AN-321

Application Note

The acquisition and recovery of analog signals in a M6800 data processing system

Some practical examples of the interfacing required in analog data processing with microprocessors are discussed. A multi-channel process control data acquisition module and the graphic display of digital data on a standard X-Y chart recorder are presented as typical interface units.

3

The acquisition and recovery of analog signals in a M6800 data processing system

INTRODUCTION

The fundamental circuits used in processing analog signals by conventional methods include the operations of filtering and detection. These are, in fact, simplified electronic circuit realisations for the operations of convolution and correlation. The introduction of the digital computer into the realm of signal processing permits these mathematical operations to be performed in practice. Once the analog signal has been transformed into digital format, any required sequence of operations for processing the signal may be performed, simply by programming the computer accordingly.

On-line computation can now be envisaged, since computers are available as circuit components. However, the limitation remains, that total calculation time must be more rapid than the rate of data acquisition. The relatively slow speed of performing digital arithmetic operations therefore limits the on-line microprocessor system to applications in the audio and subsonic area (e.g. physiological signals, vibration analysis, servo-motor controllers) at present.

Since data acquisition can usually be performed faster than the rate of calculation, the techniques of off-line and batch processing can be advantageously employed in treating higher frequency signals. To illustrate this facility, we consider here the problem of data acquisition to meet the requirements of a process control application. The time delay and/or change of time scale implied in off-line processing often necessitates visual displays (e.g. CRT, chart recorder) of computed signal parameters over defined time windows. The example of such a graphic display using an X-Y recorder is discussed later.

In considering these particular examples, the basic features of sampled data processing must be understood. A brief description is included in the Appendix.

DATA ACQUISITION

3

The particular example taken here concerns multi-channel data acquisition for an on-line process control application using an M6800 system. A large number of slowly varying analog parameters have to be made available as inputs for the control algorithm being evaluated by the digital microprocessor. The control algorithm may call for any of the analog variables to be available in digital representation with an access time that is compatible with the response time of the control loop (including the calculation time of the processor). This restriction makes it impossible to use conventional data logging[1] owing to the long conversion time of an integrating analog-to digital converter. The requirement here is that an analog variable should be available for digital data manipulation by the M6800, with a minimum time delay after the appropriate software instruction. The solution adopted is to use successive approximation analog-to-digital conversion with a digitally controlled channel selector.

The specifications taken for a data acquisition module are the following:

- 8 single ended analog input channels (extension possible to 16).
- Input voltage excursion +5 to −5 Volts.
- Conversion accuracy 0.5‰ (10 bits with sign).
- Digital output compatible with M6800 bus.
- Conversion time less than 100 microseconds after software "request" for any analog variable (channel).

The input voltage variations are assumed to be less than ½ LSB (2,5 millivolt) during the inter-sampling period (100 microseconds). This obviates the need for a "sample and hold" circuit at the input to the converter.

The multi-channel analog to digital converter is to be interfaced with the M6800 bus structure. Several combinations using the MC6820, Peripheral Interface Adapter, have been described elsewhere[2]. However the organisation of hardware/software to achieve the specifications defined above, precludes the use of the MC6820, owing to access time limitations. This leads to the adoption of a multiplexed successive approximation register converter[3]

The interface to the M6800 system may still be realiseo a number of different ways, three of which will now be discussed in more detail.

(a) Direct Memory Access (DMA) is a means of entering the digital data from the converter directly into the M6800 system RAM without going through the MPU registers (for a detailed description, see the M6800 Applications Manual). In this case, the A/D conversion is free-running, converting the analog channels in a pre-determined order. (hardwired). Whenever a channel of digital data is ready for transfer into RAM via the data bus, the assigned address corresponding to that data channel must be generated externally for the MPU address bus.

The M6800 bus structure can be forced into 3-state (high impedance) condition, in order to accept the data, by using the GO/HALT line. Alternatively, the TSC line may be used and the clock generator stretched to permit "cycle stealing". More efficient still is to use rapid 3-state buffers to isolate the RAM (which must also have rapid access time for this application) from the rest of the MPU system. During clock cycle phase Ø1, the data may be written into the RAM, without affecting the MPU in any way.

The information transferred into the RAM by DMA will be the latest converted value, so is always available to the MPU by a standard software read command (LDA). This technique has the disadvantage that the analog multiplexing is not under MPU control. The rate of updating the RAM for one particular channel (analog bandwith) depends then on the rate and sequence of scanning the channels.

[1] Data acquisition networks with NMOS and CMOS - Motorola Application Note AN-770.

[2] Analog to digital conversion techniques with M6800 Microprocessor System – Motorola Application Note AN-757.

[3] Sucessive approximation A/D conversion - Motorola Application Note AN-716.

(b) Interrupt (IRQ) routines can be used to transfer a completed analog-to-digital conversion into the MPU registers. The free-running converter signals the end of the conversion (EOC) to the M6800 via the IRQ line. An interrupt service routine is used to store the result in a defined RAM location before returning to the original program. The updated value can be accessed from the RAM by a software read command (LDA). However when using interrupts, it should be remembered that the time between the IRQ occuring and access to the interrupt routine may be as long as 23 μs. Returning to the original program will take a further 10 μs. So much permanent dead time will seriously affect the computational speed and significantly reduce the response time, if the system is part of a control loop.

(c) Software command of the data acquisition system may be realised if the analog channel selection/analog-to-digital converter is hardwire coded so that it can be addressed directly by the MPU bus. The software request to read a desired analog channel will connect that analog channel to the A/D converter and trigger the start of conversion. When conversion is complete the result is transferred into the accumulator and computation proceeds.
Although some dead time is involved in waiting for the conversion to be performed. this only occurs while the software request for analog data is being serviced. In a real system, such requests will probably occur relatively infrequently at irregular intervals (compared with the A/D conversion time), so the loss of computing time may not be significant.

In view of the above discussion, the last technique as chosen for this application.
The requirement for working to an accuracy of 0.5‰ implies the use of a 10 bit successive approximation register converter. For convenience, two consecutive MPU data bytes are used, since the data bus in only 8 bits wide. The timing sequence of the latched bits of a successive approximation converter is well defined by the clock, so it is possible to handle the two most significant bits in one MPU data byte while the remaining eight bits are being determined by the converter. These latter 8 bits are then transferred as one byte into the MPU system. The "pipeline" organisation of the system is shown in figure 1.

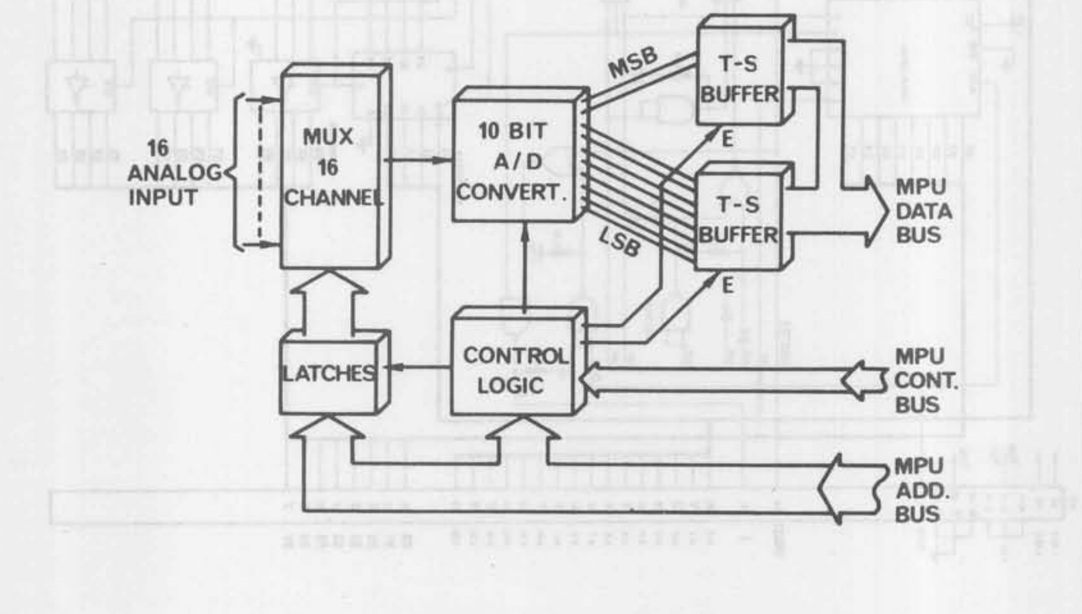

FIGURE 1 – DATA ACQUISITION BY SOFTWARE COMMAND.

To illustrate the use of this technique with a real example, an 8-channel data acquisition system has been built on an EXORciser card. The A/D converter (10 bit with sign) is not synchronous with the MPU clock. The circuit diagram of the board is shown in figure 2. This circuit permits the micro-computer (not shown) to request, by means of LDA software instructions, the two byte digital value of a desired analog channel. The result is made available in the A (LSB) and B (MSB) accumulators and also is stored in defined locations in the MCM6810 (random access memory) on the card.

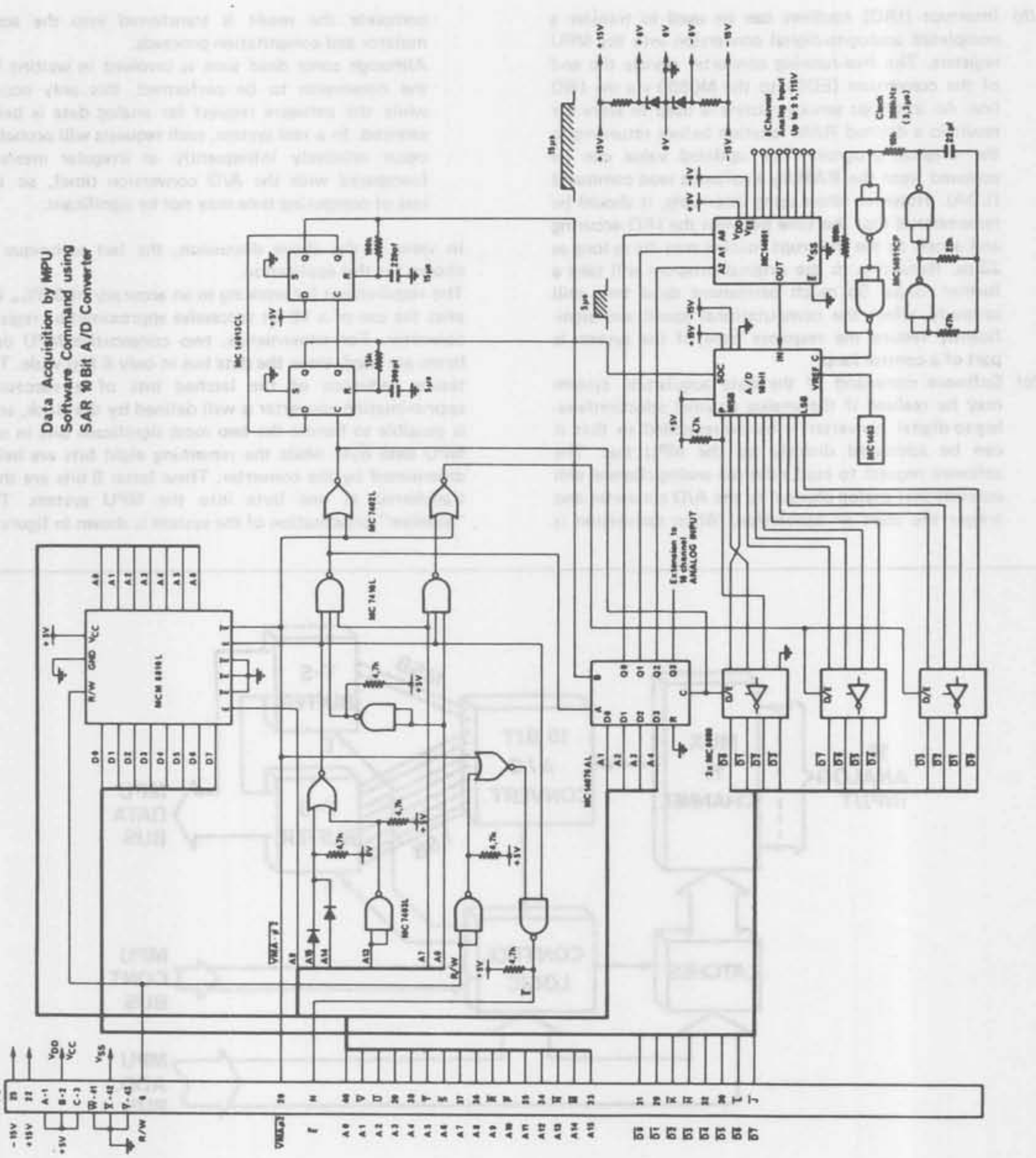

FIGURE 2 – BLOCK DIAGRAM

3

In this example, the following address assignments were used:

CHANNEL 1	2080/2081
CHANNEL 2	2082/2083
.	.
.	.
.	.
CHANNEL 8	208E/208F
RAM	2100/217F

A software instruction to read (LDA) any channel address is decoded on the card and starts the A/D converter (SOC line) with the analog multiplexer latched to the appropriate channel.

The timing diagram relevant to the sequence of operations is shown in figure 3. The conversion is completed approximately 55 µs after the first read instruction initiates the converter. An example of the use of this data acquisition card is in storing (in the RAM) the values of the 8 analog channels in rapid sequence. The software for this is shown here where the program is intended for use as a subroutine, with retrieval of index and accumulators included.

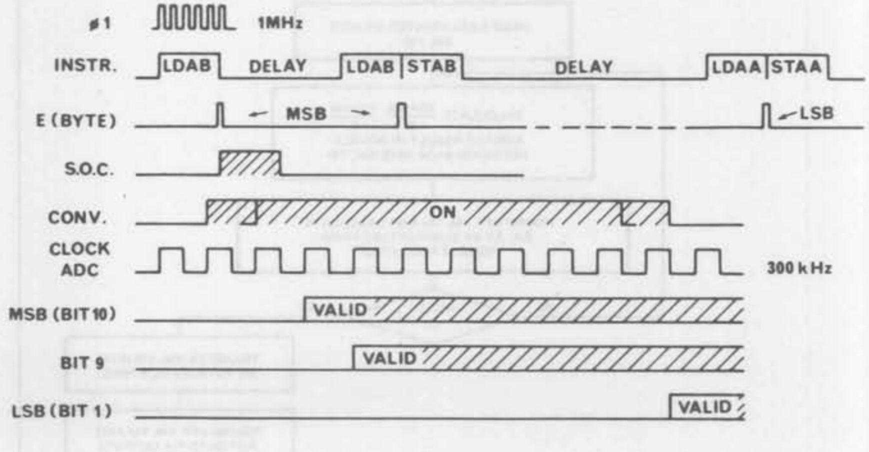

FIGURE 3 – TIMING DIAGRAM OF ONE CONVERSION

```
START   SEI                    Interrupt request masked
        PSH   A                Save ACC.A
        PSH   B                Save ACC.B
        STX         $2120      Save Index Register
        LDX         #$2080
        LDA   B     00,X       Start of Conversion (SOC)
        PSH   A                Delay 4 µs
        PUL   A                Delay 4 µs
LOOP1   NOP                    Delay 2 µs
        LDA   B     00,X       MSB BYTE Valid, SOC locked
        STA   B     $80,X      Final Data Memory (RAM)
        INX                    Prepare LSB BYTE Address
        PSH   A                Delay 4 µs
        PUL   A                Delay 4 µs
        PSH   A                Delay 4 µs
        PUL   A                Delay 4 µs
        LDA   A     00,X       LSB BYTE Valid
        STA   A     $80,X      Final Data Memory (RAM + 1)
        INX                    Prepare next channel input
        LDA   B     00,X       Next channel SOC
        CPX         #$2090     Need 16 BYTE for 8 channel
        BNE         LOOP1
        PUL   B                Restore ACC.B
        PUL   A                Restore ACC.A
        LDX         $2120      Restore Index Register
        NOP
        CLI                    Clear Interrupt Mask
        RTS                    Return
END
```

Note: that if a 16 channel multiplexer is used, then the CPX instruction must be raised by 16 more bytes (to $20A0).

3

3

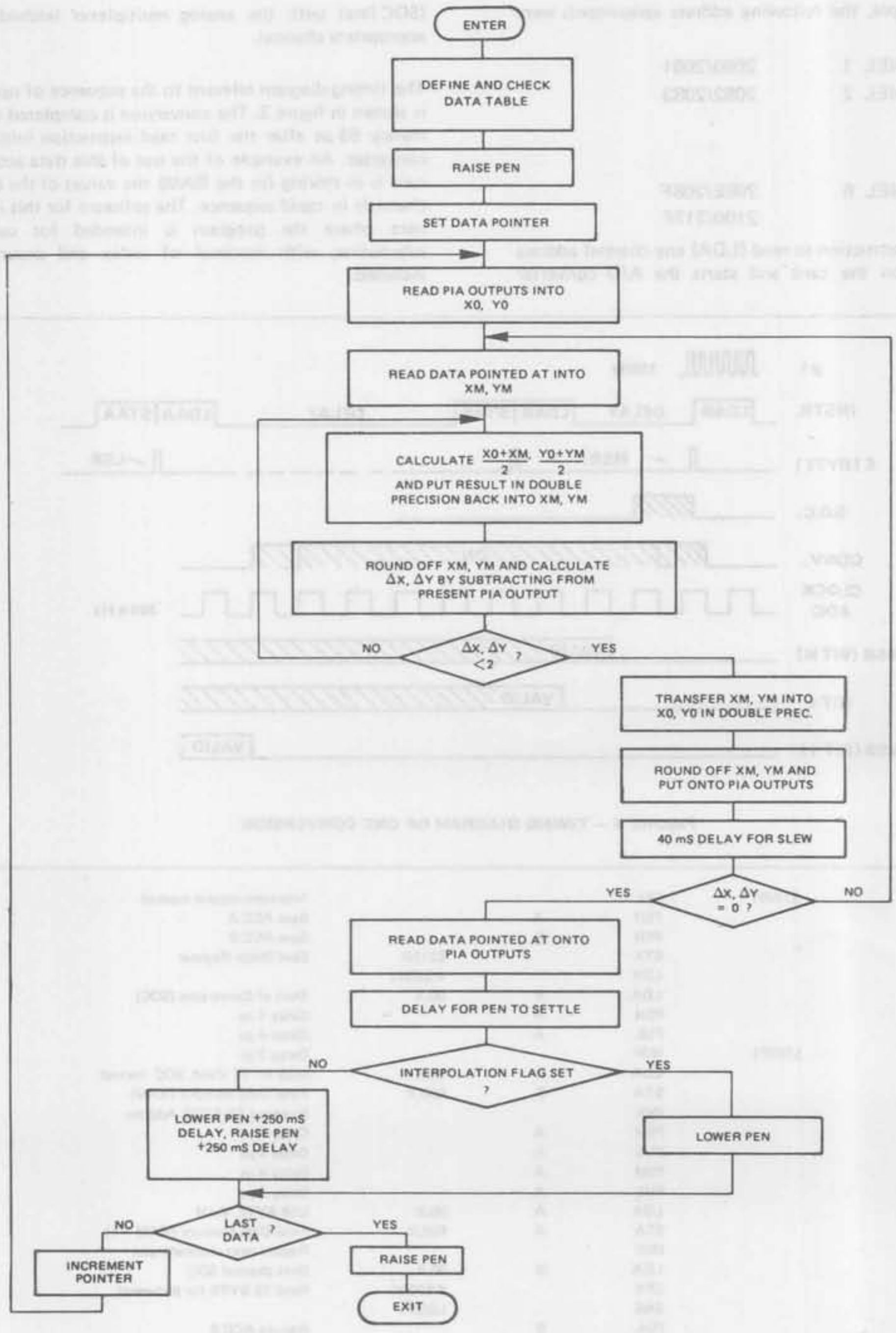

FIGURE 4 – CHART PLOTTER FLOW CHART

A distinct advantage of the system just described is that, once the digitisation process has been initiated, it is possible to predict exactly when the data will become available. The microcomputer could therefore be made to do useful work (not just stack shifting as shown in this example) during this time interval.

DISPLAY FOR DATA/GRAPHICS

Although acquisition, data manipulation, and output are now purely digital, a display of some parameters in analog form is very often needed. As pointed out in the Appendix, provided the data is defined at regular time intervals, respecting the Nyquist criterion, then complete signal recovery is possible by convoluting the digital data with the appropriate sinc function. Examining the more general case though, where the time intervals are not regular, which may occur when the data held in the memory represents discontinuous signals of graphic information, then the above procedure cannot be used to generate a visual display. The general case will now be treated in more detail, since it is also applicable to the specific case of signal display where the signal has been stored again after the convolution process has been performed. The two most common displays are the CRT (usually for on-line) or X-Y chart plotter where off-line hard-copies are required. This latter display will now be considered in more detail, since it is easily compatible with the M6800 as far as speed is concerned. The electro-mechanical parts of X-Y plotters are usually controlled either by stepper motors (digital pulse train inputs) or analog servo-motors (analog voltage inputs). In either case a dedicated interface circuit is usually employed, which relieves the host computer of the inefficient task of generating the slowly varying signals required to drive the pen correctly. A digital differential analyser (DDA) may be used, which generates incremental X and Y control signals based on the co-ordinates and graphic "primitive" information calculated by the host computer. Owing to the slow writing speed of standard X-Y recorders, a microcomputer can provide, at low cost, a viable general purpose interface which can easily calculate and generate the pen drive signals itself. The classical method would be to use it to calculate the characteristic co-efficients of the chosen "primitive" and evaluate the pen coordinates as X or Y (as appropriate) are slowly incremented. This procedure involve the formal operations of multiplication and division. For simplicity, the graphic "primitives" used here are restricted to straight line segments between points, so a less formal algorithm of "continuous averaging" is used to generate the pen movements, as discussed below.

The X, Y co-ordinates of the mid-point between the two points to be joined are calculated by subtraction and a right shift (in double precision). If the pen would have to move more than 1 bit to reach this point, the mid-point of the actual pen position and the previous average is calculated. This procedure is repeated until the difference is only one bit; the pen is then advanced to the calculated point. Averaging starts again, to obtain the next point; this continues repetitively until the pen is within one bit of the defined X, Y co-ordinates held in memory. The pen is finally advanced to this point, finishing the straight line traverse between the two sets of co-ordinates.

The flow chart describing this operation is shown here in figure 4.

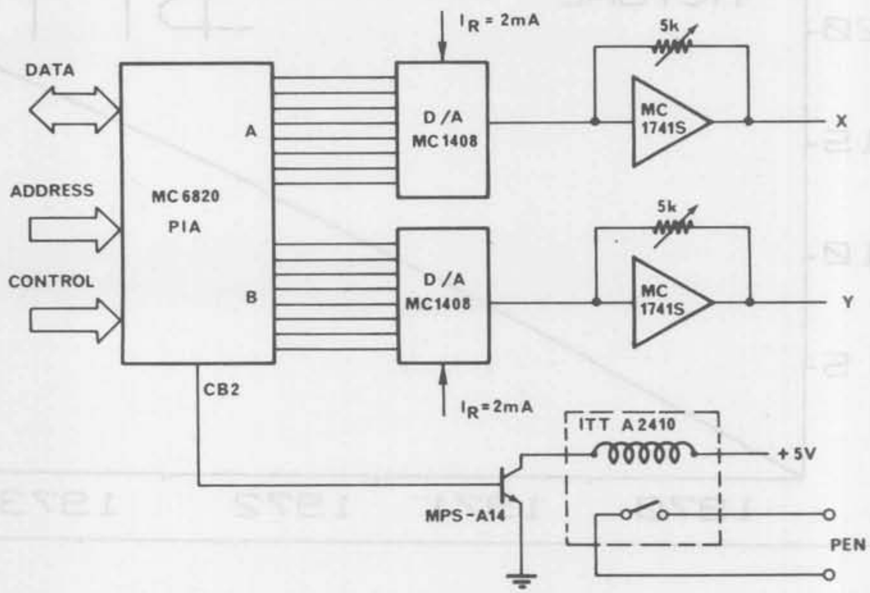

FIGURE 5 – GRAPHIC PLOTTER INTERFACE

3

A typical display unit is a Hewlett-Packard 7035B chart recorder. For this plotter, the precision of positioning the pen (±0.2%) is of the same order as the resolution of the 8 bit data bus. The display unit therefore only requires that two (X and Y) D/A converters (MC1408) be interfaced to the MPU bus via a PIA (MC6820). The circuit is shown in figure 5 where the peripheral control line, CB2, is also used to provide the pen-lift function. The gain of the X and Y channels may be adjusted independantly so as to obtain graphics suited to the particular paper size used.

If more than 8 bit precision is required (i.e. a more accurate plotter), double precision (2-byte) data becomes necessary. Both hardware and software now become more complex, but the same principle is applied in the interpolation for the pen movement.

A program for use with the HP7035B recorder has been written to allow one of three modes of operation to be selected by the user.

OPTION I — Plots straight lines between adjacent points specified by the co-ordinates held in memory. The data is unsigned and assumed to be in memory as a table; the lowest address specifies the first Y co-ordinate and the highest, the last X co-ordinate. Pen movement is slowed to 25 mm/sec. by software delay in order to avoid slewing errors.

OPTION P — Plots individual co-ordinate points with the pen lifted between points while traversing. Otherwise, identical to Option I.

OPTION T — Plots alphanumeric text as a series of straight lines. Part of the memory table contains the co-ordinates for alphanumerics. The appropriate addresses are called up by the ASCII code for the character required. Three possible sizes of character may be specifed by the user as well as the starting point for every line of text.

These three options are illustrated by the plot shown here in figure 6.

The basic program consists of about 700 bytes, while the alphanumeric ASCII character look-up table occupies another 700 memory bytes.

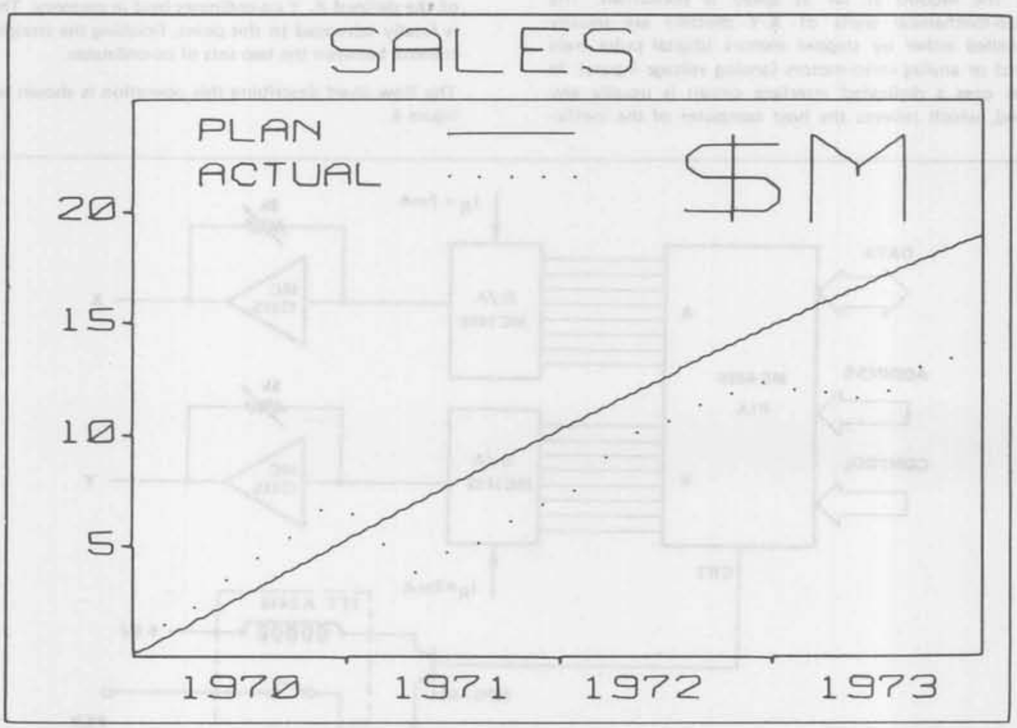

FIGURE 6

CONCLUSION

With suitable A/D and D/A conversion circuits, the microcomputer can be considered as a general purpose circuit component in the same sense as an operational amplifier. The defined task is specified by external components for an operational amplifier, while this function is performed by the software program in the case of the microcomputer. When processing analog signals, the advantages of using digital computation are in the stability and high accuracy achieved; the disadvantage is the speed restriction imposed in handling the sampled-data signals, so limiting the operational band-width. However, the impact of the microcomputer will be felt most in areas where the procedures used call for long storage time and for versatile algorithms which are adaptable in real time. In the low-frequency part of the spectrum, microcomputer calculations can be performed sufficiently rapidly to implement processing in real time. Higher frequency signals however can be treated off-line requiring an understanding of efficient data acquisition and display methods. The two examples described in this note are intended to be practical without resorting to discussions that may be too generalised.

APPENDIX

SAMPLED DATA SYSTEM CONSIDERATIONS

The input to a digital signal processor is a series of discrete quantized data points. In order that a continuously varying analog signal be compatible with the processor, an interface is needed which will make available the quantized values (with the required precision) at accurately known time intervals. For the purposes of this discussion it will be assumed that the quantisation is multi-level (binary coding) and that the time intervals are all identical. These are real restrictions, since a number of other useful representations do exist. However, accepting the above assumption, leads to the simplification that the interface has two functions, which can be considered separately. One is the data sampling function (sample and hold) while the other is the analog to digital converter. Both of these are well known to electronic engineers and the various possible implementations will not be discussed further here.

The decision to use regular sampling intervals imposes constraints on the analog input signal which is to be processed. From basic sampling theory developed by Shannon, any analog variable can be completely specified as a discrete time series, provided that sampling is performed at a frequency higher than the "Nyquist frequency". That is, above twice the frequency of the highest frequency component present in the analog signal. If the analog signal does not conform to this limitation then an over-lapping phenomenon known as "aliasing" can occur. In practice, the analog input signal is band-limited by using a suitable low-pass filter to attenuate the higher frequency components, so avoiding digitising incorrect values due to under-sampling.

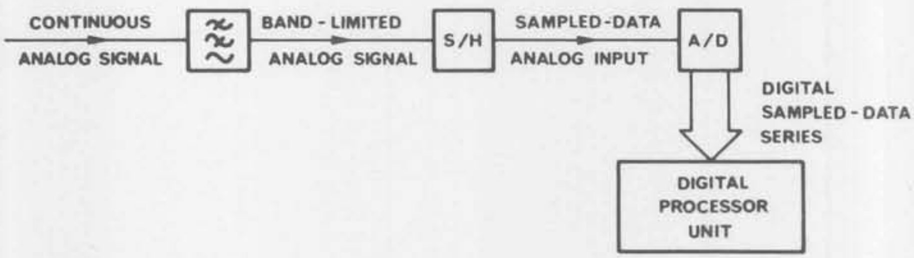

FIGURE 7 – SAMPLED DATA INPUT

3

The above procedure, illustrated in figure 7, describes a simple technique for entering data into a digital processor. However, the problems encountered in the inverse procedure for the reconstitution of analog signals from the digital processor output are not so widely appreciated. Continuous analog signals may be generated from a discrete time series by interpolation, the inverse procedure to sampling. Mathematically this necessitates the convolution of the output time series with the "sinc" function[4]. The convolution may be performed by an ideal low-pass linear phase analog filter, as shown in figure 8.

Such an ideal filter is not easy to realise; the usually adopted approximation is to use a first or second order low pass filter. This has to be designed so as to limit the interpolation distortion due to phase non-linearity to the same order as the precision of the D/A converter which preceeds it, a relatively simple task.

If the output signal is to be presented slowly enough (as when using off-line display) the convolution may be carried out in digital form by the processor itself. This necessitates the evaluation of the convolution process being performed faster than the slew-rate of the display device. Such a system, eliminating the analog low-pass filter, is a realistic possibility for the case of a chart recorder display.

The particular arithmetic manipulations carried out on the digital time series between digitisation and analog recovery will not be considered here. The actual processes are defined by algorithms usually in the time domain (e.g. Z-transforms) which are realized entirely by the software program, thereby giving the versatility and on-line adaptability which characterises this form of signal processing.

[4] $\text{sinc}\, x \triangleq \dfrac{\sin \pi x}{\pi x}$

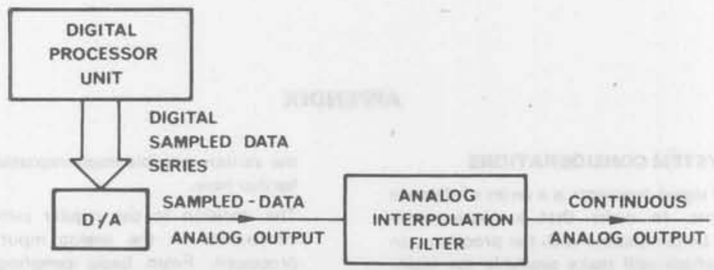

FIGURE 8 – SAMPLED DATA OUTPUT