John Bell Engineering's Apple II Parallel Interface Board

Ned W. Rhodes
2001 North Kenilworth
Arlington, VA 22205

One reason I bought an Apple II was the potential for expansion on its motherboard. I'd planned to add a parallel I/O (input/output) port, a real-time clock, and a couple of other items I was going to design and build. After working with the board for two years, though, I concluded that buying one that already had these features would put me ahead of the game.

Fortunately, I discovered that John Bell Engineering produces an Apple II parallel interface board—actually a multifunction module. It contains two 6522 Versatile Interface Adapters (VIAs) and can function as a parallel interface, clock, or counter. To explain the capabilities of the card, I need only elaborate on the capabilities of the 6522 chip.

About the Author

Ned Rhodes earned a BEE from the University of Minnesota and a master's in computer science from George Washington University. He presently develops minicomputer-based distributed processing systems for the MELPAR division of E-Systems Inc. in Falls Church, Virginia.

The 6522 VIA

The 6522 is a 40-pin support chip compatible with the 6502 microprocessor family. The chip is designed for connection to the data and address bus of a 6502 microprocessor, and it provides two bidirectional, 8-bit I/O ports (where the direction of each bit is programmable). In addition to the parallel ports, each 6522 has two 16-bit, fully programmable clocks that can be used as counters or interval timers. The chip also includes a shift register for use with one of the timers to clock serial data into or out of the 6522. Each 6522 fully supports the 6502 interrupt structure, finally allowing you to constructively use

At a Glance

Name
Apple II Parallel Interface

Use
Board may be used for parallel I/O, timing, or serial-to-parallel/parallel-to-serial conversions

Manufacturer
John Bell Engineering
POB 338
Redwood City, CA 94064
(415) 367-1137

Dimensions
3 inches by 5 (7.5 by 12.5 cm); plugs into any Apple slot

Price
Assembled, $69.95; kit, $59.95; board only, $22.95

Features
Board contains two 6522 Versatile Interface Adapters with a total of four 8-bit, bidirectional I/O data ports; eight I/O control lines; four independent, 16-bit timers; and two 8-bit, serial-to-parallel/parallel-to-serial shift registers. User can choose the IRQ or NMI interrupt lines

Software Needed
All user-written—no software provided

Documentation
A 16-page booklet containing a circuit board description and a 6522 data sheet

Audience
Assembly-language programmers and others with some hardware experience

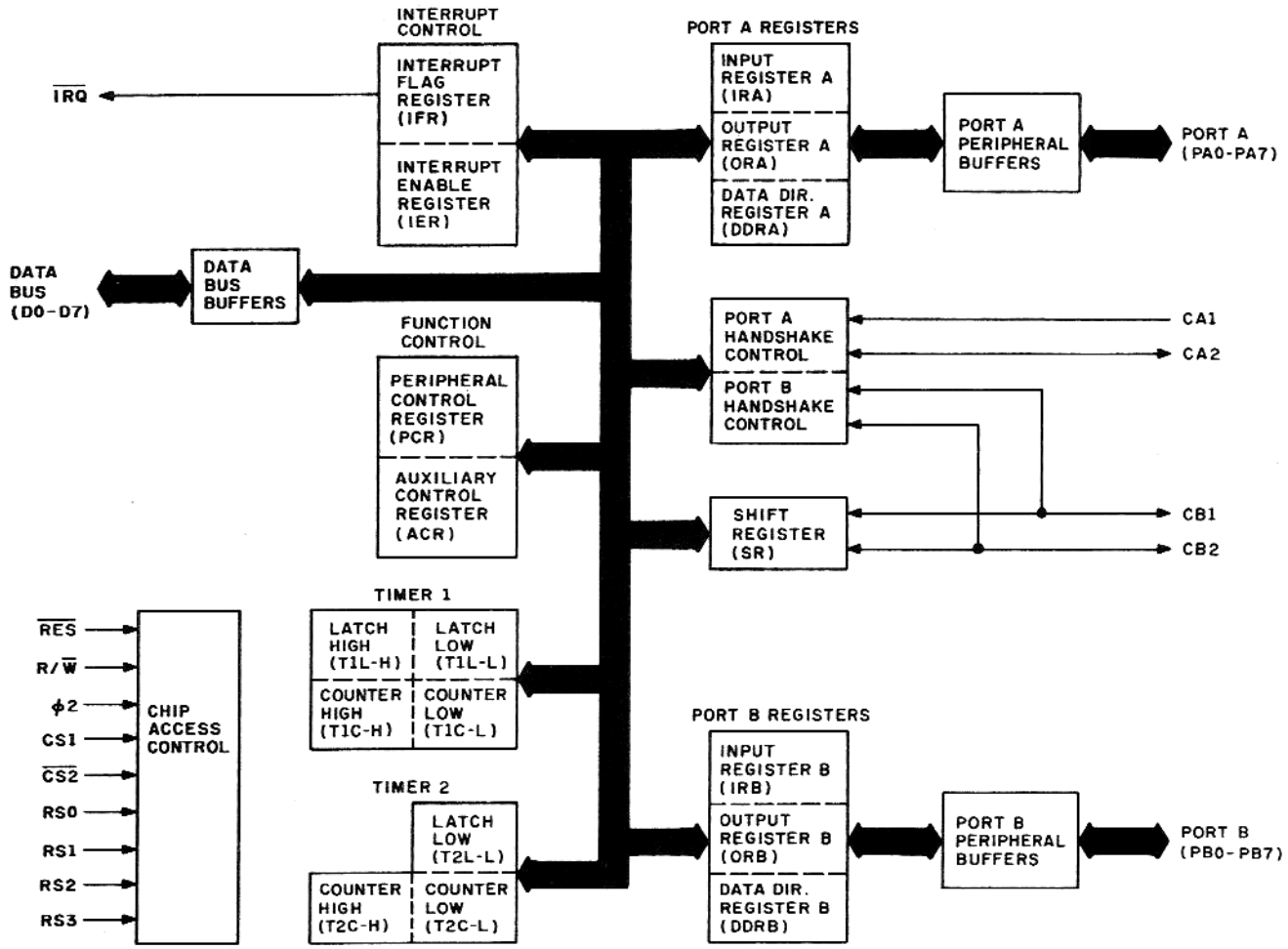

Figure 1: *Block diagram of the internal configuration of the 6522 VIA (Versatile Interface Adapter) integrated circuit.*

the Apple interrupts.

All communication with the 6522 occurs through 16 internal registers. Two of the 16, IRB/ORB and IRA/ORA, are used as I/O registers for the two 8-bit parallel ports. Two others, DDRB and DDRA, are data-direction registers that define the direction of each bit (either input or output) of the parallel ports. Four registers are set aside to control the two programmable counter/timers, and one I/O register controls the serial-shift register. Two registers select the operating mode of the timers and shift register; they also determine whether the chip will recognize positive- or negative-going control pulses.

The 6522 has a dedicated interrupt flags register that allows the chip to generate interrupts upon detection of (1) a positive- or negative-going edge on any of the four control lines, (2) a timeout (overflow) condition on either of the timers, or (3) the completion of a shift-register shift cycle. One register selectively enables and disables interrupt generation, while the last register is reserved for special forms of I/O through port A. Figure 1 is a block diagram of the 6522 chip's internal layout.

6522 on the Apple

Due to a design limitation in the Apple II, the 6522 can't work properly if it's merely attached to the bus; the 6522 requires a phase 2 clock pulse that isn't available on the Apple. The Apple 6502 processor generates the phase 2 clock signal, but that pin is unavailable at the expansion slot connectors.

Therefore, the I/O board must generate its own phase 2 clock signal. The phase 2 clock pulse is simulated by delaying the phase 0 clock signal by 80 nanoseconds. I must point out that simply delaying phase 0 may not match the duty cycle specification of the phase 2 clock, but that doesn't seem to matter. The 6522s accept the simulated phase 2 clock signal and work just fine.

The Circuit Board

The board may be purchased in three different forms. For those of you with no hardware experience, it's available as a fully assembled and tested card. It may also be bought as a complete kit or as a bare board for which you supply the parts. I chose the bare board, then ordered the sockets and 6522 chips from a mail-order supplier.

The board is very simple to build.

All you do is mount two 40-pin sockets, four 16-pin sockets, one 14-pin socket, and two bypass capacitors. Then plug in the chips and you're ready to go. The documentation suggests that you use "standard assembly and soldering techniques." I guess that means you shouldn't lift the solder donuts by applying excessive heat and that solder bridges between pins are taboo. I managed to avoid both perils.

Connections are made through the four 16-pin DIP (dual in-line package) sockets; each socket handles eight bits. If interrupts are used, two jumper wires must be installed to enable them. One of the 6522s can be attached to the IRQ (interrupt-request) line, while the other can be attached to the NMI (nonmaskable interrupt) line. Note that the interrupt lines *cannot* be shared—you can have only one 6522 attached to an interrupt line.

Documentation accompanying the board is sparse, and the unadventurous user may get lost. The board comes with a two-page circuit diagram and register identification list, a two-page circuit description, and a two-page list of all possible board addresses (whose availability depends upon which slot is used in the Apple). A ten-page 6522 data sheet is also provided.

If you can read the data sheet, you can use the 6522. If you find the data sheet difficult to understand, chances are this product isn't for you. The manufacturer has provided no software examples because of "the numerous uses of the board." I believe that limits the board's usefulness. Hold on, though; I've provided two software routines to demonstrate the capabilities of the parallel interface board and the 6522s.

Software

I was unable to write software that would test *all* of the 6522 functions, so I chose two of the more common applications: parallel I/O and clocks.

Parallel Printer Routine. The first software example, in listing 1, is a parallel output routine for a printer such as the Epson MX-80. Two basic sections comprise the routine. In the first section, the output routine is "hooked" through DOS, so that any character output to the screen will also appear on the printer. The horizontal-tab counter and the screen-echo flag are initialized at this time. The 6522 is then set up for output, and a Control X is sent to the printer, clearing its internal buffer. In the routine's second section, characters are output, one at a time, to the printer.

The 6522 initialization is unique. First, you enable port A for output by placing a hexadecimal FF in the data-direction register (DDRA) for port A. Then set up the data-output strobe and the data-ready flag, which are the handshaking signals required for parallel communications. When the printer is ready to receive data, it indicates this with a pulse. With the MX-80, a negative-going pulse indicates ready, so you tie it to the CA1 line (one of the control lines for port A). The other signal, the data strobe,

Text continued on page 428

Listing 1: *Parallel printer output routine for the John Bell Engineering parallel board. Written in 6502 assembly language, this program is designed to drive an Epson MX-80 printer.*

```
                    1000 *
                    1010 *
                    1020 * MX-80 PRINTER DRIVER
                    1030 *  DJG 3/81
                    1040 *  MODIFIED BY NWR FOR USE WITH JOHN BELL CARD
                    1050 *
                    1060 * NORMAL MODE ECHOES ON SCREEN
                    1070 * CNTL-H (BACK ARROW) ABORTS SCREEN ECHO SO
                    1080 * BASIC LISTINGS WILL USE FULL 80 COLUMNS
                    1090 * CNTL-G (BELL) IS ALWAYS SENT TO SCREEN ONLY
                    1100 *
                    1110 *
                    1120 *     6522 REGISTER EQUATES
                    1130 *
                    1140 *
C300-               1150 SLT1 .EQ $C300     SLOT 3 6522 REGISTERS
C301-               1160 ORA  .EQ SLT1+1    OUTPUT REGISTER A
C303-               1170 DDRA .EQ SLT1+3    DATA DIRECTION REGISTER
C30C-               1180 PCR  .EQ SLT1+$C   PERIPHERAL CONTROL REGISTER
C30D-               1190 IFR  .EQ SLT1+$D   INTERRUPT FLAG REGISTER
                    1200 *
                    1210 *
                    1220 *     PROGRAM EQUATES
                    1230 *
                    1240 *
0024-               1250 CH   .EQ $24       SCREEN HOR. CURSOR POSITION
0479-               1260 HCNT .EQ $0479     CHARACTER COUNTER IN SLOT 1 RAM
04F9-               1270 FLAG .EQ $04F9     FLAG FOR ECHOING ON SCREEN
0579-               1280 ACC  .EQ $0579     ACCUMULATOR STORAGE
0011-               1290 ENBL .EQ $11       ENABLE PRINTER CHARACTER
0098-               1300 CNTX .EQ 152       CONTROL X CHARACTER
FDF0-               1310 SCRN .EQ $FDF0     SCREEN SUBROUTINE
0036-               1320 DHOK .EQ $36       OUTPUT HOOK LOCATION
03EA-               1330 HOOK .EQ $3EA      DOS HOOK ROUTINE
0087-               1340 BELL .EQ $87       BELL CHARACTER
0008-               1350 SLNT .EQ $8        CONTROL-H  ECHO ON PRINTER ONLY
000D-               1360 CAR  .EQ $D        CARRIAGE RETURN
00A0-               1370 SPAC .EQ $A0       SPACE CHARACTER
                    1380 *
                    1390 *
                    1400      .OR $0300     START IT HERE
                    1410      .TF MX80.BELL.OBJ
                    1420 *
                    1430 *
                    1440 *     MX80 INITIALIZATION
                    1450 *
                    1460 *     CALL THIS PORTION OF THE ROUTINE TO
                    1470 *     SET UP THE PRINTER DRIVER TO SEND
                    1480 *     CHARACTERS TO THE PRINTER
                    1490 *
                    1500 *
0300- A9 FF         1510 MX80 LDA #$FF      SET PORT A FOR
0302- 8D 03 C3      1520      STA DDRA      OUTPUT
0305- A9 0A         1530      LDA #$0A      CA1 ON NEGATIVE EDGE
0307- 8D 0C C3      1540      STA PCR       CA2 GIVES PULSE
030A- A9 11         1550      LDA #ENBL     ENABLE PRINTER (SO IFR1 WILL
030C- 8D 01 C3      1560      STA ORA       BE SET FOR FIRST CHARACTER)
030F- A9 98         1570      LDA #CNTX     CNTL X ERASES BUFFER
0311- 8D 79 05      1580      STA ACC       SAVE IT
0314- 20 69 03      1590      JSR POUT      SEND IT
0317- A9 2B         1600      LDA #PRNT     ADDRESS OF NEW OUTPUT ROUTINE
0319- 85 36         1610      STA DHOK      SAVE IT
031B- A9 03         1620      LDA /PRNT     ADDRESS OF NEW OUTPUT ROUTINE
031D- 85 37         1630      STA DHOK+1    SAVE IT
031F- 20 EA 03      1640      JSR HOOK      HOOK NEW OUTPUT ROUTINE TO DOS
0322- A9 00         1650      LDA #00       GET INITIAL VALUE
0324- 8D 79 04      1660      STA HCNT      ZERO HORIZONTAL CHARACTER POSITION
0327- 8D F9 04      1670      STA FLAG      SCREEN ECHO ON
032A- 60            1680      RTS           DONE WITH INITIALIZATION
                    1690 *
                    1700 *
                    1710 *     OUTPUT ROUTINE
                    1720 *
                    1730 *
032B- C9 87         1740 PRNT CMP #BELL     A BELL CHARACTER??
032D- F0 36         1750      BEQ TV        YES, AVOID PRINTER'S RACKET
032F- 29 7F         1760      AND #$7F      REMOVE MSB
0331- 48            1770      PHA           SAVE ACCUMULATOR
0332- C9 08         1780      CMP #SLNT     DISABLE SCREEN ECHO??
0334- D0 03         1790      BNE CR        NO
0336- 8D F9 04      1800      STA FLAG      PUT NON-ZERO IN FLAG
0339- C9 0D         1810 CR   CMP #CAR      CARRIAGE RETURN?
033B- D0 05         1820      BNE TAB       NO
033D- A9 FF         1830      LDA #$FF      COUNTER WILL BE ZERO AFTER CR
033F- 8D 79 04      1840      STA HCNT      RELOAD HORIZONTAL CHARACTER-1
0342- AD 79 04      1850 TAB  LDA HCNT      COMPARE COUNTER WITH SCREEN
0345- C5 24         1860      CMP CH        HOR. POSITION
0347- B0 0B         1870      BCS CHAR      BRANCH IF IN PROPER POSITION
                    1880 *
                    1890 *
                    1900 *     OUTPUT SPACES UNTIL PRINTER IS AT THE PROPER
                    1910 *     HORIZONTAL CHARACTER POSITION
                    1920 *
                    1930 *
0349- A9 A0         1940      LDA #SPAC     GET A SPACE
034B- 8D 79 05      1950      STA ACC       SAVE AS PRINT CHARACTER
034E- 20 69 03      1960      JSR POUT      PRINT IT
0351- 4C 42 03      1970      JMP TAB       CHECK HOR. POSITION AGAIN
0354- 68            1980 CHAR PLA           GET CHARACTER
0355- 8D 79 05      1990      STA ACC       SAVE AS PRINT CHARACTER
0358- 20 69 03      2000      JSR POUT      GO PRINT IT
035B- AD F9 04      2010      LDA FLAG      ECHO ON SCREEN?
035E- D0 08         2020      BNE RET       NO
```

Listing 1 continued on page 422

Listing 1 continued:

```
0360- AD 79 05 2030       LDA ACC     GET CHARACTER AGAIN
0363- 09 80    2040       ORA #$80    SET MSB FOR SCREEN
0365- 4C F0 FD 2050 TV    JMP SCRN    OUTPUT TO SCREEN
0368- 60       2060 RET   RTS         NORMAL RETURN
               2070 *
               2080 *
               2090 *     POUT SUBROUTINE
               2100 *
               2110 *     HANDLES OUTPUT TO PRINTER
               2120 *     CHARACTER TO BE PRINTED IS IN ACC
               2130 *
               2140 *
0369- A9 02    2150 POUT  LDA #02     LOAD COMPARE MASK
036B- 2C 0D C3 2160       BIT IFR     IS PRINTER READY??
036E- F0 F9    2170       BEQ POUT    NO, WAIT
0370- AD 79 05 2180       LDA ACC     GET CHARACTER TO PRINT
0373- 8D 01 C3 2190       STA ORA     PRINT IT
0376- EE 79 04 2200       INC HCNT    BUMP CHAR. COUNTER
0379- 60       2210       RTS         RETURN
               2220       .EN

SYMBOL TABLE

0579- ACC
0087- BELL
000D- CAR
0024- CH
0354- CHAR
0098- CNTX
0339- CR
C303- DDRA
0036- DHOK
0011- ENBL
04F9- FLAG
0479- HCNT
03EA- HOOK
C30D- IFR
0300- MX80
C301- ORA
C30C- PCR
0369- POUT
032B- PRNT
0368- RET
FDF0- SCRN
0008- SLNT
0300- SLT1
00A0- SPAC
0342- TAB
0365- TV
```

Listing 2: *This routine uses the parallel board as a real-time clock. The time will be continuously displayed on the screen.*

```
*** SYNTAX ERROR

:ASM

1000 *
1010 *
1020 *      DOSCLOCK -- REAL TIME CLOCK WITH CORRECTION
1030 *      FOR DISK USE.  THIS PROGRAM USES THE JOHN BELL
1040 *      ENGINEERING PARALLEL BOARD AS A REAL TIME DIGITAL
1050 *      CLOCK.
1060 *
1070 *      NOTE -- THIS ROUTINE IS ASSEMBLED TO START NEAR
1080 *      THE END OF THE INPUT LINE BUFFER.  THIS MEANS THAT
1090 *      YOU WILL NOT BE ABLE TO ENTER REALLY LONG LINES
1100 *       OF TEXT.  THIS ROUTINE IS NOT COMPATABLE WITH
1110 *      THE PROGRAM LINE EDITOR BECAUSE THAT PROGRAM USES
1120 *      LOCATION $45 WHICH IS WHERE THE MONITOR STUFFS
1130 *      THE ACCUMULATOR DURING INTERRUPT PROCESSING.
1140 *
1150 *      BECAUSE THE 6522 IS ATTACHED TO THE APPLE
1160 *      BUSS, THE PUSHING OF THE RESET BUTTON WILL RESET THE
1170 *      6522 AS WELL AS THE APPLE.  THAT MEANS THAT YOU WILL
1180 *      HAVE TO RESTART THE CLOCK EVERY TIME RESET IS PUSHED.
1190 *
1200 *      IN ORDER TO USE THIS ROUTINE WITH THE JBE BOARD, YOU
1210 *      HAVE TO ENABLE TIMER T2 TO COUNT THE NUMBER OF TICKS
1220 *      OF TIMER T1.  THIS IS ACCOMPLISHED BY JUMPERING PINS
1230 *      7 AND 8 ON J2 TOGETHER
1240 *
1250 *
1260 *
1270 * RAM VERSION FOR SLOT 3
1280 *
1290 *   DJG  4/81 (DERIVED FROM AN AIM CLOCK ROUTINE
1300 *              BY DEJONG IN MICRO)
1310 *   NWR  8/81 (MODIFIED FOR SLOT 3 AND THE
1320 *              JBE PARALLEL BOARD)
1330 *
1340 *
1350 * PROGRAM ADDRESSES:
1360 *
1370 *    ENTRY POINT (TO START CLOCK):
1380 *        $0280      (CALL 640)
1390 *
1400 *    TO CONTROL SCREEN TIME DISPLAY:
1410 *        $7F      (POKE 1919,X)
1420 *        (NON-ZERO VALUE DISPLAYS TIME CONTINUOUSLY)
1430 *
1440 *    TO STOP CLOCK:
1450 *
1460 *     LOAD LOCATION $C30E WITH $40
1470 *     (POKE -15602,64)
1480 *
1490 *
```

Listing 2 continued on page 424

Listing 2 continued:

```
                    1500 *      PROGRAM EQUATES
                    1510 *
                    1520 *
0045-               1530 ACC  .EQ $0045   MONITOR SAVE ACC HERE ON IRQ
0420-               1540 SCRN .EQ $0420   RIGHT HAND TOP LINE OF SCREEN
047F-               1550 FREE .EQ $047F   SLOT 7 SCRATCH RAM -- SEE APPLE REFERENCE MANUAL
047F-               1560 FRAC .EQ FREE    INTERRUPT COUNTER
04FF-               1570 SEC  .EQ FREE+$80  SECONDS COUNTER
057F-               1580 MIN  .EQ FREE+$100 MINUTES COUNTER
05FF-               1590 HOUR .EQ FREE+$180 HOURS COUNTER
067F-               1600 TMPL .EQ FREE+$200 TEMP STORAGE
06FF-               1610 TMPH .EQ FREE+$280 TEMP STORAGE
077F-               1620 FLAG .EQ FREE+$300 DISPLAY FLAG
03FE-               1630 IRQV .EQ $03FE     IRQ VECTOR
                    1640 *
                    1650 *
                    1660 *      6222 REGISTER EQUATES
                    1670 *
                    1680 *
C300-               1690 DS1  .EQ $C300  SLOT 3 6522 ADDRESSES
C300-               1700 PB   .EQ DS1    PORT B
C304-               1710 T1L  .EQ DS1+4  TIMER 1 LOW LATCH
C305-               1720 T1H  .EQ DS1+5  T1 HIGH
C309-               1730 T2L  .EQ DS1+8  TIMER 2 LOW LATCH
C309-               1740 T2H  .EQ DS1+9  T2 HIGH
C30B-               1750 ACR  .EQ DS1+$B CONTROL REG
C30D-               1760 IFR  .EQ DS1+$D INTERRUPT FLAGS
C30E-               1770 IER  .EQ DS1+$E INTERRUPT ENABLE
                    1780 *
                    1790 *
                    1800      .OR $0280  START IT HERE
                    1810      .TF DOSCLOCK.BELL.OBJ
                    1820 *
                    1830 *
                    1840 *      CLOCK ENTRY POINT
                    1850 *
                    1860 *
                    1870 *      SET UP IRQ VECTOR AND START THE CLOCK
                    1880 *
                    1890 *
0280- 78            1900 CLOK SEI        DISABLE IRQ
0281- A9 B0         1910      LDA #ISR   ADDRESS OF INTERRUPT SERVICE ROUTINE
0283- 8D FE 03      1920      STA IRQV   SAVE IN IRQ
0286- A9 02         1930      LDA /ISR   LAST HALF OF ADDRESS
0288- 8D FF 03      1940      STA IRQV+1 SAVE IT
                    1950 *
                    1960 *
                    1970 *      SET UP THE 6522
                    1980 *
                    1990 *
028B- A9 C0         2000      LDA #$C0   ENABLE T1 INTERRUPT
028D- 8D 0E C3      2010      STA IER    BY LOADING THIS LOCATION
0290- A9 E0         2020      LDA #$E0   T1 FREE RUN MODE
0292- 8D 0B C3      2030      STA ACR    AND T2 COUNTS PB6
                    2040 *
                    2050 *
                    2060 *      YOU CAN PLAY WITH THE VALUE IN T1L AND T1H TO
                    2070 *      CORRECT TIME INACCURACIES.
                    2080 *
                    2090 *
0295- A9 20         2100      LDA #$20   SET T1 TO F920
0297- 8D 04 C3      2110      STA T1L    WHICH IS 1/16TH OF
029A- A9 F9         2120      LDA #$F9   A SECOND
029C- 8D 05 C3      2130      STA T1H    START T1
029F- A9 08         2140      LDA #$08   T2 OVERFLOWS AFTER 1 SECOND IF
02A1- 8D 08 C3      2150      STA T2L    DOS TURNED OFF IRQ INTERRUPT
02A4- A9 00         2160      LDA #00    $800 IS ONE SECOND
02A6- 8D 09 C3      2170      STA T2H    T2 COUNTS T1
02A9- A9 F0         2180      LDA #$F0   COUNT 16 INTERRUPTS
02AB- 8D 7F 04      2190      STA FRAC   PRELOAD THE LOCATION
02AE- 58            2200      CLI        ENABLE IRQ
02AF- 60            2210      RTS        RETURN
                    2220 *
                    2230 *
                    2240 * INTERRUPT SERVICE ROUTINE
                    2250 *
                    2260 *
02B0- EE 7F 04      2270 ISR  INC FRAC   BUMP COUNTER
02B3- D0 70         2280      BNE UNDO   NOT A FULL SECOND
02B5- A9 F0         2290      LDA #$F0   RESET INTERRUPT COUNTER
02B7- 8D 7F 04      2300      STA FRAC   SAVE IT HERE
02BA- AD 00 C3      2310      LDA PB     IF T1 HAS ALREADY TICKED
02BD- 10 03         2320      BPL TOUT   WE HAVE TO ADD
02BF- EE 7F 04      2330      INC FRAC   ONE TO THE COUNT
02C2- A9 20         2340 TOUT LDA #$20   GET MASK
02C4- 2C 0D C3      2350      BIT IFR    T2 TIMED OUT??
02C7- D0 62         2360      BNE CORR   NO
02C9- A9 08         2370      LDA #$08   RESET T2 COUNTER
02CB- 8D 08 C3      2380      STA T2L    WITH A $800
                    2390 *
                    2400 *
                    2410 *      INCREMENT THE SOFTWARE CLOCK
                    2420 *
                    2430 *
02CE- EE FF 04      2440 SECS INC SEC    BUMP SECONDS
02D1- AD FF 04      2450      LDA SEC    GET CURRENT SECONDS
02D4- C9 3C         2460      CMP #60    60 SECONDS??
02D6- 90 23         2470      BCC SHOW   NO
02D8- A9 00         2480      LDA #00    RESET SECONDS
02DA- 8D FF 04      2490      STA SEC    TO ZERO
02DD- EE 7F 05      2500 MINS INC MIN    BUMP MINUTES
02E0- AD 7F 05      2510      LDA MIN    GET CURRENT MINUTES
02E3- C9 3C         2520      CMP #60    60 MINUTES??
02E5- 90 14         2530      BCC SHOW   NO
02E7- A9 00         2540      LDA #00    RESET MINUTES
02E9- 8D 7F 05      2550      STA MIN    TO ZERO
02EC- EE FF 05      2560 HRS  INC HOUR   BUMP HOURS
02EF- AD FF 05      2570      LDA HOUR   GET CURRENT HOUR
02F2- C9 18         2580      CMP #24    24 HOURS??
02F4- 90 05         2590      BCC SHOW   NO
02F6- A9 00         2600      LDA #00    RESET HOURS
02F8- 8D FF 05      2610      STA HOUR   TO ZERO
                    2620 *
                    2630 *
                    2640 *      DISPLAY THE TIME IF DESIRED
```

Listing 2 continued on page 426

Listing 2 continued:

```
                    2650 *
                    2660 *
02FB- AD 7F 07      2670 SHOW  LDA FLAG   DISPLAY TIME??
02FE- F0 20         2680       BEQ DONE   NO
0300- 8E 7F 06      2690       STX TMPL   SAVE X
0303- 8C FF 06      2700       STY TMPH   SAVE Y
0306- A2 00         2710       LDX #00    CLEAR X
0308- AD FF 05      2720       LDA HOUR   GET THE CURRENT HOUR
030B- 20 94 03      2730       JSR DSPL   PRINT IT
030E- AD 7F 05      2740       LDA MIN    GET CURRENT MINUTE
0311- 20 7C 03      2750       JSR DCOL   PRINT IT
0314- AD FF 04      2760       LDA SEC    GET CURRENT SECONDS
0317- 20 7C 03      2770       JSR DCOL   DISPLAY IT
031A- AE 7F 06      2780       LDX TMPL   RESTORE X
031D- AC FF 06      2790       LDY TMPH   RESTORE Y
                    2800 *
                    2810 *
                    2820 *       INTERRUPT DONE
                    2830 *
                    2840 *
0320- A9 00         2850 DONE  LDA #00    GET A ZERO
0322- 8D 09 C3      2860       STA T2H    START T2
0325- AD 04 C3      2870 UNDO  LDA T1L    CLEAR INTERRUPT FLAG
0328- A5 45         2880       LDA ACC    RESTORE ACCUM.
032A- 40            2890       RTI        INTERRUPT RETURN
                    2900 *
                    2910 *
                    2920 *       CORRECTION TO TIME WHEN DOS TURNED OFF IRQ
                    2930 *
                    2940 *
032B- 38            2950 CORR  SEC        SAVE 2'S COMPLEMENT OF T2
032C- A9 00         2960       LDA #00    BY SUBTRACTING
032E- ED 08 C3      2970       SBC T2L    FROM ZERO
0331- 8D 7F 06      2980       STA TMPL   SAVE PARTIAL RESULT
0334- A9 00         2990       LDA #00    ANOTHER ZERO
0336- ED 09 C3      3000       SBC T2H    AND THE SUBTRACT
0339- 8D FF 06      3010       STA TMPH   SAVE IT
                    3020 *
                    3030 *
                    3040 *       DO THE CORRECTION
                    3050 *
                    3060 *
033C- AD 7F 06      3070 SETF  LDA TMPL   SET FRACTION
033F- 29 07         3080       AND #$07   TO CORRECT
0341- 0A            3090       ASL        1/16
0342- 6D 7F 04      3100       ADC FRAC   OF A SEC
0345- 8D 7F 04      3110       STA FRAC   SAVE IT BACK
0348- 29 0F         3120       AND #$0F   CORRECT T2
034A- 4A            3130       LSR        TO THE PARTIAL
034B- 49 FF         3140       EOR #$FF   NUMBER OF
034D- 18            3150       CLC        OF TICKS
034E- 69 09         3160       ADC #$09   LEFT IN ITS INTERVAL
0350- 8D 08 C3      3170       STA T2L    AND SAVE BACK
0353- 4E FF 06      3180       LSR TMPH   DIVIDE BY EIGHT
0356- 6E 7F 06      3190       ROR TMPL   TO GET NUMBER
0359- 4E FF 06      3200       LSR TMPH   OF FULL SECONDS
035C- 6E 7F 06      3210       ROR TMPL   TO ADD TO THE
035F- 4E FF 06      3220       LSR TMPH   TIME TO CORRECT
0362- 6E 7F 06      3230       ROR TMPL   FOR DOS BEING ON
0365- 18            3240       CLC        SETUP THE CARRY
0366- AD FF 04      3250       LDA SEC    ADD THE FULL
0369- 69 01         3260       ADC #01    SECONDS
036B- 6D 7F 06      3270       ADC TMPL   AND STORE IN
036E- 8D FF 04      3280       STA SEC    SECONDS COUNTER
0371- 38            3290       SEC        CHECK FOR GREATER
0372- E9 3C         3300       SBC #60    THAN 60 SECONDS
0374- 90 85         3310       BCC SHOW   TO SEE IF A MINUTE
0376- 8D FF 04      3320       STA SEC    UPDATE IS
0379- 4C DD 02      3330       JMP MINS   REQUIRED
                    3340 *
                    3350 *
                    3360 * DISPLAY SUBROUTINE
                    3370 *
                    3380 *
037C- A8            3390 DCOL  TAY        SAVE COUNT
037D- A9 BA         3400       LDA #$BA   GET A COLON
037F- 9D 20 04      3410       STA SCRN,X SHOW IT
0382- E8            3420       INX        BUMP COUNTER
0383- 98            3430       TYA        RESTORE COUNT
0384- A0 FF         3440 DSPL  LDY #$FF   DISPLAY TIME
0386- C8            3450 CNTY  INY        Y WILL COUNT BY 10
0387- 38            3460       SEC        SET CARRY
0388- E9 0A         3470       SBC #10    MINUS 10
038A- B0 FA         3480       BCS CNTY   GET RID OF TENS
038C- 69 0A         3490       ADC #10    RESTORE REMAINDER
038E- 48            3500       PHA        AND SAVE
038F- 98            3510       TYA        DISPLAY TENS DIGIT
0390- 09 B0         3520       ORA #$B0   MAKE IT ASCII
0392- 9D 20 04      3530       STA SCRN,X SHOW IT
0395- E8            3540       INX        BUMP X
0396- 68            3550       PLA        GET ONES DIGIT
0397- 09 B0         3560       ORA #$B0   MAKE IT ASCII
0399- 9D 20 04      3570       STA SCRN,X SHOW IT
039C- E8            3580       INX        BUMP THE COUNT
039D- 60            3590       RTS        RETURN
                    3600       .EN
```

SYMBOL TABLE

```
0045- ACC           0300- IFR           06FF- TMPH
C30B- ACR           03FE- IRQV          067F- TMPL
0280- CLOK          02B0- ISR           02CF- TOUT
0386- CNTY          057F- MIN           0325- UNDO
032B- CORR          02DD- MINS
037C- DCOL          0300- PB
0320- DONE          0420- SCRN
C300- DSI           04FF- SEC
0384- DSPL          02CE- SECS
077F- FLAG          033C- SETF
047F- FRAC          02FB- SHOW
047F- FREE          0305- T1H
05FF- HOUR          0304- T1L
02EC- HRS           0309- T2H
030E- IER           0308- T2L
```

ext continued from page 418

locks data into the printer's internal uffer. Again, the MX-80 requires a egative-going pulse for the data tr use control pin CA2 for this unction.

The 6522 allows you to choose a egative- or positive-going pulse for ither of two signals; inform the 6522 f the desired polarity by loading the eripheral Control Register (PCR). Vith the MX-80, hexadecimal 0A is he proper code. This bit pattern is etermined by consulting the coded alues on the data sheet. We enable he printer by sending it a Control Q hexadecimal 11) and then a Control to clear the internal buffer.

The actual output routine is quite imple. First, check the horizontal haracter position and compare it ith the current character position in he output line. If they differ, output paces until reaching the proper char- cter position. To print characters, heck bit 2 in the Interrupt Flag egister (IFR) to see if the printer has ent its data-ready flag. This bit will be set if the 6522 has detected a negative edge on control pin 1 (CA1), which is the ready line.

If the printer is busy or has yet to send the ready pulse, keep testing the bit until the printer is ready. When the printer is ready to receive data, store the character to be printed in the output register for port A. As you place the character in the output register, it's clocked into the printer's internal register because pin CA2 goes low and acts as the data strobe. The printer becomes busy while accepting the character. Once it's processed, the ready pulse is given and the printer will accept another character.

Time-of-Day Clock. Listing 2's routine is a time-of-day clock that continuously displays the time on the screen. The routine uses interrupts so that the clock runs while you develop and run BASIC programs. The routine is compatible with DOS 3.3; DOS disables the IRQ interrupt while it does I/O and then re-enables the interrupt when finished. (I haven't tried the routine with earlier DOS versions. If you plan to, back up your disks in the event of failure.)

My method of implementing the clock involves both timers on one 6522 and a couple of tricks. First, set up timer T1 to interrupt (tick) every $1/16$ second. Simultaneously, enable timer T2 to count the number of times that T1 ticks by simply installing a jumper wire to feed the output of T1 to the input of T2. T2 is now counting the number of times T1 ticks. If DOS turns off the IRQ interrupt (for I/O), when it is re-enabled T2 will contain the number of clock ticks you missed.

The interrupt service routine for the clock keeps the hours, minutes, and seconds in dedicated locations. Whenever the seconds count is changed, the top line of the screen is updated with the current time. The BASIC routine in listing 3 will set up the current time of day and protect the top line. From then on, time will be displayed continuously. The clock routine can determine execution times of routines or schedule other events at certain times during the day. Because no two Apples have identical time bases, some correction factors may have to be used. The listing indicates where to apply those factors.

st 3: *This BASIC routine will load and initialize the clock. It will also protect the n play.*

```
]R#0
]LIST
  10 REM
  20 REM
  30 REM  ROUTINE TO LOAD AND START THE DOSCLOCK
  40 REM
  50 REM
  60 PRINT "BLOAD DOSCLOCK.BELL.OBJ"
  70 POKE 34,0: CALL -936
  80 PRINT "THE CURRENT TIME IS -- >": POKE 34,2
  90 VTAB 10
100 INPUT "ENTER HH,MM,SS ",H,M,S
110 POKE 1535,H: REM  HOURS
120 POKE 1407,M: REM  MINUTES
130 POKE 1279,S: REM  SECONDS
140 CALL 640: REM  START THE CLOCK
150 POKE 1919,1: REM  DISPLAY TIME
160 END
```

Conclusions

- The Apple parallel board may be used for all interfacing projects where parallel I/O is needed or where timing or counting is required.
- The board contains two 6522 support chips for input or output, timing or counting, and serial-to-parallel/parallel-to-serial operations.
- The board is available fully assembled, as a kit, or alone. The kit is easy to build, but you must be able to read a circuit diagram.
- Documentation is sparse, though all required information for use of the 6522 is included. The manufacturer does *not* hold your hand, relying instead on the user community to publish software that uses the board.
- The Apple parallel board is a good, inexpensive way to enhance the Apple with the power of the 6522 Versatile Interface Adapter. ■