THE APPLICATION OF MICROPROCESSORS IN ELECTROCHEMISTRY

Lubomír Pospíšil[a] and Miroslav Štefl[b]

[a] *J. Heyrovský Institute of Physical Chemistry and Electrochemistry, Czechoslovak Academy of Sciences, 118 40 Prague and*
[b] *Development Works, Czechoslovak Academy of Sciences, Prague*

Received December 28th, 1981

A microcomputer based on Intel SDK 80 and an arithmetic microprocessor is described. The design and interfacing was made in such a way in order to fulfill most of the electrochemical measurements. The application to the staircase voltammetry and FFT admittance measurements is described.

The digital on-line computer enhanced remarkably the effectivity of research not only by the direct data treatement and/or reduction but also by the possibility of a qualitatively novel approach to the implementation of various experimental techniques[1,2]. However, in many laboratories the price of a minicomputer is still the main obstacle in its application. The introduction of microprocessors (μP) as the building unite into the instrumentation seems to be a turn-point in this respect[3-6]. Microprocessor-controlled commertial instrument usually does not permit to a user to change the instrument soft-ware and to use the device for a non-standard research experiments.

In this communication will be described a microcomputer (μC) based on two μP's which was designed in order to fulfil most of the demands of the electrochemical research.

The system was designed with the aim to furnish the following functions: *1*) Sufficiently fast and accurate analogue output for various electrochemical relaxation techniques. *2*) Fast digital data recording. *3*) Programmable experiment timing. *4*) Simple interfacing to locally available peripherals. *5*) Possibility of arithmetic operations for a simple data treatment. *6*) Possibility of writting and debugging user's programs.

It is therefore assumed that the μC will controll the input of a potentiostat and will read the cell response from a suitable current-to-voltage convertor.

RESULTS

Microcomputer

Our μC system shown schematically in Fig. 1 is built around the Intel 8080 central processing unit (CPU) equipped with a standard set of LSI supporting integrated

circuits including[3,7,8] 8224 clock generator(CG); 8228 system controller/bus driver (SCBD); 8212 bi-directional latch/buffer (AB); 8205 address decoder (D); 8255 programmable input/output interface (PIO); 8251 programmable serial input/output port (USART).

All components listed above (most of them in multiple quantities) were used in the form of a small "system design kit" supplied by Intel Co under the trade name SDK-80. SDK-80 was assembled in such a way that the hold and wait states of the CPU were disabled and system bus was enabled at all times. However, the system SDK-80 had to be slightly modified in order to allow for the necessary expansion fulfilling our purposes. First of all the manufacturer's soft-ware supplied with the SDK-80 was abandoned because it was strictly bounded to the serial I/O communica-

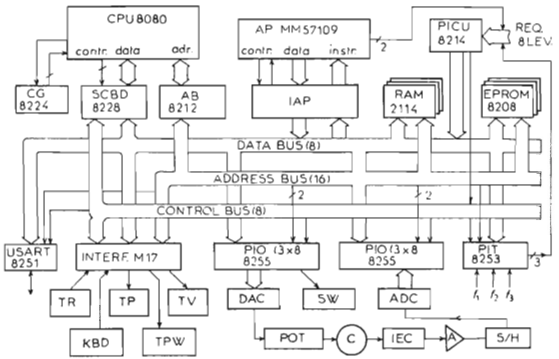

FIG. 1

The block diagram of the microcomputer for electrochemical measurements. The LSI circuits are: CPU central processing unit, AP arithmetic processor, CG clock generator, SCBD system controller/bus driver, AB address buffer, PICU programmable interrupt control unit, RAM random access memory, EPROM erasable programmable read-only memory, USART synchronous/asynchronous receiver/transmitter, PIO programmable input/output, PIT programmable interval/timer. The hybrid circuits are: ADC analogue-to-digital converter, DAC digital-to-analogue converter and SH sample/hold circuit. IAP interface for the arithmetic microprocessor, M–17 interface box for peripherals. Peripheral instruments are: TR paper-tape reader, TP paper-tape puncher, TV TV-display, KBD keyboard (full ASCII), TPW type-writer, and SW solid-state switches. The electrochemical instruments are: POT potentiostat with the IEC current/voltage converter, amplifier A and the electrochemical cell C

tion. The address decoding at the SDK-80 also assumes only a very small memory configuration and hence two inputs of the 8205 address decoders had to be disconnected from the ground and properly connected to the address bus (also some extra 8205 were added). SDK-80 accepts only one RESTART signal and therefore the interrupt acknowledge signal ($\overline{\text{INTA}}$) of the control bus was disconnected from +12 V (*via* a 1 K resistor) and used as an "enable level read" ($\overline{\text{ELR}}$) signal for the priority interrupt control unit 8214 (PICU) and as the "device select" signal DS_1 for 8212 which transfers the restart commands from PICU to the data bus.

The system soft-ware was written in such a way that the user can select the output device from the keyboard depending whether the hard-copy is needed or the TV-screen output is preferred. The eight level interrupt is allowed which is hard-ware controlled by the 8214 interrupt controller with the directory of the service routines in the random access memory (RAM) which therefore can be readily modified by the software. Otherwise our monitor program included similar functions as those in the Intel software[7] (like the commands for storing and recalling the hexadecimal codes on or form a specified memory address, jump of the program control to a given address, memory move, memory clear, memory list routines, routines for the tape-reader and tape-puncher, TV-screen servicing, the IBM code conversion and others).

The analogue output from the μC was made by means of a 12-bit digital-to-analogue convertor supplied by Datel model DAC-69-12-BI having the offset binary input and current output. The settling time is 200 ns. Its output was amplified to $\pm 2{\cdot}5$ V giving a resolution of 1·2 mV/bit.

The analogue input signal was pre-amplified then fed to the sample/hold circuit (SH) model SHA III (Analogue Devices) and digitized by a 12 bit analogue-to-digital convertor (ADC) model ADC-MA 12 B 2 B (Datel) featuring the 20 µs conversion time. The conversion was started from one of the eight outputs of 8205 address decoder by means of either OUT *n* or IN *n* instruction. This way of clearing and starting ADC is faster then using a single output bit of the PIO port. The sampling signal for the SH was derived from the "conversion time" pulse produced by the ADC during the period of conversion. This pulse starts sufficiently prior to the real start of conversion thus allowing the necessary settling time of the SH circuit. Both converters ADC and DAC were connected to the system data bus *via* two programmable peripheral input/output circuit (PIO) model 8255 (Intel). Since both convertors use 12 bits each of them used 8 bits of the port A and 4 bits of port C of the PIO (for a thorough description of the 8255 operation the reader should refer to the manufacturer's literature[7,8]). The remaining 4 bits of the port C were used to trigger the switching of the drop-time control or the XY recorder pen lowering.

The schematic connection of both analogue signals, the potentiostat (POT) and the electrochemical cell (C) is also shown in Fig. 1. The potentiostat was of conventional type constructed from operational amplifiers Burr–Brown namely model 3342/15 C

(high slew-rate, fast settling, and low output impedance type) as a control amplifier and two amplifiers model 3403A (high slew-rate, fast settling, differential type) as a voltage follower and as a current-to-voltage transducer. The positive feedback loop for additional solution resistance compensation was used as well.

The programmable interval timer and counter contains three independent 16 bit counters and is therefore a very versatile device for desired timing of experiments. It requires to supply the external frequency for each counter which will determine the counting rate. Also the counting can be externally gated. One of the counters we have supplied with 20 KHz square wave derived from the CPU clock generator 8224 through a series of frequency dividers. The output of this counter was used as an interrupt request pulse for PICU 8214. By this arrangement one can easily select the intervals between 50 μs and 3·27 s. The other two counters can be used as required by the user (like the longer timing, event counting, as a one-shot *etc.*).

The SDK-80 microcomputer itself can accomodate only 4 K of EPROM (errasable programmable read only memory) and 2 K of RAM (random acces memory). Hence we designed another memory board enabling to enlarge the memory up to 8 K. of EPROM and 16 K of RAM. This board included also an EPROM programming circuit. The EPROM's were of the type 8708 or 2708 and the static RAM's were 2114 (both manufactured by Intel).

Interfacing to Standard Peripheral Devices

Our μC system uses as a main interfacing device the "Interface box M17" which was introduced in the Czechoslovak Academy of Sciences[9] in early 1970's prior to the IEEE interface standard. The system M17 features first of all useful simplicity for small computer configurations like those involving desk-top calculators and minicomputers and already a great variety of interface cards for all kinds of peripherals (mainly of czechoslovak production) is now available. Within the M17 the signals are transmitted *via* 36 lines containing following signals: 8 data lines (D0 to D7); 8 address lines (A0 to A8); address expander (E) if more than one M17 is used; multiple response enable (MRE) indicating the validity of information on the address lines; input/output signal (I/O) showing the direction of data flow; data valid signal (DAV) indicating the presence of a byte to be transmitted on the data lines; data accepted signal (DAC) informing the computer that the data byte was read and its presence on data lines is no more needed; ready for data signal (RFD); stop signal (STOP) clearing all interface card functions.

The peripheral interface cards are specific for each device regardless of the type of a computer controlling the M17 box. For each type of a computer a specialized computer interface card has to be used. In this section such a computer interface card is described which will enable to use the SDK-80 μC in connection with the M17 Interface box. Part of this interface was mounted directly on the SDK-80

board in order to prevent possible CPU damage in the case of an accidental cable short connection and is shown in Fig. 2. In this and all following descriptions we will be as brief as possible and therefore we will not mention in the text occasional usage of several invertors needed to achieve the proper signal logic level for a given purpose.

The main function of the circuit on Fig. 2 is to derive a signal indicating that the Interface box M17 was selected for data transfer (signal "interface select", ISEL), if so then to pass the proper address bus information to M17 and also to consider the current state of the interface M17 namely whether the M17 is ready to accept a new task (signal "interface ready", IRDY). Also the data flow direction is indicated as a READ signal appearing if either "memory read" or "input read" is issued from the CPU and the system controller. Gates 1 and 2 are NANDing the CPU data bits D4 and D6 respectively with appropriate output of a decoder 16 for M17

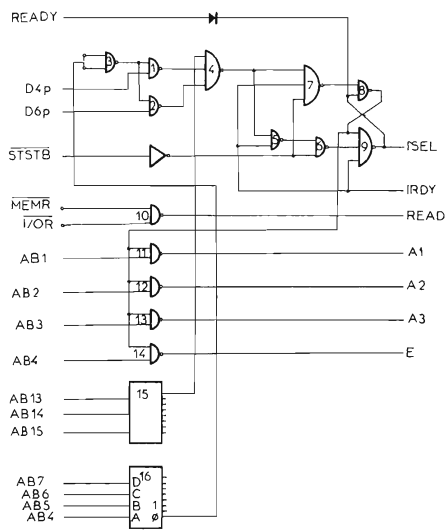

FIG. 2

The circuit for addressing peripherals *via* the Interface Box M—17. The gates and the inverters are of MH 74 family (Tesla) of the TTL logic and the address decoders 15 and 16 are Intel 8205. The diode is GAZ 51. For signal description see the text

addressing. At the beginning of each CPU machine cycle the so-called status word[7,8] appears on the data lines. The status word is a byte which has either bit D4 or bit D6 on logic level 1 provided that the instruction IN or OUT is going to be executed respectively. The status word appearence is indicated by a status strobe pulse ($\overline{\text{STSB}}$) used in our design for gating D4 or D6 through the gates 6 or 7. However, this transfer is blocked on gates 5 and 7 if the Interface box M17 is not yet ready to accept another input/output (I/O) task shown by the state of the "interface ready" signal (IRDY). The peripheral device can be considered also as a memory locatation where the CPU is either writing (for output devices) or reading a byte (for input devices). Such a configuration of peripherals is called the "memory mapped input/output" and in our system is implemented *via* the gate 4 where the signal selecting upper 8K of the memory (from the decoder 15) is allowed also to produce the ISEL signal for M17. This feature enables to use all kinds of memory transfer instructions from the 8080 instruction set and also enhances the number of possible peripheral devices. The complement of ISEL signal (the output of the gate 8) is used to transfer the state of the four low address bits from the address bus through the gates 11 to 14 yielding thus A1, A2, A3 and E signals used further in Fig. 3 for addressing individual interface cards within the M17 box. Therefore the same device number can be used for one input and one output instrument at the same time. The direction of data flow is determined from the state of the gate 10 being active for either INPUT instruction or a memory read type instruction (like LDA addr., MOV r, M, and others[3,7,8]).

The main part of the interface SDK80–M17 shown in Fig. 3 was assembled as a plug-in card to the M17 main-frame. Its principal function is to enable the bi-directional data transfer together with appropriate timing of the control signals (like MRE, DAV, DAC, and I/O). The bi-directional data lines are arranged by means of eight pairs of inverters connected in parallel but in opposite direction (only one pair is shown in Fig. 3). The power supply for these inverters is switched on by corresponding switching transistors Tr1 to Tr4 which are opened depending on the desired direction (determined by the state of READ signal). Because the data transfer takes place during the third machine cycle of either IN or OUT instruction the controll signals are derived by means of three D-type flip-flops 1 to 3 using ISEL as the input and Φ_2 pulses of the CPU clock generator as the clock signal. When an ISEL pulse is stored in the flip-flop 1 the interface is busy which is indicated by setting the flip-flop 4. Busy state is cleared after the data transfer through gates 1 to 3 or when the system RESET (and therefore interface STOP) signal was issued from the CPU. The other control signals, MRE, DAC, and DAV, are derived according to M17 and CPU timing requirements. The detailed description of the circuit function is beyond the scope of this article. The last function of the interface circuit is the proper addressing of the interface cards which is achieved by means of a decoder 5 having as the input signals the address information transmitted by the first part of the interface (Fig. 2) from the address bus.

Peripheral Arithmetic Microprocessor (AμP)

A small μC system as the one described here can be quite economically equipped with the arithmetic capabilities by incorporating a number-oriented processor now available on the market. In this design we used the model MM 57109 (National Semiconductors, USA) (ref.[10]) which is designed in such a way that it can work as a peripheral device of a mini- or a microcomputer (or be just a stand-alone system). Its instruction set features all kinds of functions (trigonometric, logarithmic, exponen-

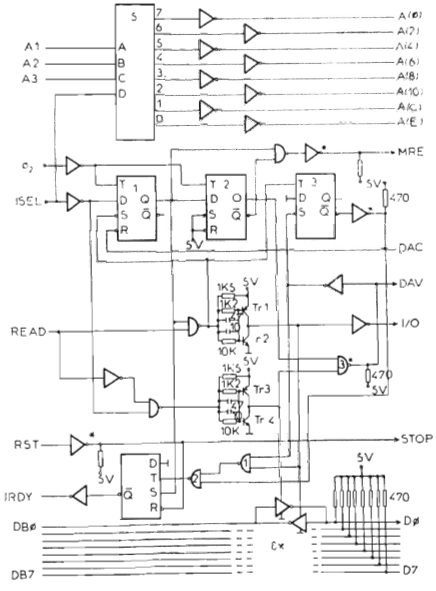

FIG. 3

The interface circuit for a bi-directional data transfer between the CPU and Interface Box M—17. The gates and the inverters are MH 74 family of TTL logic, the D-type bistable circuits are MH 7475, the decoder 5 is MH 7475, the transistors Tr_1 and Tr_3 are KSY 81, Tr_2 and Tr_3 are KSY 71. The values of capacitors are in pF and resistors in ohms

tial act.) usually found in pocket-type calculators. This chip accepts 6 bits as an instruction code (I1 to I6 inputs) and by means of a "ready pulse" (RDY) signalises the end of the instruction execution. The result is outputed as a stream of up to 8 BCD digits (binary coded decimal) and each digit output is indicated by means of the "write" signal ($R/\overline{W}$). Furthermore under certain circumstances the error bit and the braching condition flag is issued by this microprocessor. The communication between AμP and the 8080 CPU is schematically shown in Fig. 4. The program instructions for AμP are resident in the μC memory and are fetched to AμP by means of a software *via* the data latch 8212 (the circuit 2 in Fig. 4) and then the interrupt system is activated. The interrupt request signals from the AμP are derived from the RDY and $R/\overline{W}$ signals, the last one being of a higher priority. The communication program checks each new instruction for the operation code of the "result output" instruction and activates the interrupts accordingly. When data are being waited for in a loop the CPU is reading the AμP status by means of gates 4 to 6 and masking first the RDY and then the $R/\overline{W}$ bits. When the $R/\overline{W}$ bit indicates the digit output the digit is read

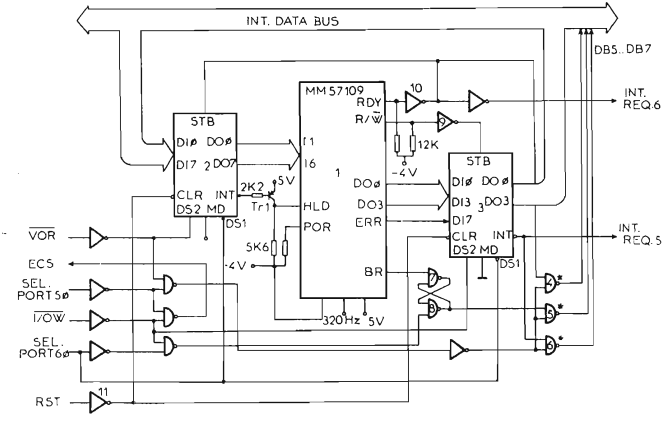

FIG. 4

The peripheral arithmetic microprocessor interface to the data bus. The LSI circuits are: 1 the arithmetic processor MM 57109 (Natl. Semiconductors), 2 and 3 are 8212 (Intel) interrupting I/O ports, the gates 7 to 10 are the low-power MH 74LS00 and the others are standard TTL logic. The transistor Tr_1 is TR 15

via a data latch 8212 (the circuit 3) and stored in the RAM memory and the waiting loop for R/$\overline{W}$ is reentered again until all digits have been obtained. In Fig. 4 the port No 60 is assigned to the instruction output and result input whereas the port No. 50 is for the AμP status input. In the case of a conditional branching instructions the branch flag is read by a similar means through a bistable circuit (the gates 7 and 8) and the gate 5. The inverters 9 and 10 and the gate 7 are of a "low power TTL logic" type because of the AμP fan-our requirements for these signals. The transistor T_1 is used in order to have TTL voltage level compatibility. When the whole μC is reset the latches 2 and 3 in Fig. 4 are cleared. The AμP is not restarted *via* its POR (power--on or reset) signal but is always initialized by sending the instruction "master clear" as the first one.

The instruction set of the AμP includes 64 operation codes (opcodes) from 00_H to $3F_H$ some od them being a 2-word instructions. It is desirable to keep records about the current state of the AμP, *e.g.* as far as the fixed/floating point or mantisa digit count selection is concerned. Also the four register stack and one memory register of the AμP is not sufficient for most of the intended calculations and hence a provision was made within the communication program to treat certain instructions by suitable subroutines prior to sending them as an operation code to the AμP. From the value of the highest opcode ($3F_H$) one can see that all ASCII codes of upper- and lower-case letters and the brackets { } will not be misinterpreted as a valid instruction for the AμP. Selected ASCII codes of letters listed in the Table I are used to represent various subprogrammes for sevicing the AμP. Within the series of opcode of an arithmetic program may be included these selected ASCII codes and the whole arithmetic program is exited upon recognition of the RUB OUT ($7F_H$) code. By means of these pseudoinstructions one can write subroutines within the arithmetic program and access them through C (call) and R (return) commands. Data storage in the RAM memory is possible by S (store) and G (get). The main features of the pseudocommands are summarized in the Table I. The whole communication program has the size of about 1 1/4 Kbytes of memory and is EPROM resident utilizing RAM only for the stack of return addresses, data register storage, the arithmetic program counter act. Fairly large computing capabilities requiring only relatively small memory occupation, as mentioned above, clearly demonstrate that the application of a peripheral arithmetic μP is an economical solution of this measurement-oriented system.

Applications in Electrochemistry

Staircase Voltammetry (SCV)

The most straight-forward application is to use the μC as a source of a linearly varying polarization voltage with time for the working electrode and to record

TABLE I

Commands available for the arithmetic microprocessor programming within the communication programe

Command syntax	Name	ASCII code	Description	No of bytes
J address	Jump	4A	Jump to a given location within the arithmetic program	3
C address	Call	43	Call a subroutine within the arithmetic program	3
R	Return	52	Subroutine termination with return	1
S n	Store	53	Store the last result in a register n_H	2
G n	Get	47	Get the register n_H content to X register	2
I address	Increment	49	Increment by 1 a specified memory location	3
D address	Decrement	44	Decrement by 1 a specified memory location	3
K	Keyboard	4B	Keyboard data entry in FXD or FLT format	1
W	Write	57	Write the last outputed result	1
{text}	Message	7D, 7B	Codes within the brackets are displayed or printed but not sent to the AμP	
	RUB OUT	7F	Arithmetic program termination and exit from the communication program	1

and store the resulting current response of the electrochemical cell. The μC can rather simply generate a staircase shaped voltage function instead of the linear voltage sweep and it was shown that both cases are analogous with a sufficiently small step size. A detailed theoretical treatment of the staircase method was published[11-15] and it was found that the parameters of a voltammetric curves are dependent on the voltage step size and the current read-out timing relative to the step edge. The main advantage of a computerized cyclic voltammetric experiment is the digital data record which can be directly used for further analysis of resulting curves instead of using the storage oscilloscope screen and photographic procedure prior to the evaluation. Data punched on a paper tape were numericaly analyzed on the Hewlett–Packard desk-top computers model 9820A or 9845S.

Our program for the staircase voltammetry allows an operator to input *via* the keyboard: *i*) polarization voltage limits; *ii*) voltage step size; *iii*) time between the steps; *iv*) scan direction; *v*) number of cycles to be repeated.

For operator information the scan rate is calculated and printed and after completing the task the μC offers to select: *i*) experiment repetition; *ii*) entry of a new set of input parameters; *iii*) permanent data storage; *iv*) program exit.

An example of a computer dialogue for the cyclic staircase experiment. In Fig. 5: POSIT. POTENTIAL (MV) = 200; NEGAT. POTENTIAL (MV) = −150; POTL. INCREMENT (NO. BITS) = 2; TIME DEALY (MS) = 20; SCAN RATE (V/S) = = 0·118; SCAN DIRECTION (P/N) : P; NO OF CYCLES = 1; CELL READY? PRESS CONT.!; SELECT: REPEATE – PUNCH – NEW – QUITT? P.

This version of the SCV programme uses the 12-bit precision both for D/A and for A/D conversions and hence all data input/output on the 8-bit μP has to be done in two steps. In this case the highest scan rate achieved is 36 V/s if one allows 20mV steps as a maximum. By sacrifying the precision an increase of this scan rate limit is possible but for majority of investigations is not needed.

An experimental result of the SCV is shown in Fig. 5*a* and was chosen with respect to demonstrate below the advantage of digitized data treatment especially when the noise level substantilly deteriorates the data quality for a precise analysis. Fig. 5*a* shows the reversible oxidation of bis-(maleonitriledithiolate)$Ni(II)^{2-}$ in 0·1M-tetra-butylammonium hexafluorophosphate in dichloromethane as a solvent. The positive feedback *iR* drop compensation was used and due to high solution resistance and a low scan rate the signal-to-noise (S/N) ratio was rather poor. Three methods of S/N ratio improvement were tried in order to compare them for routine work.

i) Data smoothing by means of the least square approximation is shown in Fig. 6*a*. We have used the centred step-wise least square fit which is based on calculation of a new-value in the center of a five points data segment according to a formula[2]

$$I_i' = 0{\cdot}0286(-3I_{i-2} + 12I_{i-1} + 17I_i + 12I_{i+1} - 3I_{i+2}) .$$

where I_i is the i-th current point and I'_i is the i-th calculated point of a smoothed curve. Then the segment is moved one data point further until all data (except the two first and the two last points) are smoothed. As can be seen already qualitatively the result of this operation is not very good at least for this S/N level.

ii) The next method used here was a simple ensemble averaging of several data records under the assumption of time invariant system. This procedure gives better results already for four ensembles (Fig. 6*b*) but often cannot be used because only one set of data is measured.

iii) The most powerful even more computational demanding method is the digital filtering[16] which employs the Fast Fourier Transform (FFT) (ref.[17]). In an example in Fig. 5 (the same data array as above) the data set to be filtered is first shifted vertically in order to achieve the first data point to coincide with zero, then a rotation around the first point is performed which results in a new data array starting from and merging to the zero line. This new array is transformed by FFT yielding a frequency spectrum (Fig. 5*b*) which contains the information about the SCV

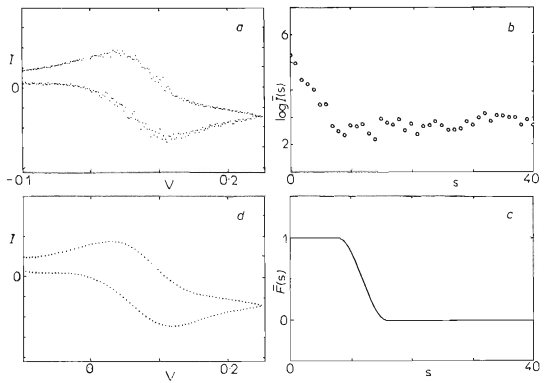

FIG. 5

The cyclic staircase voltammetry of 0·5 mM bis-(maleonitriledithiolate) nickel(II)$^{2-}$ in 0·1M tetrabutylammonium hexafluorophosphate in dichloromethane on the Pt-electrode (*a*). The voltage-step size 2·4 mV, the step time interval 20 ms, the current reading at 19·9 ms. (*b*) Part of the Fourier-transformed current spectrum. (*c*) The filtration function. (*d*) Smooth data after inverse FFT. The potential scale is *vs* Ag/AgCl/1M-LiCl electrode

curve at low frequencies whereas the high frequency part of the spectrum is mainly due to the noise contribution. The noise filtration is achieved by multiplication of the spectrum by a filter function F(s) having the value 1 up to a suitably chosen cut-off frequency Sc and the value 0 for all higher frequencies. Here we have employed slightly modified filter function[18,19] in order to get a less abrupt shape around the Sc, namely,

$$F(s) = 1\,, \quad S \leqq Sc$$

$$F(s) = 0{\cdot}5 + 0{\cdot}5 \cos\left(\pi(S - Sc)/2\,c\right), \quad Sc < S < 2\,Sc$$

$$F(s) = 0\,, \quad S > 2\,Sc\,.$$

After the inverse FFT, the back rotation and shifting the resulting SCV curve (Fig. 5*d*) has a considerably better S/N ratio. When multiplication of two spectra is done then the inverse transform corresponds to the convolution of respective transform pairs in the time domain[17]. Since the step-shaped F(s) function coorresponds in the time domain to the sin t/t function certain ringing can occur. This ringing is minimized here by a less steep F(s) shape.

Fourier Transform Electrode Admittance Measurements

Most of the experimental techniques used for the faradaic impedance $Z(\omega)$ or admittance $Y(\omega)$ measurements are based on a single frequency perturbation[20,21]. A more sophisticated approach taking advantage of some kind of a broad-band perturbating small-amplitude signal are always used in connection with a minicomputer[19,22]. In this section we can briefly demonstrate that even a much smaller

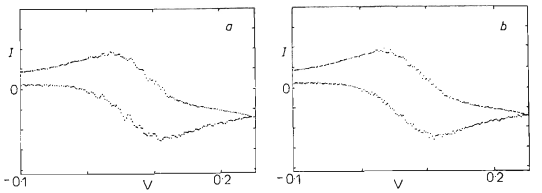

FIG. 6

The staircase voltammetry data smoothing: (*a*) the centred step-wise least-square fit, (*b*) the ensemble averaging of four data records. The same data as in Fig. 5*a*

computing system such as our μC is capable of such an application. The FFT admittance measurements are especially useful in investigation of redox system of limited stability like coordination compounds sensitive to oxygen or moisture *etc.* The evaluation of parameters characteristic for a given redox system (the charge-transfer rate or adsorption parameters) is usually done by an analysis of the frequency domain profile of $Y(\omega)$ which may turn out to be impossible if the data collection from point-by-point assesment is attempted for above-mentioned systems.

The definition of the cell admittance in the frequency domain

$$Y(\omega) = I(\omega)/E(\omega),$$

where $E(\omega)$ is the ac component of the perturbation voltage across the electrical double-layer and $I(\omega)$ is the resulting cell ac response current, is not restricted to a single-frequency $E(\omega)$. In our arrangement the μC acts as a multiple-frequency signal generator periodicaly scaning through a 1024-data array in the memory which represents 1024 discrete samples of a perturbation function

$$E(t) = \sum_{i=1}^{18} \sin(n_i \omega t + \varphi_i).$$

Eighteen values of frequencies $n_i\omega$ and phase angles φ_i in $E(t)$ are selected according to[22] in such a way that n_i are the prime numbers and φ_i are the random numbers between 0 and $\pi/2$. Values of $E(t)$ are scaled in order to cover fully the D/A converter range and are stored in the binary form. Output of every $E(t)$ sample is followed by reading and storage a $I(t)$ sample until all 1 024 points were obtained and then the whole cycle is repeated for a selected number of cycles. For a better S/N ratio the ensemble averaging is applied using the tripleprecision addition subroutine. In order to evaluate $Y(\omega)$ one experiment is performed with a dummy cell of precisely known impedance and its response is used for $E(t)$ calibration. The calculation of $E(\omega)$ and $I(\omega)$ from respective $E(t)$ and $I(t)$ records is done by means of the FFT algorithm. Similarly as in the case of SCV we are using 12-bit data format which allows one cycle of 1 024 points to be completed in 70 ms. The highest frequency component is for $n_{18} = 249$ which gives the frequency interval of our measurements from 14 Hz to 3·5 KHz. Also here the upper limit can be increased on expense of accuracy or the ensemble averaging. In Fig. 7 are shown the results obtained for the first reduction step at $-0{\cdot}36$ V (*vs* S.C.E.) of tris(5-nitrophenanthroline) iron(III) complex in acetonitrile (0·1M-tetrabutylammonium hexafluorophosphate as an indifferent electrolyte). The first row (Figs 7*a* and *b*) is the time- and frequency domain response of the 10 kΩ dummy cell, the second row (Figs 7*c* and *d*) is the response of the investigated system, Fig. 7*e* is the complex impedance plot and Fig. 7*f* is finally the faradaic phase angle *vs* frequency plot yielding the value of the hetero-

geneous charge-transfer rate constant. A more detailed description of the kinetics of this complex is in preparation. Prior to the calculation of $Y(\omega)$ both $E(\omega)$ and $I(\omega)$ spectra were identically treated by the application of twice the Hanning function[18] which in the frequency domain is simple implemented as

$$H(\omega) = \tfrac{1}{2}\delta(\omega) - \tfrac{1}{4}[\delta(\omega - 1) + \delta(\omega + 1)] ,$$

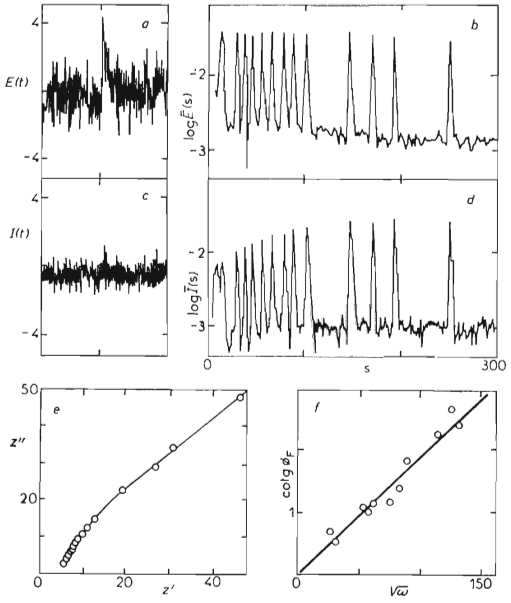

FIG. 7

The FFT impedance measurements of 0·43 mM tris-(5-nitro-1,10-phenanthroline) iron(III) complex in 0·1M tetrabutylammonium hexafluorophosphate in acetonitrile at the D.M.E. (*a*) and (*b*) is the time and frequency domain display of the perturbation signal respectively. (*c*) and (*d*) are the resulting time and frequency domain responses. (*e*) is the complex cell impedance plot and (*f*) the faradaic phase-angle *vs* the square-root of the angular frequency plot after substraction of the solution resistance and the double-layer capacity

where $\delta(\omega)$ is the unity height delta function. The application of $H(\omega)$ on the data array has the effect of minimization of errors inherent to the digital transform techniques when an infinite smooth function is described by a discrete data array of finite length.

Two applications of the μC described here were selected in order to demonstrate the capability even much smaller computing facility than usually was used for such a purpose in the past.

REFERENCES

1. Perone S. P., Jones D. P.: *Digital Computers in Scientific Instrumentation.* McGraw-Hill, New York 1973.
2. Matson J. S., Mark H. B., McDonald H. C.: *Computers in Chemistry and Instrumentation*, Vol. 1—3. M. Dekker, New York 1972.
3. Souček B.: *Microprocessors and Microcomputers.* Wiley, New York 1976.
4. Dessy R. E., Van Vuuren P. J., Titus J. A.: Anal. Chem. *46*, 917 A (1974).
5. Dessy R. E., Titus J. A., Van Vuuren P. J.: Anal. Chem. *46*, 1055 A (1974).
6. Haarmann T., Koser H. J. K.: Fresenius' Z. Anal. Chem. *296*, 18—105 (1979).
7. Intel: MCS-80™ User's Manual, 1977.
8. Mikroprozessor Bausteine, System SAB 8080, Siemens 1977.
9. Interface Box M 17, Technical report, VD Czechoslovak Academy of Sciences, Prague 1974.
10. MM 57109 MOS/LSI Number-Oriented Microprocessor, National Semiconductor Corp., Santa Clara, USA, 1977.
11. Ferrier D. R., Schroeder R. R.: J. Electroanal. Chem. Interfacial Electrochem. *45*, 343 (1973).
12. Ferrier D. R., Chidester D. H., Schroeder R. R.: J. Electroanal. Chem.Interfacial Electrochem. *45*, 361 (1973).
13. Zipper J. J., Perone S. P.: Anal. Chem. *45*, 452 (1973).
14. Ryan M. H.: J. Electroanal. Chem. Interfacial Electrochem. *79*, 105 (1977).
15. Miaw L.-H. L., Bourdeau P. A., Pichler M. A., Perone S. P.: Anal. Chem. *50*, 1988 (1981).
16. Smith D. E.: Anal. Chem. *48*, 517 A (1976).
17. Brigham E. O.: *The Fast Fourier Transform.* Prentice-Hall, Engelwood Cliffs, N. J. 1974.
18. Olivier B. M., Cage J. M.: *Electronic Measurements and Instrumentation*, McGraw-Hill, New York 1971.
19. Creason S. C., Smith D. E.: J. Electroanal. Chem. Interfacial Electrochem. *36*, App. 1 (1972).
20. Sluyters-Rehbach G., Sluyters J. H.: *Electroanalytical Chemistry* (A. J. Bard, Ed.), Vol. 4, p. 1. M. Dekker, New York 1970.
21. Pospíšil L.: Chem. Listy *74*, 694 (1980).
22. De Levie R., Thomas J. W., Abbey K. M.: J. Electroanal. Chem. Interfacial Electrochem. *62* 111 (1975).