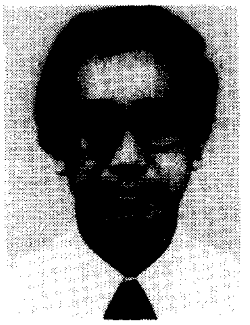
Standalone low-cost interface, controller and emulator for the Apple II

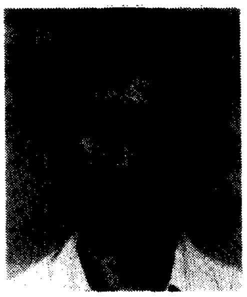
Not only does the interface card described by **B T G Tan and A K Tan** provide the Apple II with parallel ports, programmable timers, serial interface and an EPROM — it also acts as an independent microcomputer, once it has been interfaced to external devices

An I/O interface card for the Apple II microcomputer is described. The card provides a 6522 versatile interface adaptor and a 2716 EPROM. An onboard 6502 microprocessor and 1k RAM enable the interface card to be used independently of the Apple II as a standalone microcomputer controller. The card thus serves as a simple incircuit emulator for itself when it is in the Apple II.

microprocessors interface emulation



B T G Tan graduated in 1965 with a B Sc (Hons) in physics from the University of Singapore, and in 1968 with a D Phil 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.



A K Tan obtained a B Sc (Hons) in physics from the National University of Singapore in 1983. He is now an assistant systems engineer with Data People (S) Pte Ltd. He is also doing part-time research for an M Sc at the National University of Singapore.

Personal computers which have I/O slots for the insertion of plug-in interface cards are widely used in laboratory and other applications where realtime control and data acquisition are required. Two of the most popular personal computers used in this way are the Apple II and the IBM Personal Computer. The Apple II has an eight-slot I/O bus and 2k addresses reserved in the memory map for I/O use, while the IBM PC has a five-slot I/O bus, and a separate I/O address space for this bus.

In this paper we describe an I/O interface card for the Apple II. The card is designed to be plugged into one of the Apple II's slots. The card provides the Apple II with two parallel ports, two programmable timers, a simple serial interface and a 2k EPROM. The unique feature of the card is its ability to act as a standalone microcomputer independently of the Apple II, once the card has been interfaced to external equipment or devices. It can then be used as a controller or in any other application where a microcomputer of modest capability is required. The design principles adopted can be applied to any other microcomputer system with an I/O bus, such as the IBM PC.

APPLE II I/O STRUCTURE

We first review briefly the memory map and I/O structure of the Apple II[1]. The microprocessor used in the Apple II is the 8-bit 6502, which has a 64k address range; 2k addresses are allocated to I/O. The eight slots on the Apple II motherboard share these 2k addresses. The memory map is allocated as follows

from $0000 to $BFFF (48k)	RAM
from $C000 to $C7FF (2k)	I/O
from $C800 to $CFFF (2k)	user ROM
from $D000 to $FFFF (12 k)	system ROM

The addresses from $C000 to $C07F are allocated to the built-in I/O of the Apple II such as the keyboard, cassette interface, game paddles and loudspeaker. Each of the eight I/O slots has access to the system bus, which includes all the eight data lines, 16 address lines and control lines of the 6502. Each slot is allocated 16 I/O addresses as follows

Department of Physics, National University of Singapore, Singapore 0511

0141–9331/84/020067–06 $03.00

from $C080 to $C08F	slot 0 I/O
from $C090 to $C09F	slot 1 I/O
from $C0A0 to $C0AF	slot 2 I/O
from $C0B0 to $C0BF	slot 3 I/O
from $C0C0 to $C0CF	slot 4 I/O
from $C0D0 to $C0DF	slot 5 I/O
from $C0E0 to $C0EF	slot 6 I/O
from $C0F0 to $C0FF	slot 7 I/O

ie from $C080+*n*0 to $C08F+*n*0 where *n*(= 0–8) is the slot number.

Each slot has a $\overline{\text{Device select}}$ pin which is normally high. Accessing any one of the 16 addresses allocated to that slot will bring the voltage on the $\overline{\text{Device select}}$ pin low, thus enabling a peripheral card in that slot to be selected.

Each of the seven slots from 1 to 7 is also allocated 256 addresses for an onboard ROM. The addresses (from $C*n*00 to $C*n*FF) are allocated to ROM for slot *n*, where *n* varies from 1 to 7.

Each slot has an $\overline{\text{I/O strobe}}$ pin. This pin is normally high, but will be set low if any of the 256 addresses allocated for onboard ROM for that slot is accessed.

User ROM

The 2k addresses from $C800 to $CFFF are allocated for a user ROM. This user ROM may be a single 2k chip, eg a 2716 EPROM residing on an interface card in any of the slots. Each slot is also provided with an $\overline{\text{I/O select}}$ pin. This pin is normally high, but goes low when any of the addresses within the 2k range is selected. It is possible for two or more 2k ROMs to be simultaneously present in different slots. However, only one of the ROMs can be active at any time, since they would all be occupying the same range of addresses.

It is recommended[1] that a user ROM in any slot be enabled in the following way.

- The $\overline{\text{I/O select}}$ for that slot should be used to set a flipflop which partially enables the 2716 ROM.
- The $\overline{\text{I/O strobe}}$ should then be brought low to enable the ROM fully.

The address $CFFF is a special location which is used to disablę the ROM by resetting the flipflop. Hence this location cannot be used by programs residing on the ROM, except to disable it. Fewer address lines may be used to disable the ROM, but this entails a correspondingly greater number of unusable locations in the ROM.

6522 versatile interface adaptor

The 6800 and 6502 microprocessors have similar I/O features, such as memory-mapped I/O and a synchronous clock. There are many I/O chips which can be used by both microprocessors in common. Perhaps the most basic I/O chip is the 6821 peripheral interface adaptor (PIA). The 6821 provides two bidirectional I/O ports, each with eight data lines and two control lines.

A later improvement on the 6821 PIA is the 6522 versatile interface adaptor (VIA), which is in effect the combination of a 6821 PIA and two programmable timers. In addition, it has some rudimentary serial I/O facilities. The 6522 VIA is an extremely useful chip for general-purpose control and experimentation, since the built-in timer makes data input and output at precise time intervals much easier to program. The 6522 occupies 16 memory addresses, which fit exactly the 16 addresses allotted to each Apple II slot for I/O.

I/O INTERFACE CARD WITH ONBOARD 6502

Interface cards for the Apple II which provide I/O facilities of various kinds are widely available. In particular, cards which provide a 6522 VIA are easy to obtain, either as ready-made cards or as published designs (eg by Hofacker and Ekkehard[2]). Cards which provide a 2716 EPROM are also available. Hence it was decided to design an interface card which would carry both a 6522 and a 2716 and thus provide both I/O and EPROM on a single card.

An interface card with a 6522 and a 2716 would enable an Apple II (or any microcomputer with I/O slots, eg an IBM PC) to perform data acquisition and control in a laboratory or industrial situation, with a dedicated application program in EPROM. However, many such applications require much less RAM than the Apple II actually provides. In these situations a dedicated microcomputer card with its own microprocessor, I/O, EPROM and RAM would be a cheaper solution.

The interface card we describe can be made to work as an independent microcomputer with the addition of an onboard 6502 and RAM. Initially a dedicated application can be set up and interfaced to the card in the Apple II slot, using the Apple II's own 6502 and RAM. The application program can be written, burnt into the EPROM, tested and debugged while the card is in the Apple II. When the program has been debugged and is working properly, the card can be taken out of the Apple II slot. The onboard 6502 and RAM can then take over, making the card an independent microcomputer for controller or other dedicated applications.

The card can still be used as a normal interface card when it is plugged into an Apple II slot. The onboard 6502 and RAM should be disabled whenever the card is in a slot.

The interface card described here is a combination of an Apple II I/O card (when in the Apple II slot) and an independent dedicated microcomputer (when out of the Apple II slot). In a sense, the card is a microcomputer for which debugging is performed by a rudimentary form of incircuit emulation while the card is in the Apple II slot.

Design of the interface card

Our design concept is not uniquely applicable to the Apple II. It can be applied to any microcomputer system which has an I/O bus structure, such as the IBM PC. In the case of the Apple II, design and construction was easy because of the availability of Apple II interface prototyping boards and the Apple II bus structure.

The final design of the card uses eight integrated circuits

- 6522 VIA
- 2716 2k EPROM
- 6502 microprocessor unit (MPU)
- 2114 1k x 4 RAM (two integrated circuits)
- 7400 quad two-input nand gates
- 7404 hex inverter
- 7430 eight-input nand gate
- 74123 dual monostable multivibrator

It was decided that 1 kbyte of onboard RAM was ample for most dedicated applications. The 2114 RAM was chosen since it is easily available. A block diagram of the design is given in Figure 1.

The addressing scheme of the card was partially deter-

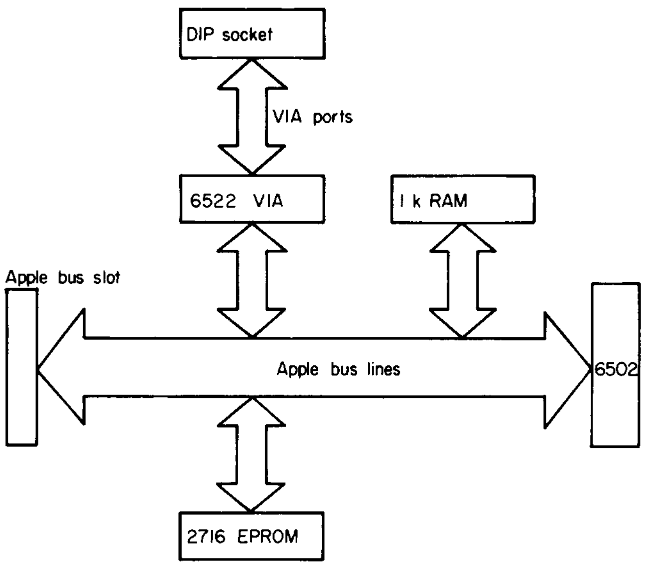


Figure 1. Interface card design — block diagram

mined by the I/O address map of the Apple II. The 6522 and 2716 are accessible to the 6502 of the Apple II while the card is plugged into the slot. When the card is out of the slot, the onboard 6502 is able to address these two integrated circuits. In addition the onboard 2114 RAM is only accessible to the onboard 6502, and never to the 6502 of the Apple II.

Bus contention may arise between the onboard 6502 and the 6502 of the Apple II while the card is in the slot. This must be prevented, since the 6502s share the MPU buses on the card. One way is by removing the 6502 from the card while in the slot. In practice, it was found sufficient to disconnect the power supply of the onboard 6502.

Onboard address decoding

The onboard 6502 addresses the 6522, 2716 and 2114s through the addressing scheme shown in Table 1. This partial address decoding scheme will result in the integrated circuits occupying the following address ranges

2716	from \$*p*800 to \$*p*FFF where *p*=8–F
6522	from \$*pqr*0 to \$*pqr*F where *p*=8–F, *q*=0–7 and *r*=0–F

2114s	from \$*q*000 to \$*q*3FF or
	from \$*q*400 to \$*q*7FF or
	from \$*q*800 to \$*q*BFF or
	from \$*q*C00 to \$*q*FFF where *q*=0–7

In the Apple II, the 2716, 6522 and RAM are allocated the following addresses

2716	from \$C800 to \$CFFF
6522	from \$C080+*n*0 to \$C08F+*n*0 where *n* (=0–7) is the slot number
RAM	\$0000 to \$BFFF

Suppose an application program is being written for the 2716 EPROM. If the 2716 and 6522 are correctly addressed while in the Apple II, they will also be correctly addressed by the onboard 6502 because the Apple II addresses fall within the onboard addresses. In the Apple II free RAM space is limited to addresses between \$0C00–\$1FFF and \$6000–\$BFFF, because of requirements for system use and the graphics and text display. The RAM addresses which are used by the program should fall within the acceptable ranges for both the Apple II and the onboard 6502. For example, a suitable range would be \$6000–\$63FF which is just above the high-resolution graphics page two of the Apple II.

It should also be noted that, for the 6502, page zero of the memory map should be available for zero-page addressing. Furthermore, page one (from \$0100 to 01FF) is usually reserved for the stack. The onboard partial decoding scheme ensures that page zero and page one addresses will be available in the lower 512 addresses of the 2114 RAMs. Because of the partial decoding, it would be possible to address the lower 512 bytes with page one and page zero addresses while addressing the upper 512 bytes with high page addresses such as \$6200–\$63FF.

Circuit of the interface card

The full circuit of the card is given in Figure 2. The onboard 6502 is driven by a 1 MHz crystal clock circuit. A reset push-button switch is provided to enable the 6502 to jump to the reset vector. The 6502 will also reset when power is applied to it. The 2114 RAMs are directly connected to the 6502 bus, with their negative chip selects connected to A15 from the 6502.

The +5 V and earth lines are connected to an external power supply when the card is used in the standalone mode. The switch S1-1 disconnects the 6502 and 2114s from the +5 V line when the card is plugged into the Apple II, hence disabling them. Strictly speaking, the 6502 and 2114s should be removed from the card when plugged into the Apple II, to eliminate interference to the Apple II bus from these integrated circuits. In practice, opening S1-1 appears to be sufficient to prevent interference with the Apple II bus.

The 6522 and 2716 are connected both to the Apple II bus and to the bus of the onboard 6502. Bus contention does not appear to arise if, as stated above, the onboard 6502 and 2114s are disabled while the card is in the Apple II. The 6522 and 2716 are connected to the onboard A15 and A11 lines in accordance with the onboard address decoding scheme above. These two address lines are connected to the chip selects CS1 and $\overline{\text{CS2}}$ respectively of the

Table 1. Addressing scheme for onboard 6502

Integrated circuit	Address lines															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2716	1	d	d	d	1	x	x	x	x	x	x	x	x	x	x	x
6522	1	d	d	d	0	d	d	d	d	d	d	d	x	x	x	x
2114s	0	d	d	d	d	d	x	x	x	x	x	x	x	x	x	x

d, don't care
x, internal address line

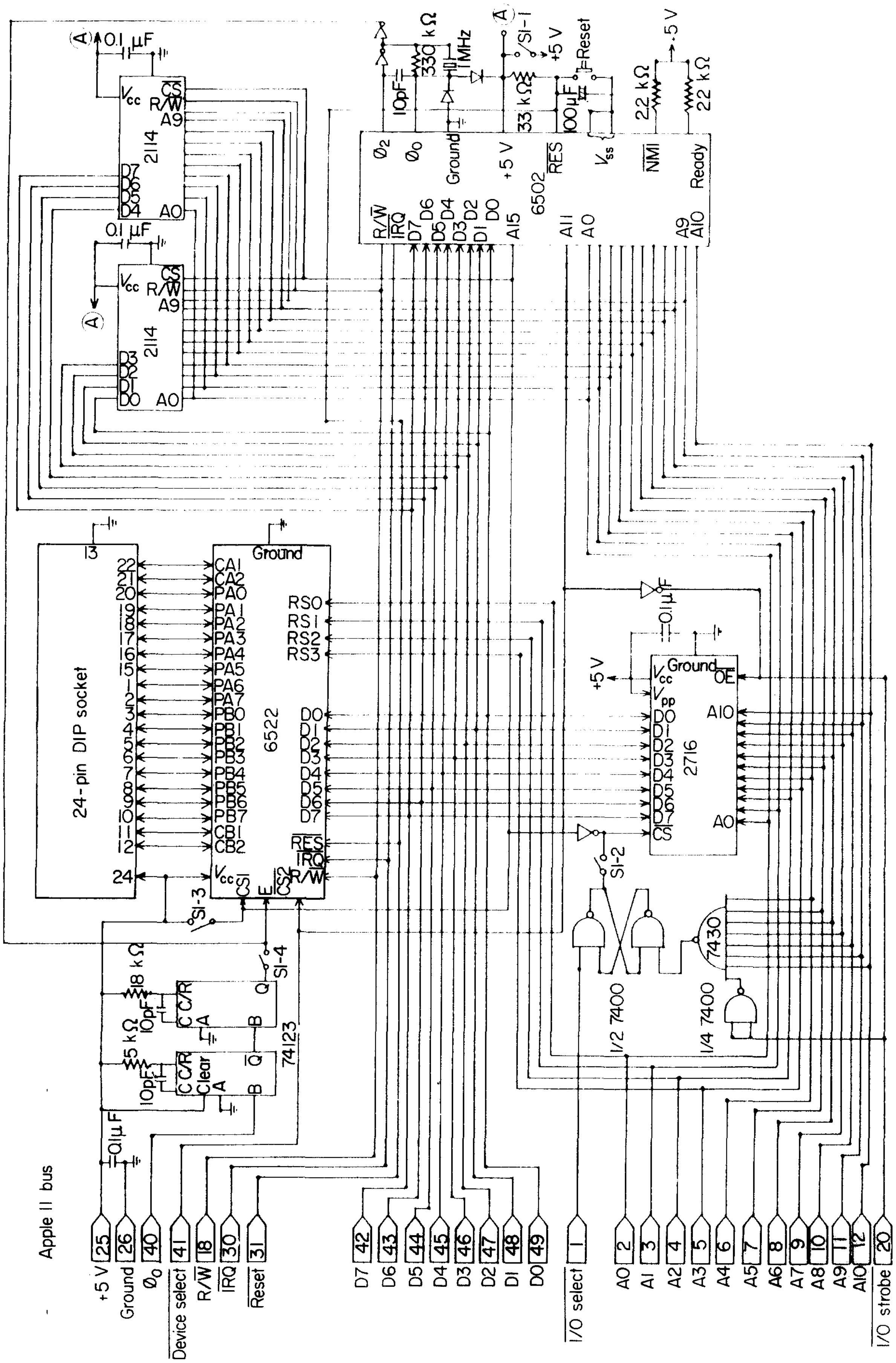


Figure 2. Circuit diagram of the interface card

6522. The same two address lines are connected to the chip select CS and output enable OE respectively of the 2716, via inverters.

While the card is in the Apple II, the 6522 is selected by the Device select line, and the 2716 is selected by the I/O select line (via a latch as described above) and the I/O strobe line. The 2716 is deselected by address lines A4 to A15 and the I/O strobe line. This means that if the last 16 addresses of the 2716 (from $CFF0 to $CFFF) are accessed, the EPROM is deselected. This also means that these 16 addresses cannot be used for programs or data.

The switch S1-2 disconnects the CS chip select of the 2716 from the latch output when the card is out of the slot and the onboard 6502 is active. Likewise, S2-3 connects the CS1 chip select of the 6522 to the +5 V line when the card is in the slot, since in that case only the CS2 chip select needs to be enabled by the Device select line of the Apple II.

The 6522 is driven by the onboard crystal clock when the card is out of the slot, and by the Apple II phase zero clock (pin 40 on the Apple II bus) when the card is in the slot. This clock is synchronized with the 6522 E clock input through two one-shot multivibrators (74123). Switch S2-4 disconnects the 6522 from the Apple II clock when the card is out of the slot.

The switch positions for proper operation should be as summarized in Table 2.

The output of the 6522 is taken to a 24-pin DIP socket. The A and B ports (with their control lines) are both available from the socket, as are connections to +5 V and earth.

Once the application program has been burnt into the EPROM, the onboard 6502 should be able to jump to the start of the program when the card is operating independently. This is ensured by storing the starting address of the program in the EPROM locations $CFFC and $CFFD (Apple II addresses). When the card is out of the slot the onboard partial decoding will ensure that, when the reset button is pressed, the onboard 6502 which is looking for the reset vector in locations $FFFC and $FFFD will find it within the 2716, since the 2716 addresses are duplicated as shown in the onboard decoding scheme above.

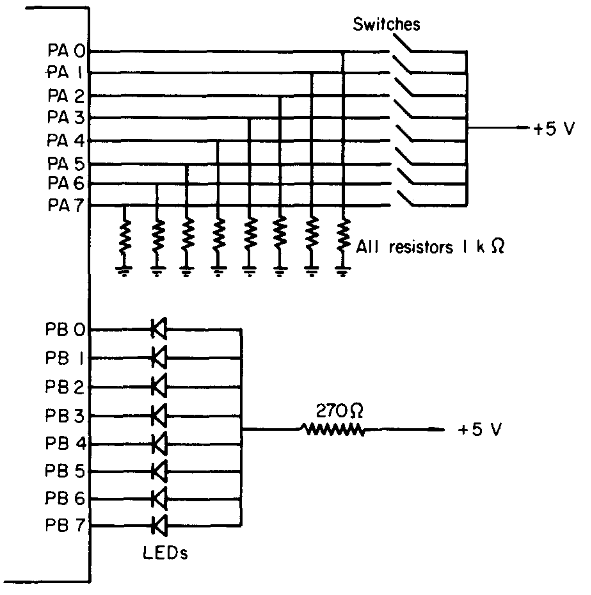


Figure 3. Test circuit

Table 2. Switch positions for proper operation

Card position	Switch			
	S1-1	S1-2	S1-3	S1-4
In slot	Off	On	On	On
Out of slot	On	Off	Off	Off

TESTING THE CARD

The card was constructed and tested with a test program. A simple test circuit was set up as shown in Figure 3. Eight binary switch inputs were connected to port A and eight LEDs to port B.

A test program was written and burnt into the EPROM (Appendix 1). This program configures port A as all inputs and port B as all outputs. The eight LEDs should display the inverse of the states of the corresponding eight switches.. In the process, the 1024 locations of the RAM are tested. The lowest 256 bytes are addressed as page zero (from $0000 to $00FF). The next 256 bytes are addressed as page one (from $0100 to $01FF) and are reserved for use by the stack. The upper 512 bytes are addressed as from $6200 to $63FF. Each pass of the program will cycle through all the locations of the RAM, page one being used as stack by a push-and-pull operation.

The program was burnt into the 2716, with the starting address ($C800) burnt into addresses $FFFC and $FFFD. The program was tested while the card was in the Apple II slot, using the 6502 and RAM of the Apple II. Next the card was taken out, and the switches S1-1 to S1-4 were set for standalone operation. The program was then run using the onboard 6502 and RAM. The program worked well in both cases. This was verified by observing the LEDs for various settings of the port A switches.

CONCLUSION

An I/O interface card for the Apple II was designed, constructed and tested. The card provides the Apple II with a 6522 VIA and a 2716 EPROM. The novel feature of the card is its onboard 6502 and 1 kbyte of RAM. This enables the card to be used as a standalone microcomputer, after the VIA interfacing and EPROM program have been tested and debugged while the card is in the Apple II. The card is thus a simple incircuit emulator when it is plugged into the Apple II slot.

Although other incircuit emulators for the Apple II have been described[3] and are available on the market, the card described here is perhaps unique in combining incircuit emulator and target system in one card. The card may be viewed in two ways: either as an I/O interface which can become a standalone system, or as a standalone system which is also an incircuit emulator with itself as the target system. Either way, we believe that the interface card may prove useful as a controller or wherever a simple standalone microcomputer is required. The same design concept may also be applied to other personal computers with an I/O bus.

REFERENCES

1 *Apple II reference manual*

2 Hofacker, W and Ekkehard, F *The custom Apple and other mysteries* IJG Inc., Upland, CA, USA (1982)

3 Ferguson, J D 'In-circuit emulation for the Apple II computer' *Byte* Vol 8 No 7 (September 1983) pp 419–444

APPENDIX 1: TEST PROGRAM

```
SOURCE FILE: LEDTEST
C0B3:          1 DDRA    EQU   $C0B3      ;DATA DIRECTION REGISTER A
C0B2:          2 DDRB    EQU   $C0B2      ;DATA DIRECTION REGISTER B
C0B1:          3 IRA     EQU   $C0B1      ;INPUT REGISTER A
C0B0:          4 ORB     EQU   $C0B0      ;OUTPUT REGISTER B
------NEXT OBJECT FILE NAME IS LEDTEST.OBJ0
C800:          5         ORG   $C800
C800:A9 00     6         LDA   #00
C802:8D B3 C0  7         STA   DDRA       ;SET PORT A AS INPUT
C805:A9 FF     8         LDA   #$FF
C807:8D B2 C0  9         STA   DDRB       ;SET PORT B AS OUTPUT
C80A:A2 00     10        LDX   #0         ;DEFINE STACK
C80C:AD B1 C0  11 START  LDA   IRA        ;READ PORT A SWITCHES
C80F:95 00     12 LOOP   STA   0,X        ;TEST PAGE 0 OF RAM
C811:B5 00     13        LDA   0,X
C813:9A        14        TXS
C814:48        15        PHA              ;TEST STACK (PAGE 1 OF RAM)
C815:68        16        PLA
C816:9D 00 62  17        STA   $6200,X    ;TEST PAGE 3 OF RAM
C819:BD 00 62  18        LDA   $6200,X
C81C:9D 00 63  19        STA   $6300,X    ;TEST PAGE 4 OF RAM
C81F:BD 00 63  20        LDA   $6300,X
C822:E8        21        INX              ;NEXT LOCATION
C823:D0 EA     22        BNE   LOOP       ;NO, GO TO NEXT LOCATION
C825:8D B0 C0  23        STA   ORB        ;YES, WRITE TO PORT B LEDS
C828:4C 0C C8  24        JMP   START
C82B:60        25        RTS

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```