12-bit digital audio time delay using the 6809

Commercial time delay units are not usually flexible enough to suit computer control, as may be required in laboratory situations. So **B T G Tan and L P Tay** have designed a programmable audio delay system based on the Apple II with a 6809 coprocessor

The paper describes a programmable digital audio time delay system which uses the 6809 microprocessor to store and retrieve 12-bit samples in RAM. A/D and D/A conversions were accomplished by a 12-bit analogue I/O board for the Apple II microcomputer. A 6809 machine code time delay program, which was run on a 6809 coprocessor board for the Apple II, was able to attain an audio frequency response of over 14 kHz.

microsystems audio time delay Apple II 6809

The digital audio time delay generator is a primary tool in the design of present-day sound reinforcement systems. Most professional time delay units are dedicated units using hard-wired logic circuitry; these units are usually designed for use in sound reinforcement system installations in auditoria and concert halls, and their time delay values are not designed for rapid alteration under computer control. The present project arose out of a need for a high-quality digital time delay unit which would be easily programmable, for use in research on the psychoacoustic effects of time delays on the perception of sound sources.

The 8-bit microprocessor is a suitably flexible and inexpensive controller for a programmable time delay unit[1]. However, the signal-to-noise ratio requirements in most applications for digital time delay units demand a data word length longer than 8 bit; 8 bit would give a signal-to-noise ratio of only 48 dB or so, which is inadequate for high-quality sound reinforcement. Thus a 12-bit or 16-bit data word length is required, implying that a 16-bit microprocessor system might be necessary to perform the delay processing with sufficient speed.

However, a system based on an 8-bit data bus was judged to be preferable from the point of view of system cost and complexity. An 8-bit microprocessor would have to perform a pair of 8-bit write operations to store one 12-bit or 16-bit sample in memory and it would take a pair of 8-bit read operations to retrieve the same sample. Most current 8-bit microprocessors would not be able to cope with 16-bit data transfers at an acceptable rate for high-quality audio sampling; the solution adopted was to use a hybrid 8/16-bit microprocessor, the Motorola 6809. There are more powerful processors available which have 8-bit data buses, such as the 8088 and the 68008, but the 6809 proved to be adequate for the present purpose.

Department of Physics, National University of Singapore, Singapore 0511

Rather than building a selfcontained standalone digital delay system, it was decided to use a personal computer with an open bus architecture as the basis of the system. The Apple II was chosen, though the IBM PC could also have been used, as both computers have open buses which can be easily interfaced through their slots. The Apple II is based on the 8-bit 6502 microprocessor and has an I/O bus with eight slots[2]. In the Apple II's 64k memory map, 2k (from $C000 to $C7FF) is reserved for I/O through the eight slots.

The digital delay system included two boards to plug into the I/O slots; one board was a specially designed and constructed 12-bit analogue I/O board and the other was a commercially available 6809-based coprocessor board.

ANALOGUE I/O BOARD

The analogue I/O board was designed to perform two related functions:

- to receive an audio analogue input, convert it to a 12-bit digital sample and store it in the Apple II's memory
- to retrieve a 12-bit digital sample from the Apple II's memory and convert it to an analogue audio signal

The circuit diagram of the Analog I/O board is shown in Figure 1. The 12-bit A/D and D/A conversions were performed by an Analog Devices AD574 A/D converter and an AD565 D/A converter respectively.

There are two methods by which the AD574 can be interfaced to an 8-bit bus[3]. One method is to connect it directly to the Apple II bus with the eight least significant bits and the four remaining bits both connected to the 8-bit data bus. The AD574 is then operated in its 8-bit mode, with 12-bit data being output in two stages. A/D conversion is initiated by executing a write instruction to the AD574 address.

The other method, which was the one actually adopted, is to use the AD574 in standalone mode and direct its 12-bit output into a 6821 peripheral interface adaptor (PIA) with two 8-bit ports. The AD574 can then be operated in 12-bit mode; AD574 interfacing and programming is also simplified. One advantage of using a PIA is that the CA2 line of port A can be configured as an output for handshaking and used to initiate A/D conversion via the AD574's read/convert (R/C) pin. This eliminates one instruction in the time delay routine and increases the data sampling rate. A second PIA was used to direct the outgoing 12-bit samples from the Apple II memory to the AD565 D/A converter.

0141-9331/86/09500-06 $03.00

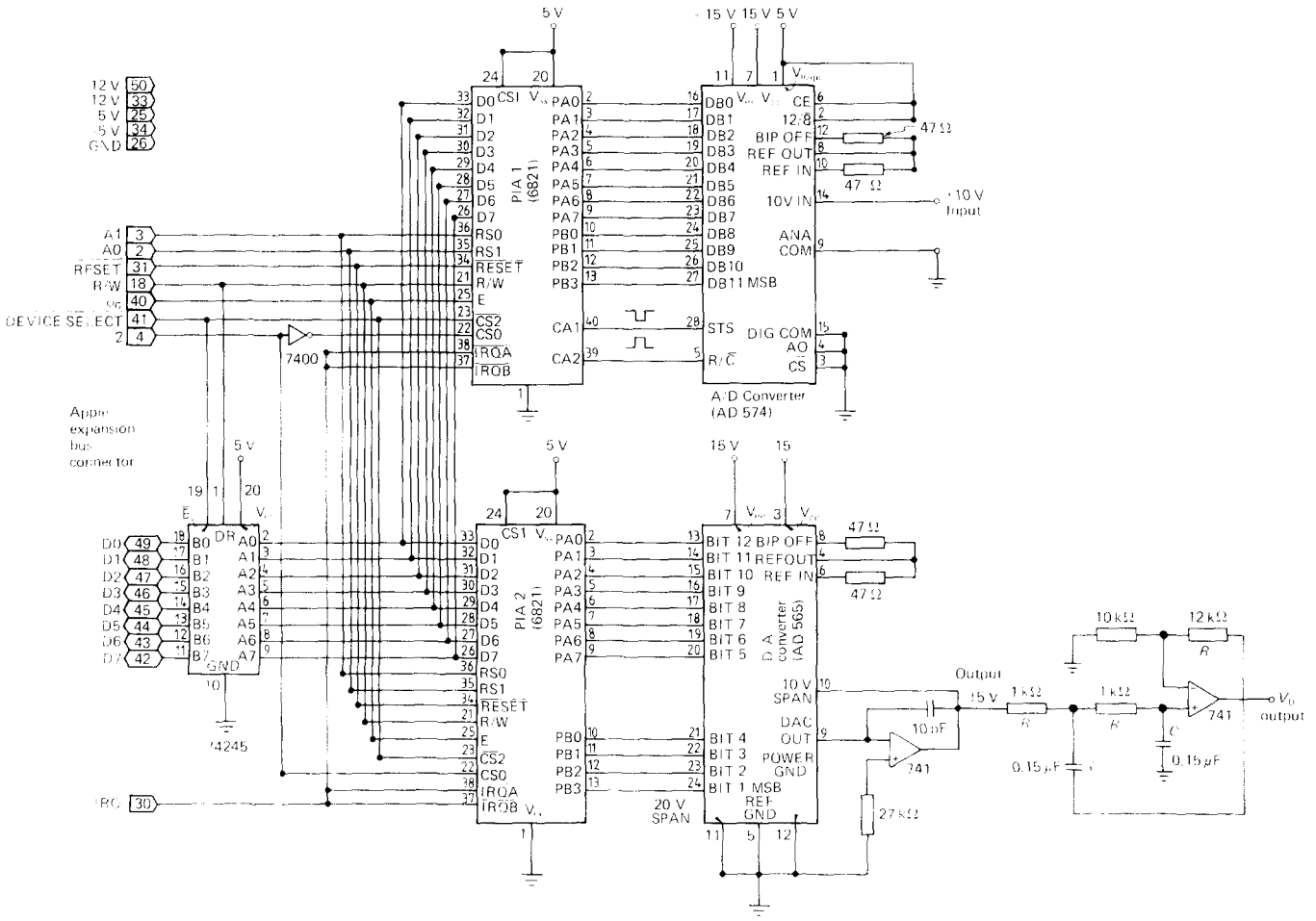


Figure 1. Circuit diagram of analogue I/O board

The addresses occupied by the two PIAs in the Apple II's memory map are $C0x0-$C0x3 or $C0x8-$C0xB (PIA 1) and $C0x4-$C0x7 or $C0xC-$C0xF (PIA 2), where the hex digit x is $n + 8$, n being the slot into which the analogue board is inserted. Normally, the four consecutive addresses occupied by a 6821 PIA are assigned to the PIA registers in the following order[4]

PDRA/DDRA
CRA
PDRB/DDRB
CRB

The register select lines of the two PIAs, RS0 and RS1, were connected to the address bus lines A1 and A0 respectively; this is the reverse of the normal practice, and was done so that the four PIA addresses would be in the order

PDRA/DDRA
PDRB/DDRB
CRA
CRB

This places the addresses for the two peripheral data registers PDRA and PDRB directly adjacent to each other and so enables them to be addressed simultaneously by the 16-bit load and store instructions of the 6809.

The AD574 A/D converter is interfaced to PIA 1. The eight least significant bits of the AD574 output are connected to port A and the remaining four bits to the four lower significant bit lines of port B of PIA 1. The AD574 A/D conversion process is initiated by its R/$\overline{C}$ pin. This pin is connected to the CA2 input of port A of PIA 1. The CA2 pin is configured as an output so that it can initiate an A/D conversion at the AD574 when a high-to-low transition occurs at CA2. The PIA port is configured for handshaking, so that a high-to-low transition occurs every time a data read operation is performed there. The AD574 status pin (STS) indicates that the conversion is complete by going from high to low[3]. This pin is connected to CA1 so that PIA 1 can initiate an interrupt request for the 6809 to read the 12-bit data. The read operation then initiates another conversion through the CA2 pin. The AD574 is configured to accept a bipolar analogue input with a maximum amplitude of 10 V.

The AD565 is also configured to give a 10 V maximum output during bipolar operation[3]. Its eight least significant digital input lines are connected to port A, and the remaining four lines are connected to the four lower significant bit lines of PIA 2 port B. The output is passed through a 741 operational amplifier acting as a current-to-voltage converter, and then through another 741 acting as

a second-order Butterworth low-pass filter to smooth the analogue output signal.

6809 COPROCESSOR BOARD

The 6809 coprocessor board used is manufactured by Stellation Two and is known as 'the Mill'. It uses the 6809E microprocessor, which runs off the 1.023 MHz clock of the Apple II. The 6809E runs concurrently with the Apple II's 6502, the 6502 having access to the bus during the 6809's dead cycles[5] (i.e. whenever the 6809 valid memory address is not valid). The operation of the 6809 is controlled by an onboard 8-bit control register. Each bit of the control register controls a particular board function. Each of these bits is turned off or on by writing $80 or $00 to an address from $C080 + *n*0 to $C087 + *n*0, where *n* is the number of the slot into which the 6809 board is inserted. By writing to the appropriate address, the 6502 or 6809 can be enabled or disabled.

When the Apple II is switched on, the 6502 is enabled and the 6809 disabled. A short program was written in BASIC to set up the starting address of the 6809 program in the Apple II memory and to activate the 6809. When the BASIC program is run, it executes the 6809 program which runs concurrently with the 6502.

DIGITAL DELAY SOFTWARE

The basic principle of the machine code program used to accomplish digital delay using the analogue I/O board is very simple. All that is needed is to load the digital samples into a contiguous block of memory. For example, the address range from $6000 to $61FF will give 512 contiguous addresses. The samples from the AD574 are stored first in $6000 and $6001 (with 2 byte for each sample) then in $6002 and $6003, and so on until $61FE and $61FF are reached. The next sample goes back to the beginning of the block at $6000 and $6001 and the whole cycle is repeated. The outgoing digital samples are retrieved from the memory block and fed to the AD565 in a similar manner.

The length of the time delay depends on two factors: the length of the memory block and the point relative to the incoming samples at which the outgoing samples are retrieved. For a given block length, the longest possible delay is obtained when the outgoing samples are retrieved from addresses immediately ahead of the incoming samples. This is accomplished in the delay programs by initially setting the pointer for the incoming samples at $6000 and that for the outgoing samples at $6002.

It is of course possible to accomplish the time delay with an 8-bit microprocessor machine code program. Figure 2 shows a program which was written for the Apple II's own 6502. The program assumes that the analogue I/O board is in slot 3. PIA 1 occupies the addresses $C0B0–$C0B3 and PIA 2 occupies $C0B4–$C0B7. The PIA initialization routine sets CA2 of PIA 1 as an output so that it can initiate the A/D conversion through the R/$\overline{C}$ input of the AD574. Port A of PIA 1 is configured for input handshaking. The AD574 requires a maximum of 35 µs for the

```
SOURCE FILE: DELAY05
----- NEXT OBJECT FILE NAME IS DELAY05.OBJ0
4000:              1          ORG  $4000      6502 TIME DELAY PROGRAM
4000:A9 00         2          LDA  #$0C       PIA INITIALIZATION
4002:8D B2 C0      3          STA  $C0B2
4005:8D B3 C0      4          STA  $C0B3
4008:8D B0 C0      5          STA  $C0B0      DEFINE PIA INPUT PORTS
400B:8D B1 C0      6          STA  $C0B1
400E:8D B6 C0      7          STA  $C0B6
4011:8D B7 C0      8          STA  $C0B7
4014:A9 FF         9          LDA  #$FF
4016:8D B4 C0     10          STA  $C0B4      DEFINE PIA OUTPUT PORTS
4019:8D B5 C0     11          STA  $C0B5
401C:A9 24        12          LDA  #$24       MASK PIA INTERRUPT
401E:8D B2 C0     13          STA  $C0B2
4021:A9 04        14          LDA  #$04
4023:8D B3 C0     15          STA  $C0B3
4026:8D B6 C0     16          STA  $C0B6
4029:8D B7 C0     17          STA  $C0B7
402C:A2 00        18          LDX  #$00       X REG POINTS TO INPUT ADDRESS
402E:A0 02        19          LDY  #$02       Y REG POINTS TO OUTPUT ADDRESS
4030:AD B0 C0     20 LOOP     LDA  $C0B0      START OF DELAY LOOP: INPUT LOW BYTE
4033:9D 00 60     21          STA  $6000,X
4036:AD B1 C0     22          LDA  $C0B1      INPUT HIGH BYTE
4039:9D 01 60     23          STA  $6001,X
403C:B9 00 60     24          LDA  $6000,Y
403F:8D B4 C0     25          STA  $C0B4      OUTPUT LOW BYTE
4042:B9 01 60     26          LDA  $6001,Y
4045:8D B5 C0     27          STA  $C0B5      OUTPUT HIGH BYTE
4048:E8           28          INX
4049:E8           29          INX
404A:C8           30          INY
404B:C8           31          INY
404C:4C 30 40     32          JMP  LOOP       REPEAT LOOP

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

Figure 2. Time delay program for 6502

conversion. The delay loop takes 45 clock cycles — 43.99 µs at a clock rate of 1.023 MHz — so there is no necessity to pad the loops with extra cycles or to interrupt the 6502 to initiate the input of a new sample from the AD574. Hence the PIA interrupt output to the 6502 is masked.

With a loop delay of 43.99 µs the sampling rate is 22.73 kbaud, and hence the highest audio frequency which the time delay can handle is 11.37 kHz. The time delay depends on the length of the memory block. In the present system the block length is maximized using the 8-bit index registers (i.e. 256 byte or 128 12-bit samples), and hence the maximum time delay is 5.63 ms.

The lack of 16-bit index registers makes programs for longer time delays very cumbersome. The indirect addressing modes of the 6502 have to be used, leading to rather long sampling routines and hence long loop times, giving slower sampling rates. The 6800 microprocessor would have provided a 16-bit index register, but two index registers are needed for the indexing of the input and output addresses in the memory block.

Using the 6809, which is capable of 16-bit operations but has an 8-bit data bus, instead of the 6502 resulted in a significant improvement to the time delay program. The 6809 has two 8-bit accumulators which can be combined as one 16-bit accumulator, known as the D accumulator. Thus several 16-bit operations, including 16-bit load and store operations, can be performed. With an 8-bit data path a 16-bit load or store operation needs to transfer the 16-bit data in two 8-bit steps, but this is transparent to the user. The 6809 also has two 16-bit index registers (the X and Y registers), which considerably simplify the indexing of the input and output memory addresses for the delay samples.

Figure 3 shows a time delay program written in 6809 machine code. PIA initialization is similar to that in the 6502 program. The X and Y index registers are initially loaded with $6000 and $6002 respectively. The delay loop routine is now shorter and occupies only 35 clock cycles. When the end of the block is reached, a special end-of-block routine performs the sample processing and returns the index register addresses to the beginning of the block. This end-of-block housekeeping routine also takes 35 clock cycles, so that the transition from the end of the block to the beginning is smooth and does not result in a sampling 'hiccup'. The loop delay is 34.21 µs, giving a sampling rate of 29.23 kbaud and a maximum audio frequency of 14.61 kHz. As the nominal conversion time for the AD574 is 25 µs, this means that it is working quite close to its specified maximum performance.

Unlike the 6502 program, the memory block can be longer than 256 byte, giving longer delay times. In the 6809 program given in Figure 3, the block length is 512 byte or 256 12-bit samples, giving a time delay of 8.76 ms. Figure 4 shows a pulse modulated 4 kHz sine wave and its delayed output. The measured delay is about 8.7 ms. The time delay can of course be extended to much longer lengths, being limited only by the amount

```
2000                  ORG    $2000    6809 TIME DELAY PROGRAM
2000 4F         START CLRA            PIA INITIALIZATION
2001 5F               CLRB
2002 FDC0B2           STD    $C0B2
2005 FDC0B0           STD    $C0B0    DEFINE PIA INPUT PORTS
2008 FDC0B6           STD    $C0B6
200B 86FF             LDA    #$FF
200D C6FF             LDB    #$FF
200F FDC0B4           STD    $C0B4    DEFINE PIA OUTPUT PORTS
2012 8624             LDA    #$24     MASK PIA INTERRUPT
2014 C604             LDB    #$04
2016 FDC0B2           STD    $C0B2
2019 8604             LDA    #$04
201B C604             LDB    #$04
201D FDC0B6           STD    $C0B6
2020 8E6000           LDX    #$6000   STARTING ADDRESS OF MEMORY BLOCK
2023 108E6002         LDY    #$6002   Y REG POINTS TO OUTPUT ADDRESS
2027 FCC0B0     LOOP  LDD    $C0B0    START OF DELAY LOOP; INPUT TWO BYTES
202A ED81             STD,X++
202C ECA1             LDD,Y++
202E FDC0B4           STD    $C0B4    OUTPUT TWO BYTES
2031 8C61FF           CMPX   #$61FF   END OF MEMORY BLOCK?
2034 25F1             BLO    LOOP     NO, RETURN TO LOOP
2036 FCC0B0           LDD    $C0B0    YES. DO END OF BLOCK HOUSEKEEPING
2039 ED84             STD,X
203B 8E6000           LDX    #$6000
203E 108E6000         LDY    #$6000
2042 ECA1             LDD,Y++
2044 FDC0B4           STD    $C0B4
2047 20DE             BRA    LOOP     RETURN TO LOOP
         2000         END    START
       SYMBOLS IN TABLE:

LOOP   $2027    START $2000

SYMBOL TABLE END: 40F9

      STATEMENT ERROR(S),
```

Figure 3. Time delay program for 6809

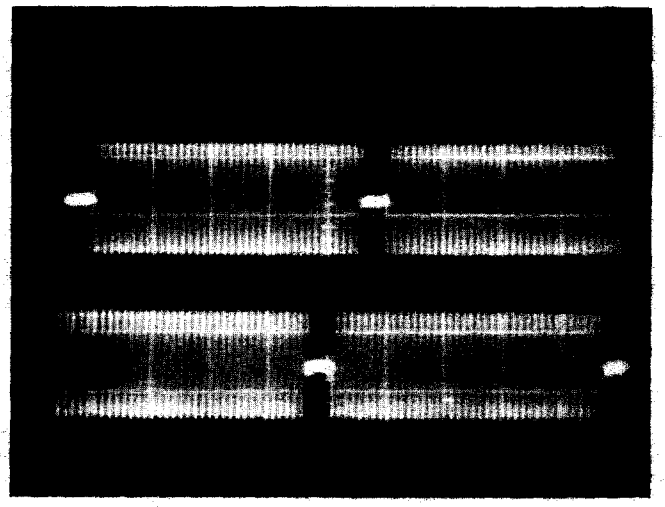


Figure 4. Time delayed pulse modulated 4 kHz sine wave: upper band is original wave (2 V cm^{-1}); lower band is delayed wave (5 V cm^{-1}); time base is 2.08 ms cm^{-1}

of free memory in the Apple II. For example, a 12 kbyte memory block would give a time delay of very nearly 420 ms which, together with the audio frequency range of over 14 kHz, makes the system perfectly adequate for a wide range of audio processing applications.

The delay program could be slightly modified so that the CMPX register instruction which tests for the end of the block could be changed from the immediate addressing mode to refer to a location which contains the block size. In this way, the block size and hence the delay time could be altered under program control, possibly by the 6502, which has concurrent access to the Apple II memory. (This could be extremely useful in situations where the delay time has to change rapidly in accordance with experimental conditions.)

In most cases 12-bit audio samples will give sufficiently good resolution. The signal-to-noise ratio from 12-bit digitization is about 72 dB, which is perfectly adequate for most applications. If 16-bit resolution is required, the same circuit design can be used with only a change in the A/D and D/A converters. The sampling rate would remain the same as for the 12-bit case, since the 6809 is already in effect taking 16-bit samples.

In its standalone mode the AD574 facilitates the input of audio samples using the 6809 interrupt. This may be necessary in a situation where the sampling rate is faster than the A/D converter's conversion time. By using the 6809's SYNC instruction[6], the 6809 can be made to wait for an end-of-conversion signal from the STS pin of the AD574. Figure 5 shows such a program, which stores 100 16-bit samples from the AD574 in the Apple II memory. During PIA initialization the PIA interrupt is enabled. The sampling routine takes 25 clock cycles, which may be faster than the conversion rate in certain circumstances. (The specified conversion rate for the AD574 is 15–35 μs.)

To initiate the 6809 interrupt properly, the 6809 coprocessor board had to be modified slightly. The 6809 $\overline{IRQ}$ line is fed by the output of one of the bits of the onboard control register described earlier. To enable the SYNC instruction to respond to an incoming interrupt request, the $\overline{IRQ}$ line from the Apple II bus was directly connected to the 6809's IRQ pin. This enables the 6809 to respond to the AD574 end-of-conversion signal through the STS and CA1 lines and thus to transfer the 16-bit samples to the Apple II memory. Strictly speaking, for the program to work properly, the loop routine delay should always be less then the A/D conversion time.

CONCLUSIONS

A programmable 12-bit audio time delay system using the 6809 for the audio processing has been successfully

```
2000                    ORG     $2000       6809 DATA ACQUISITION PROGRAM
2000 4F         START   CLRA                PIA INITIALIZATION
2001 5F                 CLRB
2002 FDC0B2             STD     $C0B2
2005 FDC0B0             STD     $C0B0       DEFINE PIA INPUT PORTS
2008 8625               LDA     #$25        ENABLE PIA INTERRUPT
200A C604               LDB     #$04
200C FDC0B2             STD     $C0B2
200F 1A10               ORCC    #$10        MASK 6809 INTERRUPT
2011 8E6000             LDX     #$6000      START OF MEMORY BLOCK
2014 C6C8               LDB     #$C8        NO OF INPUT SAMPLES
2016 10BEC0B0           LDY     $C0B0       DUMMY LOAD TO INITIATE FIRST SAMPLE
201A 13         LOOP    SYNC                WAIT FOR 6809 INTERRUPT REQUEST
201B 10BEC0B0           LDY     $C0B0       ACQUIRE ONE 12-BIT SAMPLE
201F 10AF81             STY,X++
2022 5A                 DECB
2023 26F5               BNE     LOOP        REPEAT IF NOT END
2025 1CEF               ANDCC   #$EF        CLEAR 6809 INTERRUPT MASK
        2000            END     START
    2 SYMBOLS IN TABLE:

LOOP   $201A  START $2000

SYMBOL TABLE END: 40F9

    0 STATEMENT ERROR(S), LAST PC:2026
```

Figure 5. Data sampling program for 6809

demonstrated. The 6809 was able to handle the audio throughput at a rate sufficient for high-quality audio applications. The system could be used for 16-bit processing with only a change in the A/D and D/A converters from 12 bit to 16 bit. The system could be further improved by using a more powerful microprocessor such as the 68008. Further work using a 68008 coprocessor is currently under way.

REFERENCES

1 **Stajic, D** 'Microprocessor-based simulator of a variable time delay' *Microprocessors Microsyst.* Vol 6 No 2 (March 1982) pp 73–75

2 *Apple II reference manual* Apple Computers, Cupertino, CA, USA

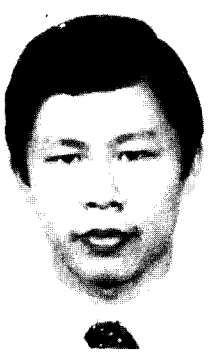
3 *Analog Devices data acquisition databook 1984, Vol 1: integrated circuits* Analog Devices, Norwood, MA, USA (1984)

4 **Leventhal, L A** *Introduction to microprocessors: software, hardware, programming* Prentice-Hall, Englewood Cliffs, NJ, USA (1978)

5 *The Mill — principles of operation* (manual for 6809 coprocessor board) Stellation Two, USA (1981)

6 **Leventhal, L** *6809 assembly language programming* Osborne/McGraw-Hill, New York, USA (1981)

Bernard Tan graduated in 1965 with an honours degree in physics from the University of Singapore, and in 1968 with a DPhil from Oxford University, UK. He is a chartered engineer and a member of the IEE and the IERE. Since 1968 he has taught at the National University of Singapore where he is now an associate professor in physics. His research interests include microwave properties of semiconductors, computer-assisted acoustical analysis and synthesis, and microprocessor applications.



L P Tay obtained an honours degree in physics from the National University of Singapore in 1984. He is now teaching physics at River Valley High School, Singapore. He is also involved in part-time research in solid-state physics, working towards an MSc degree, at the National University of Singapore.