The "Find_length" subroutine uses HP-IB and the 5180 teach/learn commands to find the correct value for "Length". The "Find_length" subroutine can be found in the DMA demo program. (See Appendix F.)

```
20     CONTROL @Buffer,4;0
```

Buffer register 4 gives the number of bytes of data currently held in the buffer. By resetting this register to "0", this line tells the 9826 that the buffer is empty and makes it possible for the buffer to be filled with new data from the 5180.

If Register 4 is not reset to "0" in this line, and two DMA reads ("Fast O") are performed without a DMA write ("Fast I") in between, the results can be confusing. If the buffer on the 9826 has enough capacity for both records, the second record is entered after the first one. If the record length is 1024 points, the second record will fill array positions 1025 to 2048.

If the buffer is only large enough for one full record, the second DMA transfer terminates when the buffer is full. This means that if the buffer is full at the beginning of the second DMA transfer, the entire second record is lost. Setting register 4 to "0" ensures that every time "Fast_o_read" is performed, the transferred DMA record is stored at the beginning of the 9826 buffer.

```
30     CONTROL Gpio,0;2
```

By writing a "2" to the 9826 GPIO control register 0, this line resets the GPIO Interface. This is identical to a "RESET Gpio" command.

```
40     CONTROL Gpio,3;0
```

This line clears the data registers by writing a "0" to them. It also causes the GPIO interface to set the IN/OUT line high (see line 50). Since the IN/OUT line must be set high before data can be transferred from the 5180 to the 9826, a dummy read is performed in the following line.

```
50     STATUS Gpio,3;A
```

By performing a dummy read from the GPIO interface data registers, this line sets the IN/OUT line high. The IN/OUT line causes the 5180 to send data to the 9826 ("Fast O") when IN/OUT is high ("out"), and accept data from the 9826 ("Fast I") when it is low ("in").

```
60      CONTROL Gpio,2;1
```

Here, the "CTL0" (Control 0) line of the GPIO interface is set low. This control line is connected to the "TREQ" (transfer request) line of the 5180. When this line is pulled low by the 9826, the 5180 prepares to send a DMA record. The logic sense of "TREQ" and many of the other control lines can be reversed. To find out more about this, consult the 5180 service manual.

```
70      TRANSFER @Gpio TO @Buffer;COUNT Length*2,WAIT
```

This line performs the actual DMA transfer, sending a record from the 5180 to the 9826.

"@Gpio" is the I/O path name assigned to the GPIO interface (usually select code 12). Since "@Gpio" is the first I/O path name specified, data is transferred **from** the 5180. If it had appeared second (after @Buffer), as it does in the "Fast_i_write" subroutine, data would be transferred **to** the 5180 instead.

"@Buffer" is the I/O path name assigned to the 9826 buffer which will receive data from the 5180. In this case, "@Buffer" is the integer array "Buffer(*)". As mentioned in the discussion of line 10 of the DMA set-up, strings and simple buffers can also be used. Since "@Buffer" is the second I/O path name specified, data will be transferred into the array which has been assigned the I/O path name, "@Buffer".

If "Length" is equal to "1024", "COUNT Length*2" causes the 5180 to transfer 2048 bytes of data (1024 points). The "TRANSFER" syntax was designed to be used with all I/O operations, many of which deal with bytes (8 bits) instead of words (16 bits). Because of this, the length of a transferred record is always specified in bytes, even when 16 bits of data are transferred with each handshake. This means that the number following "COUNT" must always be twice the number of samples to be transferred.

Including "WAIT" at the end of the TRANSFER statement causes the 9826 to wait for the transfer to be completed before executing any of the program lines following the "TRANSFER" statement. If "WAIT" is not specified, the 9826 performs the DMA transfer and continues with the rest of the program in parallel. For slow I/O operations this may be useful, but for DMA, simultaneous transfer and program execution only causes unnecessary complexity and trouble. The "WAIT" modifier eliminates this possible source of errors.

```
80      CONTROL Gpio,2;0
```

Here "TREQ" (transfer request) is set high again. At the end of the next handshake, this will cause the 5180 to end the DMA transfer and return to normal operation.

```
90      CONTROL Gpio,1;1
```

This line simulates a final data handshake by toggling the DCTL (Data Control) line once. This must be done to cause the DMA state machine in the 5180 to cycle once more and recognize that TREQ has gone high. (See Appendix D: Theory of Operation.)

"Fast_i_write" DMA subroutine:

This subroutine causes the 9826 to transfer data from an internal buffer to the memory of the 5180. This is called a "Fast I" transfer since, from the perspective of the 5180, it is a data input.

```
20 Fast_i_write:   CONTROL @Buffer,4;Length*2 ! Set # of bytes to 2 x Length.
30      CONTROL Gpio,0;2                      ! Reset GPIO interface.
40      CONTROL Gpio,3;0                      ! Writing zero clears data registers.
60      CONTROL Gpio,2:1                      ! Set TREQ low.
70      TRANSFER @Buffer TO @Gpio;COUNT Length*2,WAIT ! Transfer record to 9826.
80      CONTROL Gpio,2;0                      ! Set TREQ high.
90      CONTROL Gpio,1;1                      ! Send final handshake.
100     RETURN
```

"Fast_i_write" differs from "Fast_o_read" in only four places. The first is that line 10 of "Fast_o_read" is left out. This is because "Length" has already been given a value if the 9826 has a record that it is about to transfer to the 5180. The second difference is that line 50 from "Fast_o_read" has also been left out. This dummy read is not performed because the IN/OUT line must be left low to perform a DMA write. Since IN/OUT line is already set low in line 40, line 50 can be left out. The other two lines which differ are shown below.

```
20 Fast_i_write:    CONTROL @Buffer,4;Length*2
```

This control statement resets "@Buffer" to its "full" value. This permits a "Fast I" transfer to be performed even after a transfer to mass storage or another "Fast I" transfer has taken place.

Register 4 of a buffer on the 9826 gives the number of bytes currently held in the buffer. When a record is transferred out of the buffer, this register returns to "0" even though the data has not been lost. If another write is attempted from this buffer an error will result since, to the 9826, the buffer is empty. If, however, this register is reset to its "full" value, another write can be successfully performed.

As before, if the record is 1024 samples long, the length register must be set to 2048 to indicate that the buffer contains 2048 **bytes** of data. Just as the buffer should be set to be empty before each DMA read operation, it should be set to be full before each DMA write.

```
70      TRANSFER @Buffer TO @Gpio;COUNT Length*2,WAIT
```

As discussed earlier, reversing the order in which the I/O path names are specified ("@Buffer TO @Gpio" instead of "@Gpio TO @Buffer") reverses the direction of the DMA transfer. In this case, data is transferred from the 9826 buffer ("@Buffer") to the 5180 ("@Gpio").

"Fso_read" DMA Subroutine

This subroutine transfers several records from the 5180 to the 9826 through use of the "FSO" DMA transfer mode:

```
10 Fso_read:  GOSUB Find_length          ! Find length of record before transfer.
15     OUTPUT Hp_ib;"FM1"                ! Turn on "FSO" mode with HP-IB command.
20     CONTROL @Buffer,4;0               ! Set # of bytes in buffer to zero.
30     CONTROL Gpio,0;2                  ! Reset GPIO interface.
40     CONTROL Gpio,3;0                  ! Writing zero clears data registers.
50     STATUS Gpio,3;A                   ! Set IN/OUT to 'in'.
60     CONTROL Gpio,2;1                  ! Set TREQ low.
70     TRANSFER @Gpio TO @Buffer;WAIT    ! Fill buffer with DMA records from 5180.
80     CONTROL Gpio,2;0                  ! Set TREQ high.
90     CONTROL Gpio,1;1                  ! Send final handshake.
95     OUTPUT Hp_ib;"FM0"                ! Turn off "FSO" mode with HP-IB command.
100    RETURN
```

"Fso_read" and "Fast_o_read" differ in only three places. Two involve the "FM1" and "FM0" HP-IB commands that are used to turn the "FSO" mode on and off. The third is the transfer statement. In the "Fso_read" subroutine, several records are transferred by one program line. By leaving unspecified the number of words to be transferred, this subprogram indicates that the 5180 is to keep transferring records until the buffer is full.

```
15   OUTPUT Hp_ib;"FM1"
```

The "FM1" (Fast Measure/Output) HP-IB command turns on the "FSO" mode.

```
70    TRANSFER @Gpio TO @Buffer;WAIT
```

As mentioned above, this transfer statement causes the 5180 to keep sending data until the buffer is full. Another approach would be to use a "COUNT" modifier with a number of bytes corresponding to the number of records that you want to transfer. This number could be varied through inputs to the program.

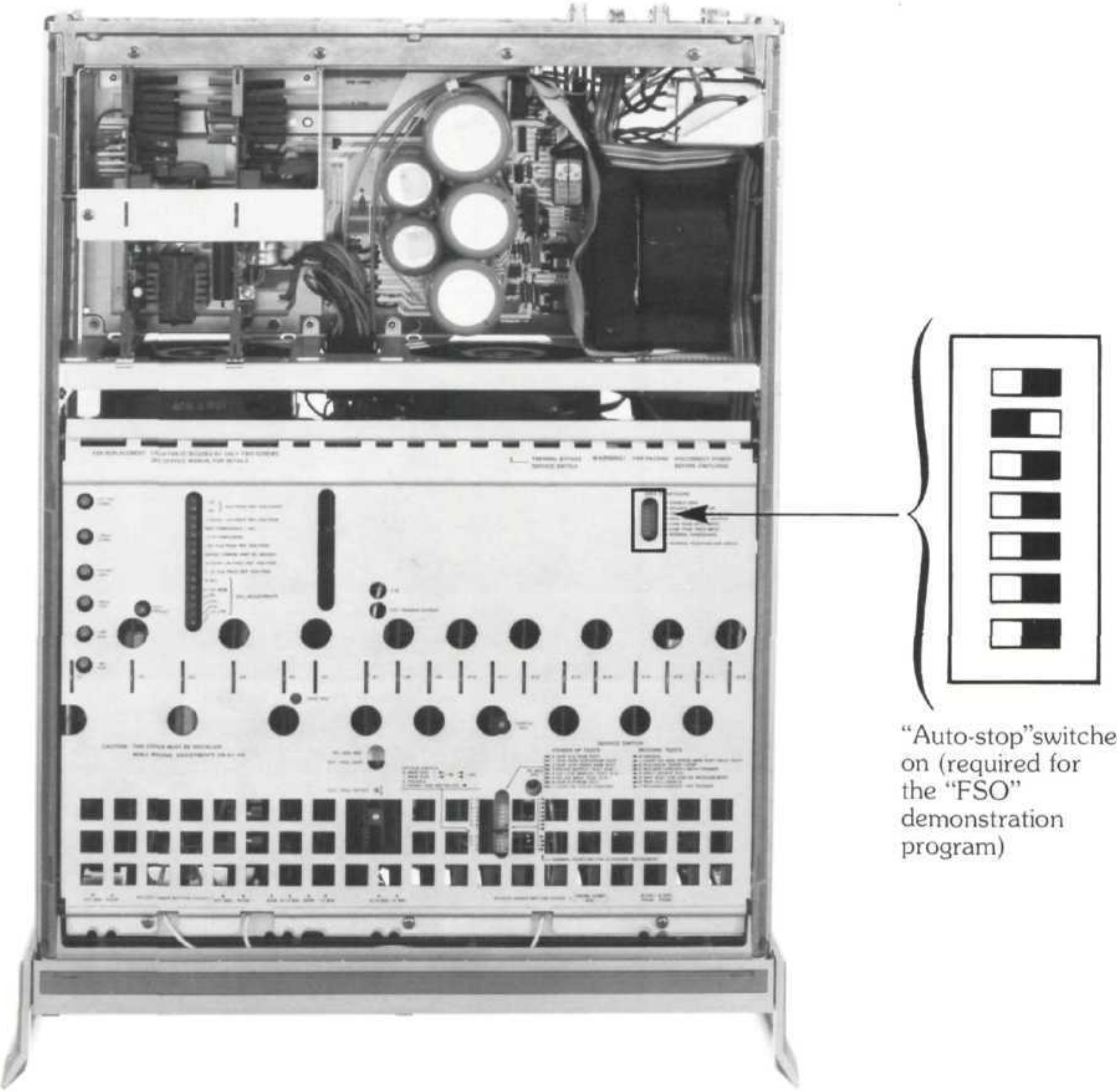
```
95   OUTPUT Hp_ib;"FM0"
```

The "FM0" HP-IB command turns off the "FSO" mode.

EQUIPMENT FOR DMA DEMO PROGRAM:

To run the DMA demo program described later in this note, you will need the following hardware items:

1) A 5180 Waveform Recorder with FSO capability (firmware revision 1.4 or later*). All of the DMA switches except "auto-stop" be pushed to the right as shown in Figure 1. The "auto-stop" switch should be pushed to the left. If you remove the top cover of the 5180, you will find these switches towards the back of the instrument on the right hand side.



"Auto-stop"switched on (required for the "FSO" demonstration program)

Figure 1. Correct position for DMA configuration switches on the A15 board inside the 5180. To run the "FSO" demonstration subroutines given in this note, the "auto-stop" switch must be pushed to the left as shown above. The "auto-stop" switch is described in Appendix E.

*Instruments with serial number 2324A00671 and above include this feature. Instruments with lower serial numbers may have this capability retrofitted. To check the firmware revision of your instrument, set the HP-IB address switch to 94, press <SHIFT> <BUS ADDR>, then press <SHIFT> <OFFSET>.

2) A 9826 or 9836 Controller with the HP 98622A GPIO interface board and the HP 98620A DMA board. On the GPIO interface board, the switches should be set and the burst jumper opened as shown in Figure 2. To run the programs given in this note, the 9826 should be configured with at least 50K bytes of user memory. The language used in these applications is BASIC 2.0, with the BASIC Extensions 2.0 (or a later revision). Without the extensions, DMA transfers cannot be performed.

3) A CRT display, such as the 5181A, a 1340A, or an oscilloscope. If you don't have a display or don't wish to use one, the programs will still work. On the other hand, debugging and program development are much more difficult when you can't see the 5180 measurements as you make them. In the rest of this note, the CRT display is referred to as the 5180 display.

4) A 3312A Function Generator or some other signal source.

5) 4 coaxial cables.

6) An HP-IB cable.

7) A 10875A or 10875B DMA interface cable. For more information on this cable see Appendix C.

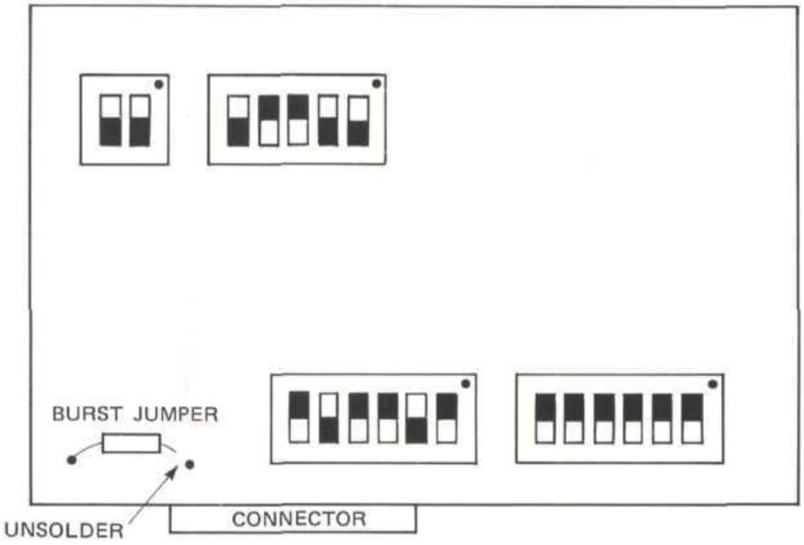


Figure 2. 98622A GPIO Interface Board, Switch, and Jumper Configuration. For rocker switches, press the shaded side down. For slide switches, move the switch to the shaded side.

THE DMA DEMONSTRATION PROGRAM:

The program listed in Appendix F is intended both as a general utility program and as a demonstration of the DMA subroutines described earlier in this note. This section will begin with instructions showing how to set up the equipment and get the program started. That will be followed by a set of demonstration procedures which will show what this program can do and how you can use it.

DMA Hardware Set-Up Checklist:

Before running the DMA demo program, be sure that the following items are covered:

1) See that all of the DMA switches inside the 5180 are properly set. (See the schematic in Figure 1. The auto-stop switch should be on.)

2) Check that the 9826 has at least 50K bytes of user memory (i.e., memory available after BASIC and the extensions have been loaded—current versions of BASIC and its extensions require about 460K of memory). Also, see that the DMA board and the GPIO interface board are present. Be sure that the switch settings on the GPIO interface board match the example in Figure 2.

3) Attach the HP-IB cable from the 9826 to the 5180. Check to see that the 5180 HP-IB address is 04. If it is not, you probably should change it to 04. If, for some reason, you want to keep the 5180 set to another address, at the end of step 5 you must change line 80 of the DMA demo program so that "Hp_ib" is given this value rather than 704.

4) Attach the DMA interface cable from the GPIO interface on the 9826 to the 5180. Be sure to check the GPIO interface select code. If it is not 12, at the end of step 5, change the "Gpio=12" statement on line 70 of the demo program so that the proper select code is used. For example, you may want to replace "Gpio=12" with "Gpio=13".

5) Now, turn on the 9826, loading in the BASIC Extensions 2.0 after BASIC 2.0 is ready. Next, load or enter in the DMA demo program. Remember to change the "Hp_ib" or "Gpio" values if that is necessary.

6) Use three coaxial cables to connect the 5181 or oscilloscope to the 5180. Use the 5180 XYZ CAL test pattern as a guide for properly centering the image. (The XYZ CAL test pattern appears when the 5180 is first switched on.)

7) Use another coaxial cable to connect the 3312A (or other signal source) to the A input of the 5180. Trigger the 5180 repeatedly, adjusting the timebase and voltage ranges until the measured waveform can be easily seen on the 5180 display.

8) Finally, press "run" to start the DMA demo program.

Special Function Keys:

At this point, ten special function key labels have appeared on the 9826 screen. These keys allow you to find the current "LENGTH" on the 5180, "READ" a record from the 5180, "PLOT ONE" waveform on the 9826 CRT, "SAVE" a record on the 9826 internal floppy disc "WRT ONE" record to the 5180, and "GET" a saved record from the floppy disc, perform an "FSO READ", transferring many records to the 5180 in one operation, "PLOT ALL" of those records on the 9826 CRT screen, or "WRT ALL" of those records back to the 5180. "EXIT" ends the program.

"LENGTH?":

First, try pressing the key labelled "LENGTH?". If the HP-IB interface is functioning properly, the key label should change to a number giving the current 5180 record length. (If this does not occur, check the HP-IB address and connection.) Within the program, this value is used to determine the number of bytes to be transferred or the number of points to be plotted by the other subroutines. This allows the DMA demo program to be used with whatever 5180 record length you desire.

If you change the 5180 record length, and press key 4 again, the number shown as the label for key 4 will change to match it. This value is also updated automatically every time you press "READ" to transfer a record from the 5180 to the 9826. When the "GET" key is pressed to transfer a DMA record from the internal floppy disc to the memory of the 9826, this value is also updated. This time, however, it is changed to match the length of the record stored on the floppy disc, not the current 5180 record length.

"READ":

Now, try pressing "READ" while watching the front panel of the 5180. If everything is functioning properly, first "Output", then "Fast O" should flash on the 5180 front panel. The "Fast O" signal indicates that a DMA transfer from the 5180 to the 9826 has just taken place. If this does not happen, check the program and the equipment and run through the procedure once more from the beginning. If you still have problems, consult Appendix B (Troubleshooting the DMA Interface).

"PLOT ONE":

To see the transferred waveform, press the "PLOT ONE" key. The waveform you see on the 9826 screen should be identical to the one shown on the 5180 display. You may notice that when you press "PLOT ONE", the key label changes to "END PLOT". If you press this key again before the plot is complete, the plot will be aborted and cleared from the screen. This feature can be helpful when you are working with long records which take a long time to plot.

"SAVE":

To save this waveform on the internal floppy disc, press "SAVE". (Be sure that there is no write protect tab on the floppy disc.) When you are asked to supply a file name, respond with a name like "TEST" or "DATA". After you have typed in the name, press "continue". If the name you supplied is not already being used, the record will be stored on the floppy disc under that name. (The program automatically creates a new file if no file with that name currently exists.)

"WRT ONE":

Now, disconnect the signal input from the 5180 and trigger it again. You should see a flat line on the 5180 display. Now press the "WRT ONE" key. The front panel of the 5180 should flash "Fast I", and the signal recorded earlier should reappear on the 5180 display. You have just written a record from the 9826 to the 5180.

"GET":

Next, trigger the 5180 again, press "READ", and then "PLOT ONE". Now both the 5180 display and the 9826 should show a flat signal. To recover the old waveform, press "GET", type in the file name that you used when you saved the record, and press "continue". Now press "PLOT ONE". The original signal should appear on the 9826 screen. Finally, press "WRT ONE" to return the old signal to the screen of the 5180 display.

"FSO READ"

Now try pressing "FSO READ". The 5180 display screen should show "FSO" for a short time, then go blank again. If "FSO" does not appear, your 5180A may not have the correct firmware revision (1.4 or later) to support the "FSO" DMA mode. Refer to the footnote on the page titled EQUIPMENT FOR DMA DEMO PROGRAM, which tells how to check the software revision number of your instrument.

If the "FSO" message appears, but then does not disappear after a few seconds, the 5180 is probably not triggering on the signal that you are supplying. Assuming that "FSO" has appeared and disappeared as expected, you have just performed and transferred 16 measurements using the "FSO" DMA mode.

One way to accurately measure the time between "FSO" measurements is to take measurements of a very slow triangle wave (perhaps 20 ms peak to peak). Although each record (if measured at the maximum sample rate) will be essentially flat, the difference between subsequent records will give you the "FSO" repetition rate.

"PLOT ALL":

Next, press "PLOT ALL". This allows you to plot all 16 measurements. If you want to abort this plot, simply press "END PLOT". If all of the measurements turned out the same, you probably did not push the internal "auto-stop" switch to the left.

"WRT ALL":

Finally, try "WRT ALL". This subroutine successively writes back to the 5180 all of the records that were captured using "FSO". Since this is a much faster way of displaying these records than "PLOT "ALL", you may find it useful. It also demonstrates the sheer volume of data transferred by the "FSO" subroutine. Naturally, the 5180 only retains the last record that is transferred to it.

How You Can Use This Program:

This program can be used to transfer 5180 records to a floppy disc. With the addition of a printer for the 9826, it can also be used to produce a hard copy of 5180 measurements. The DMA demo also can be used to test a DMA set-up to be sure that the equipment is working properly before an untested program is run on it.

The DMA demo program also provides a good example of how the DMA subroutines are actually implemented in a program. If you decide to modify them for use in your own programs, you may want to try out your modifications with the DMA demo program. By introducing your changes gradually and testing each new version, you will minimize the time you spend debugging your program.

APPENDIX A

Error Recovery Procedures:

This section is intended to aid you in recognizing and recovering from errors that you may encounter as you design DMA programs for your own use. If you only use the subroutines given earlier in this note, and if your hardware functions properly, you should never see these errors. But if you modify the 9826 DMA routines, or have a malfunction in your hardware, this section will help you to recover from any errors you many encounter and understand why they occurred.

Interrupted DMA Transfer:

If your program hangs up in the middle of a DMA transfer, the 5180 will be left in the "Fast I" or "Fast O" mode. One way to recover from this is to type in these two lines from the 9826 keyboard:

CONTROL 12,2;0 (execute)
CONTROL 12,1;1 (execute)

If the select code for the GPIO interface is some number other than 12 (e.g.:13), that number should be substituted for the "12" in the lines above.

Another way to recover from this error is to turn off the 5180 and turn it back on. When you do this an "error 14" message occasionally appears. This can be cleared by turning the instrument off and on once more.

The first of these two recovery procedures is generally preferred. That is because it preserves the data held in the 5180.

If you have tried both of these methods, but still haven't cleared the error, there are a few other recovery procedures that you may want to try. One of these is pressing the autoset button on the 5180. This may work even when turning the 5180 off and on again does not clear the error. The reason for this is that the front panel settings are saved even when the 5180 is turned off. Errors which involve these settings may persist even after power has been shut off. Pressing autoset, however, will often clear these errors.

Sometimes the 9826 will be hung up by the HP-IB or GPIO bus due to an incorrect or incomplete operation. Typically, the first thing you must do is reset the 9826 by pressing "shift" and "pause" at the same time. Next, you should type "CLEAR 7" or "CLEAR 12" and press "execute" to clear the interface that is hanging up the system. (If the GPIO select code is not 12, of course, the appropriate number should be used.)

As a last resort, you can always turn off the 5180, power down the 9826, and start over again. As always, be sure to save a copy of your program before you turn off the 9826.

5180 Enters "Fast O" or "Fast I" When 9826 is Turned Off:

The 5180 will sometimes enter the "Fast O" or "Fast I" mode if the 9826 is switched off while the 5180 is on and the DMA cable is installed. This is because the "TREQ" line (transfer request) which initiates DMA transfers is low-true. When the 9826 is turned off, "TREQ" goes low, and the 5810 enters the DMA transfer mode.

This condition can be avoided by reversing the logic sense of "TREQ". Switch 3 on the GPIO board and switch 3 on the A15 (DMA) board of the 5180 can be reversed to accomplish this. This can be done, but unfortunately, the DMA handshake rate is about 20% lower when "TREQ" is high-true. For this reason you should probably leave "TREQ" low-true.

APPENDIX B

Troubleshooting the DMA Interface:

This section is intended to be used in situations where every time a DMA transfer is attempted, the DMA transfer fails, hanging up the 9826 and the 5180. The error recovery procedures presented in appendix A make it possible to recover from individual failures, but they do not show you how to find the cause. That is the purpose of this section. Before you try the troubleshooting procedures outlined in this appendix, please be sure that:

1) The 5180 functions normally when controlled from the front panel.

2) The 9826 is running normally.

3) HP-IB operations can be performed successfully.

4) The demo program has been loaded or entered into the 9826 exactly as it appears in Appendix F of this note.

5) The switches on the GPIO card and the A15 board of the 5180 are set as shown in figures 1 and 2.

6) The GPIO and DMA interface cards are pushed firmly into their sockets on the 9826 backplane.

7) The DMA cable is connected.

DMA Cable Continuity Check:

If you have checked the above items and still cannot successfully perform a DMA transfer, you may want to check the continuity of the control lines on the DMA cable. At the end of the cable that plugs into the 5180 (the 24 pin end), the control lines are at pins 6, 7, 8, 9, and 10. As shown in Figure 3, these lines are connected to pins 19, 44, 22, 20, and 45 at the other end of the cable.

Problems with other lines will not hang up the DMA transfers. Of course, if there are errors in the data lines, the transferred data will be incorrect, but the subroutine will still run normally. Errors in the data lines can be diagnosed by transferring the ramp signal that is held in the 5180 memory when the 5180 is first powered up. Any differences between the transferred record and a smooth diagonal line indicate problems with the data lines. A close examination of the transferred record will enable you to pinpoint the problem.

Checking the 98620A DMA Board:

To check the DMA board in the 9826:

1) Turn off the 5180 and the 9826.

2) Remove the 98620A DMA Board.

3) Run through the demo procedure discussed earlier in this note.

If the DMA programs run, the DMA board is probably the source of the problem. If they do not, the problem lies elsewhere. When the DMA board is removed, the 9826 performs a "Fast Handshake" transfer rather than a DMA transfer. This kind of transfer is slower than DMA, but it operates similarly.

50 Pin End	24 Pin End	Wire Color	Signal Names (Controller)	(5180A)	
1	12,24	WHT/BLK/GRN	GND	GND	
2	10	WHT/YEL/GRY	DO15	STS	+5V
3	10	WHT/YEL/VIO	DO14	STS	
4	10	WHT/YEL/BLU	DO13	STS	
5	10	WHT/YEL/GRN	DO12	STS	
6	10	WHT/OR/VIO	DO11	STS	
7	10	WHT/OR/BLU	DO10	STS	
8	17	WHT/OR/GRN	DO9	LDA9	DMA data
9	16	WHT/OR/YEL	DO8	LDA8	
10	15	WHT/VIO	DO7	LDA7	
11	14	WHT/BLU	DO6	LDA6	
12	13	WHT/GRN	DO5	LDA5	
13	5	WHT/YEL	DO4	LDA4	
14	4	WHT/OR	DO3	LDA3	
15	3	WHT/RED	DO2	LDA2	
16	2	WHT/BRN	DO1	LDA1	
17	1	WHT/BLK	DO0	LDA0	
18	12,24	WHT	GND	GND	
19	6	WHT/GRY	PCTL	DCTL	control
20	9	WHT/BLK/BRN	I/O	IN/OUT	
21	OPEN	WHT/BLK/RED	PRESET	—	
22	8	WHT/RED/VIO	CTL0	TREQ	
23	OPEN	WHT/RED/GRY	CTL1	—	
24	12,24	WHT/BLK/OR	GND	GND	
25	12,24	Outer Shield	Safety GND	GND	
26	12,24	WHT/BLK/BLU	GND	GND	
27	10	WHT/RED/BLU	DI15	STS	+5V
28	10	WHT/RED/GRN	DI14	STS	
29	10	WHT/RED/YEL	DI13	STS	
30	10	WHT/RED/OR	DI12	STS	
31	10	WHT/BRN/GRN	DI11	STS	
32	10	WHT/BRN/YEL	DI10	STS	
33	17	WHT/BRN/OR	DI9	LDA9	DMA data
34	16	WHT/BRN/RED	DI8	LDA8	
35	15	VIOLET	DI7	LDA7	
36	14	BLU	DI6	LDA6	
37	13	GRN	DI5	LDA5	
38	5	YEL	DI4	LDA4	
39	4	OR	DI3	LDA3	
40	3	RED	DI2	LDA2	
41	2	BRN	DI1	LDA1	
42	1	BLK	DI0	LDA0	
43	12,24	Inner Shield	Safety GND	GND	
44	7	GRY	PFLG	DFLG	control
45	10	WHT/BLK/GRY	PSTS	STS	
46	OPEN	WHT/BRN/GRY	EIR	—	
47	11	WHT/BRN/BLU	STI0	ARM	
48	OPEN	WHT/BRN/VIO	STI1	—	
50	N.C.	WHT/BLK/VIO	N.C.	—	

Figure 3. 10875A DMA Cable Wiring Diagram.

Monitoring DMA Control Lines:

If you suspect that there is a hardware problem, one way to check this out is to monitor the logic levels on the DMA control lines as you step through the DMA transfer subroutine. The easiest way to do this is to put an HP-IB cable connector between the 5180 DMA interface port and the DMA cable itself. (See Figure 4.) You can probe the logic levels at the other end of the HP-IB cable as you step through the program on the 9826. The logic levels that you would see if the transfer were successful are shown in Figure 5.

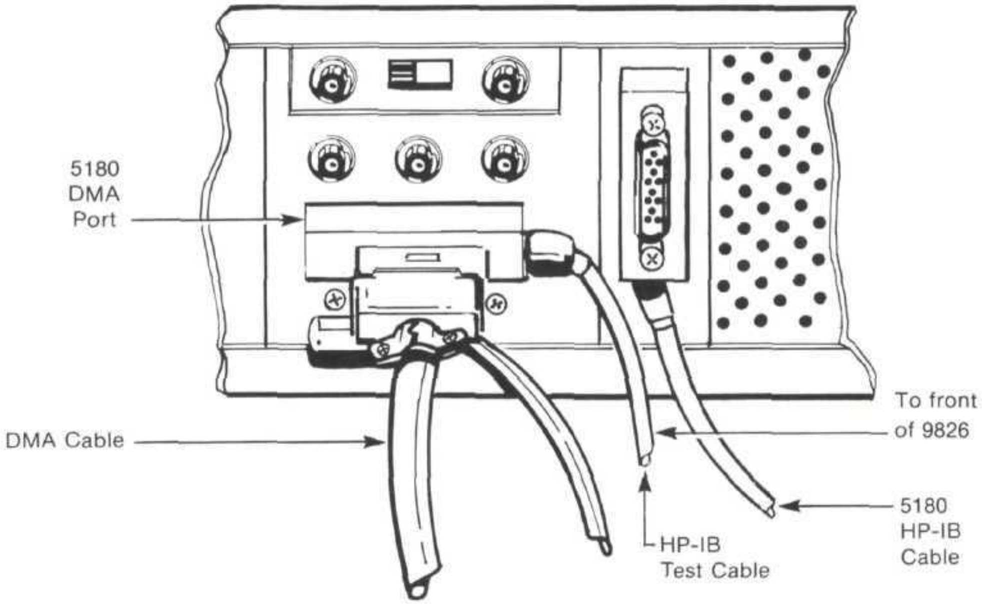


Figure 4. Cable configuration to allow DMA interface control line logic levels to be monitored during a DMA transfer.

The "Fast_o_read" subroutine from the DMA demo program is used to test the DMA function. Adding a "PAUSE" line allows you to step through the program line by line, checking logic levels as you go. The "Fast_o_read" subroutine is listed again below for your convenience:

```
10 Fast_o_read:  GOSUB Find_length        ! Find length of record before transfer.
15     PAUSE !       <--- THIS STATEMENT IS ADDED WHEN YOU TYPE IN "995 PAUSE".
20     CONTROL @Buffer,4;0                ! Set # of bytes in buffer to zero.
30     CONTROL Gpio,0;2                   ! Reset GPIO interface.
40     CONTROL Gpio,3;0                   ! Writing zero clears data registers.
50     STATUS Gpio,3;A                    ! Set IN/OUT to 'in'.
60     CONTROL Gpio,2;1                   ! Set TREQ low.
70     TRANSFER @Gpio TO @Buffer;COUNT Length*2,WAIT ! Transfer record to 9826.
80     CONTROL Gpio,2;0                   ! Set TREQ high.
90     CONTROL Gpio,1;1                   ! Send final handshake.
100    RETURN
```

To begin the test, turn off the 5180 and the 9826 and run through the DMA demo program set-up instructions up to step 8. Now, type in "995 PAUSE" and push "enter". This adds the "PAUSE" instruction to the demo program so that you can step through it. Next, press "run" and then "READ" when the special function keys appear on the screen. You will notice that a program line has appeared at the bottom of the CRT. When you press "step", that line will be executed and the next line will appear. Press "step" repeatedly to step through the program.

In order to probe the voltage levels on the pins at the unattached end of the HP-IB cable, you will need a logic probe, a voltmeter, or an oscilloscope. The pin numbers can be identified from the pin numbers at the four corners of the HP-IB connector. You will probably find it convenient to bring the unattached end of the HP-IB cable to the front of the 9826 so that you can probe logic levels and step through the subroutine while sitting in one place.

"Fast O" DMA read:

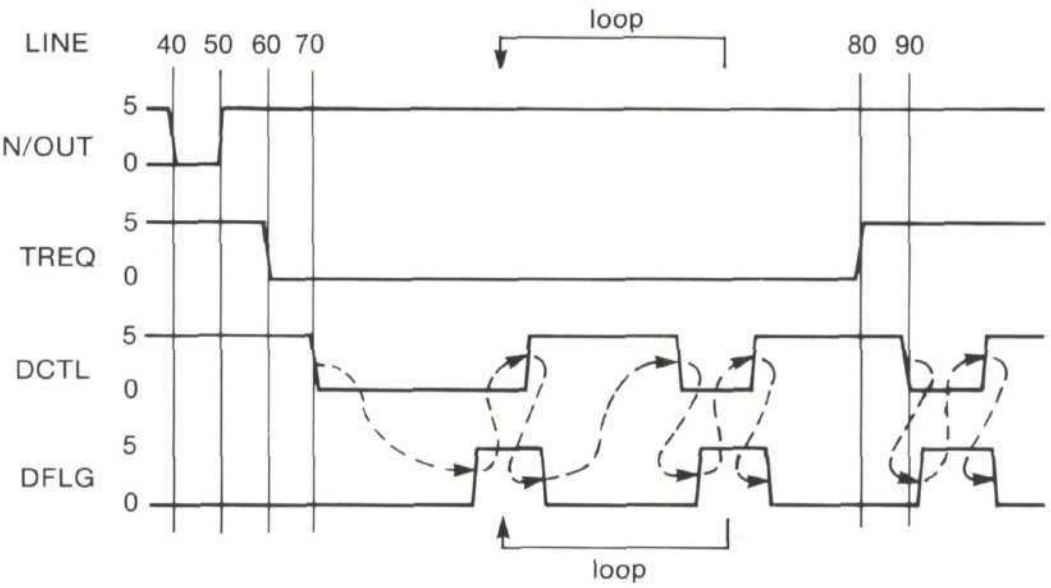


Figure 5. Typical DMA interface control line logic levels during a successful "Fast O" DMA transfer.

When the "Fast_o_read" subroutine begins, the "DCTL" (data control) line at pin 6 should be high (+5 volts), the "DFLG" (data flag) line at pin 7 should be low (0 volts), the "TREQ" (transfer request) at pin 8 line should be high, and the "STS" (status) line at pin 10 should be high. (See Figure 5.)

As you step through "Fast_o_read", when line 40 is executed, the "IN/OUT" line at pin 9 should go low if it isn't already low. Then, at line 50, the "IN/OUT" line should switch high. Next, when line 60 is executed, the "TREQ" line (pin 8) should be pulled low. If, as you step through the DMA demo program, any of the logic levels differ from those described so far, you have a problem with your program, the GPIO card on the 9826, or the DMA cable. Since the 5180 does not interact with the 9826 until line 70, any problems with the 5180 would not have shown up yet.

When line 70 is executed, the "DCTL" line (pin 6) should be pulled low. This initiates a data handshake with the 9826. When the 5180 is ready to send the data word, it indicates this by pulling the "DFLG" line (pin 7) high. Next, when the data word has been accepted, the 9826 switches the "DCTL" line high again, and in response, the "DFLG" line goes low. This process is repeated for each data handshake. (See Figure 6.)

If the DMA transfer fails on line 70, you can tell where the problem is by looking at the logic levels on the "DCTL" and "TREQ" lines. If one line is high and the other is low, the 9826 either has not initiated a data handshake or has not responded to the 5180's handshake. This means that the problem concerns the program, the 9826, or the DMA cable. If both lines are high or both are low, that means the 5180 is not responding, and that the problem is with the 5180 hardware or the way the 5180 has been set up. (Check the DMA configure switches. They should be set as shown in Figure 1.)

When the DMA transfer on line 70 is complete, both "DFLG" (7) and "TREQ" (8) should be low and "DCTL" (6) should be high. When line 80 is executed, the "TREQ" line (8) should be pulled high again. If it is not, the problem is at the 9826 end. At line 90, both "DCTL" (6) and "DFLG" (7) should toggle once as the final handshake is performed.

Whether the problems are with the 5180 or the 9826, be sure to check the switch settings in the 5180 and on the 9826 GPIO interface board, the continuity of the control lines in the DMA cable, and the correctness of the program being run. Also be sure that the current record length on the 5180 is the same the DMA transfer program calls for. If all these items check out, you may have a hardware failure. If you have checked the cable continuity and the program correctness, the most likely causes are the 5180 DMA board (A15) or the GPIO board on the 9826. The test procedure given above should indicate which one needs replacing or service.

APPENDIX C

The 18075 DMA Interface Cable:

The 18075 cable provides two handshake lines, two control lines and 10 data lines between the 5180 and the 9826 (see Figure 3). Although the 5180 provides 16 data lines, only 10 carry useful data. The other 6 lines carry the HP-IB address of the 5180. Since this information is seldom needed, these 6 lines are tied to zero (+5 volts) inside the 10875 cable.

The six HP-IB address lines are tied to zero by connecting them all to the "status" line at pin 10 at the end of the cable that plugs into the 5180. This line always remains at +5 volts. If the HP-IB address is not nulled out in this manner, every transferred data point will be offset by a factor of 1024 times the HP-IB address of the 5180. Since it is time consuming and troublesome to remove this offset once the data has been transferred to the 9826, the HP-IB address is nulled out by the 10875 cable itself.

Building Your Own Cable:

It is possible to make a DMA cable by using an unterminated GPIO interface cable (HP P/N 5061-4209) and a 24-pin Amphenol connector (HP P/N 1251-0293). The 98622A Opt 001 comes with an unterminated GPIO cable. The wiring diagram in Figure 3 shows how to attach the 24-pin connector to the unterminated end of the GPIO cable. Using this diagram, it is also possible to build a cable which does not null out the HP-IB address, if you find that your application requires this information.

APPENDIX D

5180 DMA Theory of Operation:

When the 5180 performs a DMA data transfer, the microprocessor that normally controls the 5180 is temporarily shut off. During this time, the 5180 is controlled by a simple state machine. This section describes the 5180 DMA state machine and how it is controlled by the 9826.

The DMA feature of the 5180 makes use of five control lines: "IN/OUT", "TREQ", "DCTL", "DFLG", and "STS". "IN/OUT" tells whether a transfer will be a DMA write or a DMA read. "TREQ" (transfer request) tells the 5180 to begin and end the DMA transfer. "DCTL" (data control) provides the 9826's side of the data handshake, and toggles once for each data point transferred. "DFLG" (data flag) is the 5180's side of the data handshake. Lastly, "STS" (status) is always high as long as the 5180 is switched on. This is useful for checking that the 5180 is not turned off.

The "DMA configure" switches on board A15 of the 5180 control the logic sense of the four control lines mentioned above. When switch 5 is off (pushed to the right), a low voltage on "IN/OUT" means "in" ("Fast I") and a high voltage means "out" ("Fast O"). When switch 2 is off, "TREQ" is set low to start a DMA transfer and high to end it. When Switches 3 and 4 are off, "DCTL" is low-true and "DFLG" is high-true. (All data lines are low-true.)

"Fast O" DMA Transfer (5180 to 9826):

Figures 6 and 7 show state diagrams and timing diagrams for a "Fast O" DMA transfer. This section describes the steps that the state machine goes through for "Fast O" transfers.

0) At this point, the microprocessor controls the 5180 and the state machine is in state 0 (idle). In order to perform a "Fast O" DMA transfer, the "IN/OUT" line must be high as the transfer begins. (If "IN/OUT" is low, a "Fast I" transfer is performed.) Next, the "TREQ" line and the "DCTL" line must be set low to begin the transfer.

1) The microprocessor detects the low "TREQ" output and sends a signal back to the state machine which allows it to advance. At the next (5 Mhz) clock transition, the state machine advances to state 1, and sends a signal back to the microprocessor ordering it to shut down.

2) While the microprocessor is relinquishing control of the data bus, before it turns off, it sends another signal to the state machine indicating that it is shutting down. After this happens, the state machine advances to state 2 at the next clock transition. In state 2, the top 5 bits of the 10 that make up each data point are loaded into a set of D flip-flops for subsequent transfer to the 9826.

3) After another clock interval (200ns), the state machine advances to state 3, and the lower 5 bits are loaded into a second set of D flip-flops.

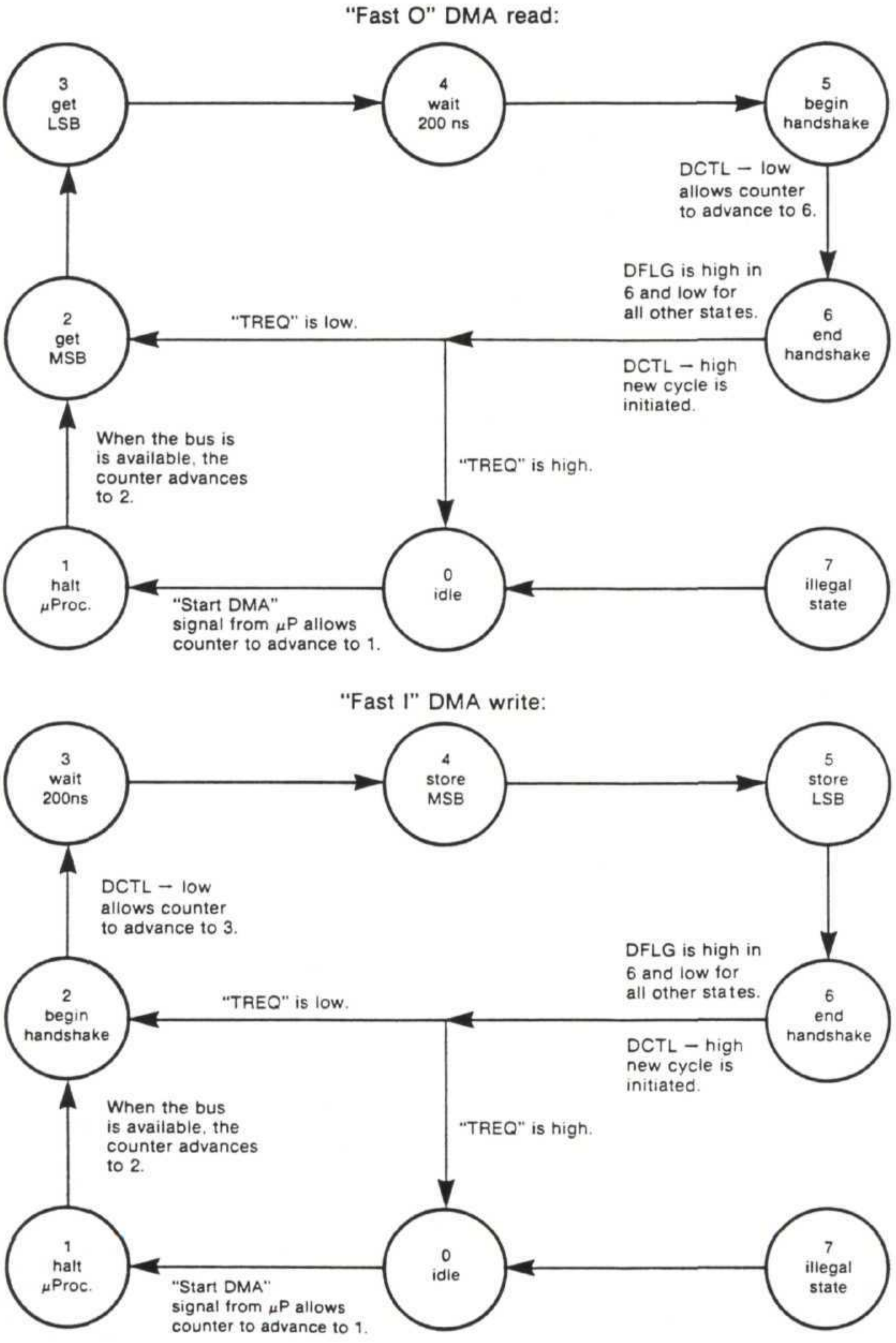
4) State 4 is a 200ns wait.

5) State 5 is another 200ns wait.

6) While the state machine is in state 6, "DFLG" is high (at all other times it is low). This indicates that the data is ready for transfer. The state machine waits in state 6 until "DCTL" is pulled high by the 9826. This indicates that the data has been accepted by the 9826 and that it is ready for more data. When "DCTL" goes high, if "TREQ" is still low, the state machine restarts at state 2. If "DCTL" goes high and "TREQ" is high, the state machine returns to state 0 (idle), the microprocessor restarts, and the DMA transfer is over.

"Fast I" DMA Transfers (9826 to 5180):

When the "IN/OUT" line is set low, a "Fast I" DMA transfer will be initiated. In a "Fast I" transfer, states 2 and 3 are waits while data are stored during states 4 and 5. Otherwise the two transfers operate similarly. Please see Figures 6 and 7 and compare the state and timing diagrams with those for "Fast O" transfers.



"Fast O" DMA read:

3 get LSB
4 wait 200 ns
5 begin handshake
DCTL → low allows counter to advance to 6.
DFLG is high in 6 and low for all other states.
6 end handshake
DCTL → high new cycle is initiated.
"TREQ" is low.
2 get MSB
When the bus is is available, the counter advances to 2.
"TREQ" is high.
1 halt μProc.
0 idle
7 illegal state
"Start DMA" signal from μP allows counter to advance to 1.

"Fast I" DMA write:

3 wait 200ns
4 store MSB
5 store LSB
DCTL → low allows counter to advance to 3.
DFLG is high in 6 and low for all other states.
2 begin handshake
"TREQ" is low.
6 end handshake
DCTL → high new cycle is initiated.
When the bus is available, the counter advances to 2.
"TREQ" is high.
1 halt μProc.
0 idle
7 illegal state
"Start DMA" signal from μP allows counter to advance to 1.

Figure 6. Flow-chart of the 5180 DMA state machine. This flow-chart assumes that the 5180 DMA configuration switches are set according to Figure 1 of this note.

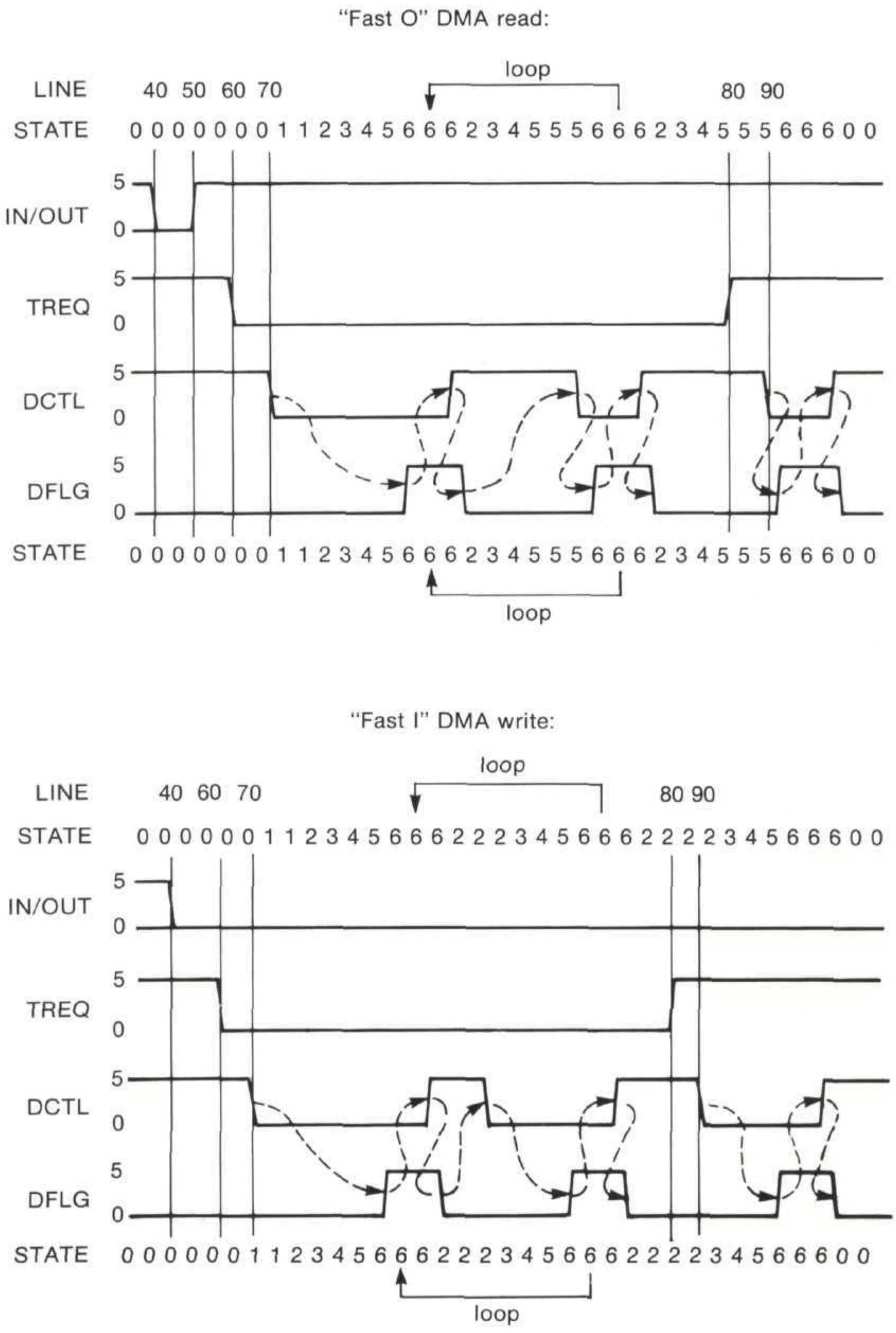

"Fast O" DMA read:
loop
LINE 40 50 60 70 80 90
STATE 0 0 0 0 0 0 0 1 1 2 3 4 5 6 6 6 2 3 4 5 5 5 6 6 6 2 3 4 5 5 5 6 6 6 0 0
IN/OUT 5 0
TREQ 5 0
DCTL 5 0
DFLG 5 0
STATE 0 0 0 0 0 0 0 1 1 2 3 4 5 6 6 6 2 3 4 5 5 5 6 6 6 2 3 4 5 5 5 6 6 6 0 0
loop
"Fast I" DMA write:
loop
LINE 40 60 70 80 90
STATE 0 0 0 0 0 0 1 1 2 3 4 5 6 6 6 2 2 2 3 4 5 6 6 6 2 2 2 2 3 4 5 6 6 6 0 0
IN/OUT 5 0
TREQ 5 0
DCTL 5 0
DFLG 5 0
STATE 0 0 0 0 0 0 1 1 2 3 4 5 6 6 6 2 2 2 3 4 5 6 6 6 2 2 2 2 3 4 5 6 6 6 0 0
loop

Figure 7. Timing diagrams for DMA interface control lines during "Fast O" and "Fast I" DMA transfers. The program lines and the states of the 5180 DMA state machine are also shown. This diagram assumes that the 5180 DMA configuration switches are set according to Figure 1 of this note.

"FSO" Data Transfer (5180 to 9826):

The operations performed by the state machine during an "FSO" DMA transfer are identical to those performed during a "Fast O" transfer. The difference is in the programming of the microprocessor. An "FSO" transfer is initiated by an "FM1" HP-IB command which puts the microprocessor into a loop where all it can do is measure and perform DMA transfers until it is released by an "FM0" command. With a "Fast O" transfer, on the other hand, no HP-IB commands are required, and any operation can preceed or follow the DMA transfer.

Speed Considerations:

The state machine only advances on clock transitions. That means that the 5180 DMA state machine requires at least five clock transitions (one microsecond) to transfer each word of data. If the conditions for advancement to the next state are not met at one of the clock transitions, the state machine must wait another 200ns. If there are no such delays, the data transfer rate will be 1 MHz. Otherwise it will be lower. A typical data transfer rate for the 9826 in BASIC with a 1 meter DMA cable is 600 KHz.

For "Fast O" and "Fast I" data transfers, the data handshakes take up only a small fraction of the time required for the transfer. Most of the time is taken up by the microprocessor as it shuts down at the beginning of the transfer. For a 1K "Fast O" transfer, the data handshakes take up only about 2ms of the 80ms transfer time. The rest is due to overhead in the 5180 as the microprocessor is shut down and the front panel displays are updated.

With the "FSO" DMA mode, however, much of the overhead is eliminated. The data handshake times are still the same, but overhead is cut from about 80ms with the "Fast O" transfer to about 2ms with "FSO". This means that a 1K data transfer using "FSO" typically takes less than 5 ms. With the proper cable (1 meter), plenty of memory in the 9826, a fast-triggering signal to measure, and the proper 5180 front panel settings, the time elapsed between one measurement and the next can be as low as 3.4ms.

APPENDIX E

The 5180 Auto-Stop Switch:

The 5180 is shipped to the customer with the "Auto-stop" switch set to "off". This means that the 5180 remains in a DMA transfer mode until the 9826 signals that the transfer is over by pulling the TREQ (transfer request) control line high and toggling the DCTL (data control) line once. If the 5180 is set to a 1024 point record length, and the 9826 executes a "TRANSFER" statement requesting 2560 points, the 5180 will send two and a half copies of the same record. This will be one continuous transfer. The microprocessor will not regain control of the 5180 until this transfer is complete.

If the auto-stop switch is turned on, DMA transfers will be terminated when the TREQ line is pulled high and the DCTL line is toggled once or when one complete record has been transferred. One implication of this is that if the auto-stop switch is on and only complete 5180 records are being transferred, it is unnecessary to toggle the DCTL line after pulling the TREQ line high at the end of each DMA transfer. Since the DMA transfer is terminated automatically by the auto-stop feature, line 90 of the Fast_o_read and Fast_i_write subroutines can be eliminated if the auto-stop switch is flipped on.

Using the same example as was used earlier, if the 9826 requests 2560 points with the 5180 set to a 1024 point record length, three DMA transfers will result. The 5180 will terminate the first transfer after 1024 points have been sent. Next, the microprocessor will regain control of the 5180, see that still more data is requested, and will initiate a second transfer. Similarly, a third transfer will be initiated after the second full record has been sent. Finally, the 9826 will terminate the third transfer after 512 points by pulling the TREQ line high and toggling the DCTL line once. Note that if the DCTL line had not been toggled, the 5180 would have stayed locked into the DMA transfer mode. Also note that from the point of view of the user and the 9826, it makes no difference whether the auto-stop switch is on or off — the same data is transferred, and the speed is approximately the same.

With FSO transfers, however, the situation is quite different. To perform an "FSO" transfer with the "auto-stop" switched off, each record must be requested individually. If multiple records are requested within one transfer statement with the "auto-stop" switched off, all that you will get is several copies of the same record. By using the auto-stop feature, several different records can be transferred using only one 9826 statement. After each record is sent, the 5180 microprocessor regains control, and takes a new measurement before returning to the DMA transfer mode. This means that hundreds of measurements can be made and transferred to the 9826 in response to a single program statement. Since this eliminates 9826 program execution time delays, it can double or triple the rate at which measurements can be taken by the 5180 and sent to the 9826.

It is important to point out that there is no reason why the "FSO" DMA mode can't be used with the auto-stop switch flipped off. To accomplish this, each DMA record would have to be transferred with a separate call to the "Fast_o_read" subroutine or a similar set of program lines. The problem with this approach is that the rate at which the 1K records can be transferred to the 9826 is decreased from about 200/second to about 100/second, assuming reasonably efficient programming. If your programming is less careful, and the execution time in the loop between DMA transfers becomes excessive, your results could be even worse.

APPENDIX F

DMA Demo Program Listing:

```
10     ! 4/6/83                         DMA DEMO
20     !
30     OPTION BASE 1                    ! First element of array is #1, not #0.
40     INTEGER A                        ! Used for dummy read in 'Dma_read'.
50     INTEGER Buffer(16,1024) BUFFER   ! Adjust to fit available 9826 memory.
60     DIM Error$[80]                   ! Used in 'Set_up_file' to hold ERRM$.
70     Gpio=12                          ! Give select code of GPIO board here.
80     Hp_ib=704                        ! Give 5180 HP-IB address here.
90     Length=1024                      ! To prevent divide by zero errors.
100    Buffer_size=.5*SIZE(Buffer,1)*SIZE(Buffer,2)*2 ! ".5" forces real #.
110    ASSIGN @Gpio TO Gpio:WORD        ! Give I/O path name to interface.
120    ASSIGN @Buffer TO BUFFER Buffer(*);WORD ! Give I/O path name to 'Buffer'.
130    ON KEY 0 LABEL "  READ" GOSUB Fast_o_read
140    ON KEY 1 LABEL "WRT ONE" GOSUB Write_one
150    ON KEY 2 LABEL "PLOT ONE" GOSUB Plot_one
160    ON KEY 3 LABEL "  SAVE" GOSUB Save
170    ON KEY 4 LABEL "LENGTH ?" GOSUB Find_length
180    ON KEY 5 LABEL "FSO READ" GOSUB Fso_read
190    ON KEY 6 LABEL "WRT ALL" GOSUB Write_all
200    ON KEY 7 LABEL "PLOT ALL" GOSUB Plot_all
210    ON KEY 8 LABEL "  GET" GOSUB Get
220    ON KEY 9 LABEL "  EXIT" GOTO Exit
230 Loop: GOTO Loop                     ! Loop until a key is pressed.
240    !
250 Exit: GCLEAR                        ! Clear graphics from CRT screen.
260    STOP
270    !
280 Plot_one: Records=1                 ! Only one record is being plotted.
290    GOSUB Plot
300    RETURN
310    !
320 Plot_all: Records=Buffer_size/Length ! Maximum # of records held in memory.
330    GOSUB Plot                       ! Plot all of these records.
340    RETURN
350    !
360 Plot: CONTROL @Buffer,4;Length
370    REDIM Buffer(Records,Length)     ! Redimension with one row per record.
380    ON KEY 2 LABEL "END PLOT",2 GOTO Graphics_clear ! Interrupt plot.
390    GCLEAR                           ! Clear screen before starting plot.
400    GRAPHICS ON
410    WINDOW 0,Length,0,1024           ! X axis: 1 record.  Y axis: 10 bits.
420    FOR Record=1 TO Records          ! Repeat until all records are plotted.
430    IF Records<>1 THEN DISP Record   ! Display record number for 'Plot_all'.
440    MOVE 1,Buffer(Record,1)          ! Move to first point of record w/o line
450    FOR Point=2 TO Length
460    DRAW Point,Buffer(Record,Point)  ! Draw lines between subsequent points.
470    NEXT Point
480    NEXT Record
490 End_plot: ON KEY 2 LABEL "PLOT ONE" GOSUB Plot_one ! Restore 'PLOT' key.
500    DISP ""                          ! Clear any displayed record number.
510    RETURN
520 Graphics_clear: GCLEAR              ! Plot only cleared if interrupted.
530    GOTO End_plot
540    !
550 Save: GOSUB Set_up_file             ! Get file name and set up for transfer.
560    IF File_name$<>"" THEN GOSUB Create_file ! "" means file found + used.
570    IF File_name$="-INVALID-" THEN RETURN ! If bad file name, then quit.
580    CONTROL @Buffer,4;Length*2       ! Set buffer length to be full.
590    CONTROL @File,6;1
600    TRANSFER @Buffer TO @File;COUNT Length*2,WAIT ! Transfer 2 bytes/sample.
610    RETURN
620    !
630 Get: GOSUB Set_up_file              ! Get file name and set up for transfer.
640    IF File_name$<>"" THEN BEEP
650    IF File_name$<>"" THEN DISP "Sorry, no luck with '"&File_name$&"'."
660    IF File_name$<>"" THEN RETURN    ! If not there or bad file, then quit.
670    CONTROL @Buffer,4;0              ! Set buffer length at 0 (empty).
680    CONTROL @File,6;1
690    TRANSFER @File TO @Buffer;COUNT Length*2,WAIT ! Transfer 2 bytes/sample.
```

```
700   RETURN
710   !
720 Set_up_file: INPUT "Type in new file name and press <continue>:",File_name$
730   ON ERROR GOTO No_file              ! Error if wrong type or doesn't exist.
740   ASSIGN @File TO File_name$:WORD    ! Try to assign I/O path name to file.
750   STATUS @File,4;File_length         ! Register 4 contains the file length.
760   Length=(File_length-50)/2          ! Get length of record on disc.
770   ON KEY 4 LABEL "  "&VAL$(Length) GOSUB Find_length ! Show record length.
780   File_name$=""
790 No_file: OFF ERROR
800   RETURN
810   !
820 Create_file: ON ERROR GOTO F_error ! If wrong type, can't create new file.
830   CREATE BDAT File_name$,1,Length*2+50 ! Error here if file type was wrong.
840   ASSIGN @File TO File_name$:WORD    ! Set up I/O path name for created file.
850   GOTO No_error
860 F_error: File_name$="-INVALID-"      ! Indicates that file name was invalid.
870   Error$=ERRM$                       ! Can't take a substring of "ERRM$."
880   DISP "    "&Error$[1,8]&".  "&Error$[POS(Error$,"  ")+2]&"."
890   BEEP                               ! Alert user to error.
900 No_error: OFF ERROR
910   RETURN
920   !
930 Find_length: OUTPUT Hp_ib;"OM"       ! Request memory settings string.
940   ENTER Hp_ib:A$                     ! Accept memory settings string.
950   Length=VAL(A$[POS(A$,"le")+2;7]) ! Get length of record from 'OM' string.
960   ON KEY 4 LABEL "  "&VAL$(Length) GOSUB Find_length ! Show record length.
970   RETURN
980   !
990 Fast_o_read: GOSUB Find_length       ! Find length of record before transfer.
1000  CONTROL @Buffer,4;0                ! Set # of bytes in buffer to zero.
1010  CONTROL Gpio,0;2                   ! Reset GPIO interface.
1020  CONTROL Gpio,3;0                   ! Writing zero clears data registers.
1030  STATUS Gpio,3;A                    ! Set IN/OUT to 'in'.
1040  CONTROL Gpio,2;1                   ! Set TREQ low.
1050  TRANSFER @Gpio TO @Buffer;COUNT Length*2,WAIT ! Transfer record to 9826.
1060  CONTROL Gpio,2;0                   ! Set TREQ high.
1070  CONTROL Gpio,1;1                   ! Send final handshake.
1080  RETURN
1090  !
1100 Fso_read:GOSUB Find_length          ! Find length of record before transfer.
1110  OUTPUT Hp_ib;"FM1"                 ! Turn on "FSO" mode with HP-IB command.
1120  CONTROL @Buffer,4;0                ! Set # of bytes in buffer to zero.
1130  CONTROL Gpio,0;2                   ! Reset GPIO interface.
1140  CONTROL Gpio,3;0                   ! Writing zero clears data registers.
1150  STATUS Gpio,3;A                    ! Set IN/OUT to 'in'.
1160  CONTROL Gpio,2;1                   ! Set TREQ low.
1170  TRANSFER @Gpio TO @Buffer;WAIT     ! Fill buffer with DMA records from 5180.
1180  CONTROL Gpio,2;0                   ! Set TREQ high.
1190  CONTROL Gpio,1;1                   ! Send final handshake.
1200  OUTPUT Hp_ib;"FM0"                 ! Turn off "FSO" mode with HP-IB command.
1210  RETURN
1220  !
1230 Write_one: N=1                      ! 'N' has other values in 'Write_all'.
1240  CONTROL @Buffer,4;Length*2         ! Tell 9826 that buffer holds 1 record.
1250  GOSUB Fast_i_write                 ! Simply write one record (not several).
1260  RETURN
1270  !
1280 Write_all:  CONTROL @Buffer,4;Buffer_size*2 ! Tell 9826:  buffer is full.
1290  FOR N=1 TO Buffer_size/Length      ! # of records fitting into 'Buffer(*)'.
1300  DISP N                             ! Give # of the record being transferred.
1310  GOSUB Fast_i_write                 ! Write the 'Nth' record to the 5180.
1320  NEXT N
1330  DISP ""                            ! Clear last displayed number.
1340  RETURN
1350  !
1360 Fast_i_write: CONTROL Gpio,0;2      ! Reset GPIO interface.
1370  CONTROL Gpio,3;0                   ! Writing zero clears data registers.
1380  CONTROL Gpio,2;1                   ! Set TREQ low.
1390  TRANSFER @Buffer TO @Gpio;COUNT Length*2,WAIT ! Transfer record to 9826.
1400  CONTROL Gpio,2;0                   ! Set TREQ high.
1410  CONTROL Gpio,1;1                   ! Send final handshake.
1420  RETURN
1430  END
```

HEWLETT PACKARD

For more information, call your local HP Sales Office or nearest Regional Office: **Eastern** (201) 265-5000; **Midwestern** (312) 255-9800; **Southern** (404) 955-1500; **Western** (213) 970-7500; **Canadian** (416) 678-9430. Ask the operator for instrument sales. Or write Hewlett-Packard, 1501 Page Mill Road, Palo Alto, CA 94304. **In Europe**: Hewlett-Packard S.A., 7, rue du Bois-du-Lan, P.O. Box, CH 1217 Meyrin 2, Geneva, Switzerland. **In Japan**: Yokogawa-Hewlett-Packard Ltd., 29-21, Takaido-Higashi 3-chome, Suginami-ku, Tokyo 168.

02-5952-7710 JULY 1983 PRINTED IN U.S.A.