Improving the OSI Challenger C2

First in a two-part series on the inner workings of Ohio Scientific's "500" boards.

Ugo V. Ré
167 Sprucewood Drive
Levittown, NY 11756

OSI's C2 is a general-purpose system incorporating several unique features. In addition to the advertised features, there are other hardware features and prototype areas on many of the circuit boards that you can use to implement some specialized circuits.

While the documentation is sparse for the C2-4P, a thorough study of the 500 manual and the schematics for the various boards will reveal several features such as: dual system clock operation, a serial interface with multiple baud rate operation and modem control leads, two selectable video screen sizes, reverse video display and a parallel interface option.

This article describes the hardware features of the various boards and the modifications I made to these boards to implement the additional features. I used an older C2-4P system, which contained two power supplies (+5 V, 3.5 Amps and −9 V, 1.5 Amps), a four slot bus backplane, a model 500 CPU board, a model 540 video board and a model 542 polled keyboard in a typewriter-style case. In the newer systems the 500 CPU board has been replaced by a 502 CPU board, and there is only one power supply (+5 V, 4.5 Amps).

Model 500 CPU Board

This CPU board includes the following hardware features:

- A 6502A microprocessor operating at 1 MHz.
- A 6850 ACIA (asynchronous communications interface adapter)-based serial interface configured for both RS-232C and 20 mA loop current.
- 4K static RAM for user programs.
- Microsoft BASIC in 8K ROM.
- System monitor and I/O controllers in three 1702 EPROMs.
- Provisions for a user-provided 6820 PIA (peripheral interface adapter) parallel I/O port.

System Clock

The 6502 clock (see Fig. 1) is provided by a dual one-shot operating as a multivibrator. The clock circuit is populated as an adjustable two-speed clock that is normally set for a high speed of 1 MHz, but with the optional WAIT diodes, it will revert to a low-speed operation of approximately 500 kHz whenever the WAIT line is brought low.

With the components supplied by OSI, you can adjust the clock via R50 (see Fig. 2) for a high speed of 1.6 MHz without creating any problems. Since the frequency will have a tendency to change with temperature, all frequency adjustments should be made only after a long warm-up and with the case closed.

If the clock speed has been set too high or has drifted high, then the following problems, which are usually caused by the slow access time of the EPROMs and the ROMs, may occur:

- Screen does not display C/W/M? after reset.
- Monitor or I/O not operating correctly.
- Keyboard operation not recognized by computer in machine or BASIC mode.
- Programs stop running or will not run.

To correct the problem, lower the clock speed or install any of the WAIT diodes (D5, D6, D8, D10 or D11). See Fig. 3.

I have been operating the system at 1.58 MHz without any WAIT diodes and have not had a problem. However, I have noticed that programs execute faster and the cassette operation is flawless even when operated at 1200 baud.

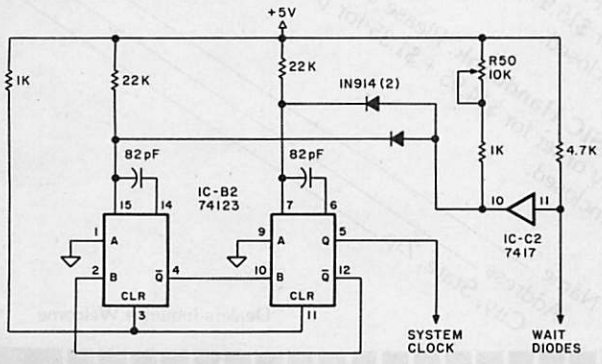


Fig. 1. CPU system clock circuit.

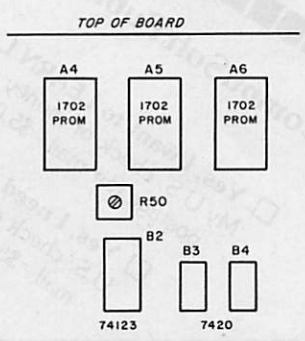


Fig. 2. System clock component location.

ACIA Clock

The ACIA-based serial interface (Fig. 4) uses a 555

astable multivibrator to provide the baud rate clock. The clock circuit has provisions for five capacitors, which are jumper selected to provide a wide range of baud rates. Potentiometer R51 allows the frequency to be fine-tuned to 16 × the baud rate. However, this hard-wired method limits the interface to operation at only one baud rate.

There are two methods you can use to switch the frequency of the 555 multivibrator: change the C5 capacitor or change resistors R22 and/or R23. Of the two methods, I chose the former. Changing resistor values involves changing multiple combinations of R22 and R23 or using large values for R22. Additionally, the duty cycle of the square wave, which is directly affected by the ratio of the resistor values, becomes a spike whenever the R22–R23 ratio is too large.

I made the following modifications to provide switch-selectable baud rates:

- Calculate the capacitors required for the desired baud rates. Table 1 gives the formula to calculate the capacitor values. Table 1, column A, gives six capacitors for six possible baud rates.
- Install the six capacitors on a two-pole, six-position rotary switch (Radio Shack #275-1386). The capacitors in column A are not available as standard values. Therefore, I used standard values wired in series to obtain the required values (see Table 1, column B).
- Connect the first capacitor, .068 uF, between the first switch position and one terminal on the other half of the switch (see Fig. 5). The five remaining capacitors are wired in series from terminal to terminal starting at the first terminal.
- Remove the jumper between the J5 donut and the capacitor on the board (see Fig. 6).
- Install a shielded wire from the J5 donut to the rotary switch pole. The shield should be grounded at the board end with the other end soldered to the ground terminal on the rotary switch (see Fig. 5). I used a shielded wire to prevent stray signals from affecting the 555 circuit and frequency.
- After the wiring has been completed and checked, turn on the computer and allow it to warm up.
- To check and adjust the frequency, connect a frequency counter to L1 connector pin 7 (see Fig. 6), then reset the computer.
- Select the first switch position, 110 baud, and adjust the potentiometer, R51 (see Figs. 5 and 6), for a frequency of 1760 Hz.
- Select the other baud rates and check that they are within 0.1 percent of the required frequency (see Table 1, column C). Do not readjust R51.

Frequency Adjustment

If the frequency is higher than the required value, a low-value trimming capacitor can be added in parallel with the capacitor being tested to lower the frequency. If the frequency is lower than required, replace the capacitor with a capacitor of lower value and add trimming capacitors in parallel to obtain the required frequency.

With the capacitors wired in series, any change made to one capacitor will affect all others after it in the chain. When trimming capacitors to obtain the required frequency, always work on the lower frequency before adjusting the next higher frequency.

Using the above capacitor trimming procedure, it is possible to get the frequencies within ± .05 percent (see Table 1, column D). Generally, I use only three baud rates — 110 for a Teletype and 300 and 1200 for the audio cassette and modem.

555 Frequency calculations

$$C(uF) = \frac{1}{.693^{*} R^{*} fHz}$$

$$R = (R_{22} + 2(R_{23} + R_{51}))$$

(pot R_{51} set at midpoint)

$$R = 12060$$

555 Frequency components

Rate	Baud calculated value (A)	Capacitor value in series (B)	frequency calculated value (C)	frequency measured value (D)
110	.068	.068	1760	1759
300	.025	.039	4800	4801
1200	.0062	.0082	19200	19194
2400	.0031	.006	38400	38379
4800	.0015	.003	76800	76869
9600	.0008	.002	153600	153785

Table 1.

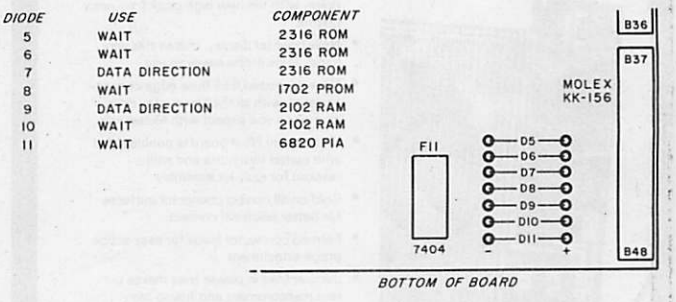


Fig. 3. WAIT diode location.

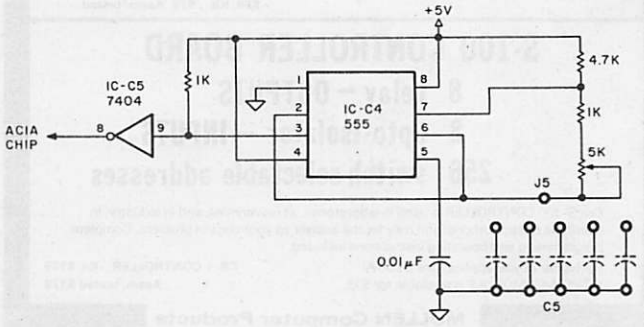


Fig. 4. ACIA clock circuit.

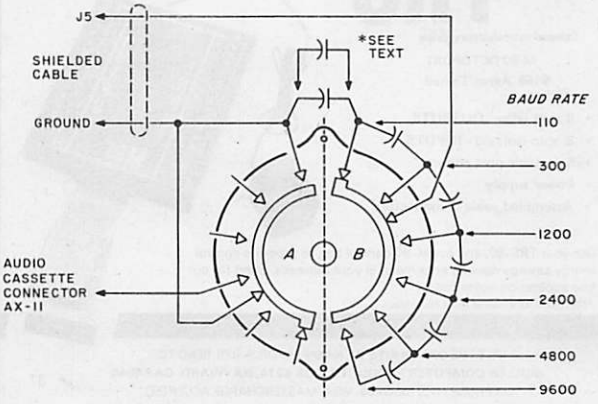


Fig. 5. Baud rate selector switch.

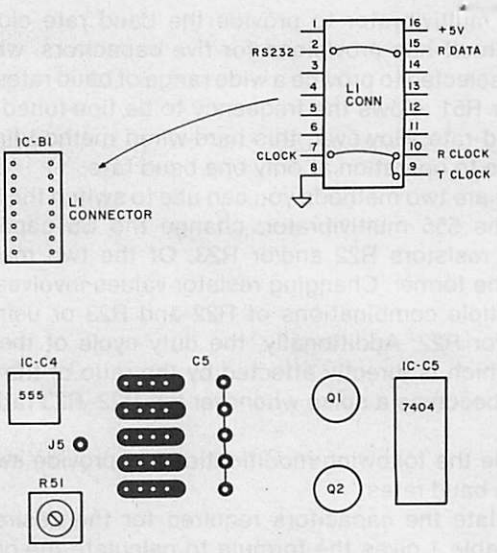


Fig. 6. ACIA clock component location.

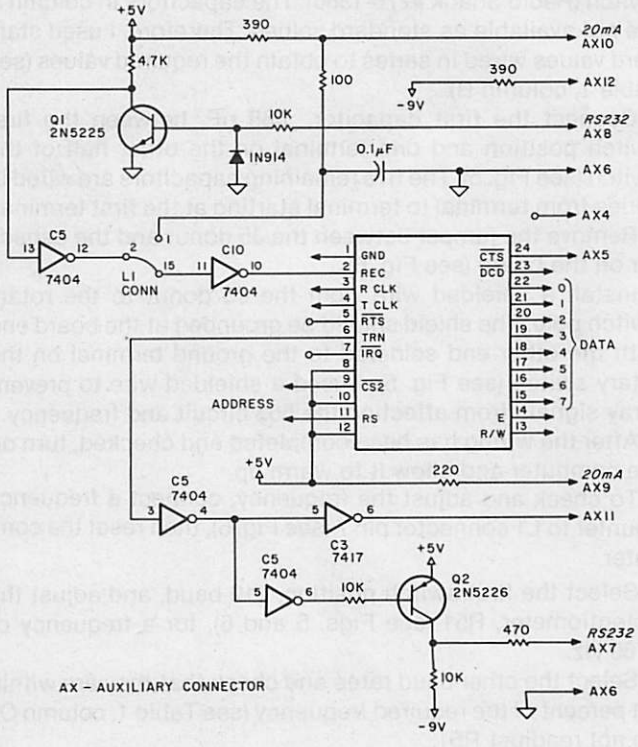


Fig. 7. Serial interface circuit.

For 300/1200 baud operation only, install a .0068 uF capacitor on the board and adjust R51 for a frequency of 19,200 Hz. Under software control, set the bits in the 6850 control register (see serial interface section) to select the ÷16 (1200 baud) or ÷64 (300 baud) clock rate.

Serial Interface

The 500 board (see Fig. 7) is provided with a 6850 ACIA chip and the components for both a 20 mA current loop and an RS-232C interface output circuit, simultaneously connected, and a 20 mA or RS-232C input circuit. Only one input circuit can be connected at a time.

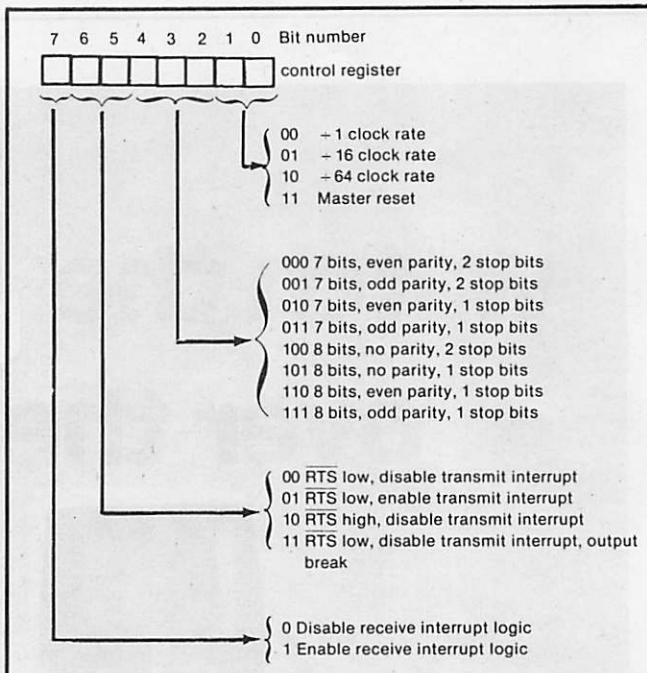
The 6850 is a 24-pin DIP that interfaces the CPU to outside devices by converting parallel I/O to serial I/O. Either a byte of parallel data or a control code is transmitted by the CPU to

the 6850, while data or status is received by the CPU. In addition to the data bus and the transmit and receive clock and data leads, the 6850 has other control leads: the modem control leads $\overline{CTS}$, $\overline{RTS}$ and $\overline{DCD}$; the RS (register select) lead, which determines which of two addressable locations will be accessed; and the R/W lead, which determines whether a read or write operation is in progress. (The two addressable locations are the control/status address (FC00 hex) and the data address (FC01 hex)).

The control register is used to control the operation of the 6850. A code, which establishes the parameters for the 6850's operation (see Table 2), is written into the control register by the CPU. When the computer is reset, the CPU first loads the register with 03 hex (reset) and then B1 hex (÷16 clock rate, eight bits, no parity, two stop bits). If this operation does not meet your needs, reset the control register and load in a new code (see Table 2).

The status register uses status flags to monitor the serial data transfer logic (see Table 3). The status register is read into the CPU's accumulator, then bit 0 or 1 is shifted right into the carry register. The carry register is then checked for the status of the receive data register or the transmit data register. If the register is set, the CPU will proceed to read or write data to the 6850 (see Table 3).

A program to read or write data to the 6850 is listed in Table 4. The other bits in the status register are used by the 6850 to determine the status of external devices, bits 2 and 3; to detect receiving errors, bits 4, 5 and 6; and to determine the source of an unacknowledged interrupt request, bit 7. The computer does not check the status of bits 2 through 7, and, in most cases, we do not need to check them for status during normal operation.



Program to change the control register.

A9; 03	LDA	Master reset code
8D; 00; FC	STA	Write code to register
A9; XX*	LDA	New parameters
8D; 00; FC	STA	Write new parameters to control register

*Note: Select new parameters from list above. Byte is then converted to hexadecimal notation and entered in program.

Example: ÷16 clock rate; 7 bits, even parity, 2 stop bits; $\overline{RTS}$ low, enable transmit interrupt; Enable receive interrupt logic 10100001 binary = A1 hex.

Table 2.

The serial I/O ports of the 6850 are wired to the components that comprise the 20 mA and RS-232C circuits. (The 20 mA current loops interface is specifically for use with the ASR-33 Teletype. It has no common ground and cannot be used with terminals requiring a common ground on output.) The 500 board has a serial interface auxiliary connector mounted near the 6850 chip (see Fig. 8), which uses pins 5 through 12 for the RS-232 and 20 mA data leads and a $\overline{CTS}$ (clear to send) control lead.

The 6850 uses a low $\overline{CTS}$ to set the status register and report a TDRE (transmit data register empty) condition. The computer determines when to transmit a byte of data to the 6850 by testing the register for the TDRE condition. A high

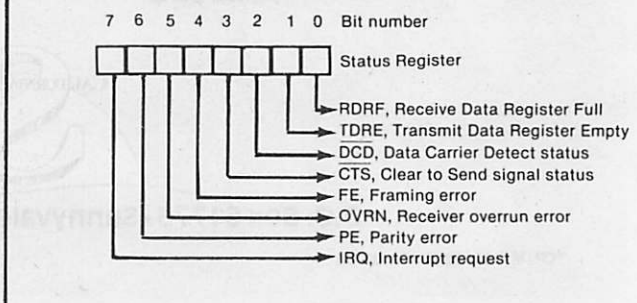


Table 3.

AD, 00, FC	LDA	load status register
4A	LSR	shift bit '0' to carry
90, FA	BCC	not ready, check again
AD, 01, FC	LDA	load receive data
9D, XX, XX	STA	store data in memory

Table 4a. Serial data operation. Receive data from the serial port. This program will continue until all data is received. It should check for end of data character, and the index register must be incremented to access the next memory address.

AD, 00, FC	LDA	load status register
4A	LSR	
4A	LSR	shift bit 'I' to carry
90, F9	BCC	not ready, check again
BD, XX, XX	LDA	load Acc from memory
8D, 01, FC	STA	store data in ACIA

Table 4b. Transmit data from memory to serial port. This program will continue until all data has been transmitted. It must increment the index register to access the next memory address. It must also check for the last data address.

$\overline{CTS}$ signal will prevent the register from reporting a TDRE condition.

The $\overline{CTS}$ is normally strapped to ground; however, the lead can be used as an interrupt signal for loss of transmission facilities, low paper alarm, printer or cassette not on line, etc., to prevent the needless transmission of data into an open line.

To modify the $\overline{CTS}$ lead, remove the strap between pin 24 and ground and install a 1k resistor from pin 5, auxiliary connector to +5 V. The $\overline{CTS}$ lead, which will be high, can now be used to determine the status of the receiving device. Do not make this modification unless you can control the $\overline{CTS}$ lead, since the computer will go in a loop on SAVE, where it will remain until $\overline{CTS}$ is brought low.

The 6850 also has an $\overline{RTS}$ (request to send) control lead used to inform a data set that it is ready to transmit data. The data set will return the $\overline{RTS}$ signal as a $\overline{CTS}$ signal when it establishes a data link. The 6850 outputs a low $\overline{RTS}$; therefore, $\overline{CTS}$ will be low and the computer will proceed to output the data. The $\overline{RTS}$ terminal, pin 5 of the 6850, is not used; however, it can be wired to the spare pin 4 of the auxiliary connector.

The only remaining work is to install an EIA connector (DB255) in the cutout at the rear of the case and wire it to a Molex plug (KK-156 cut to provide one 3-pin and one 9-pin plug) following the pin-out in Table 5. Now you can use the EIA connector to connect to a printer or data set. The 20 mA leads are wired to the connector only for convenience and

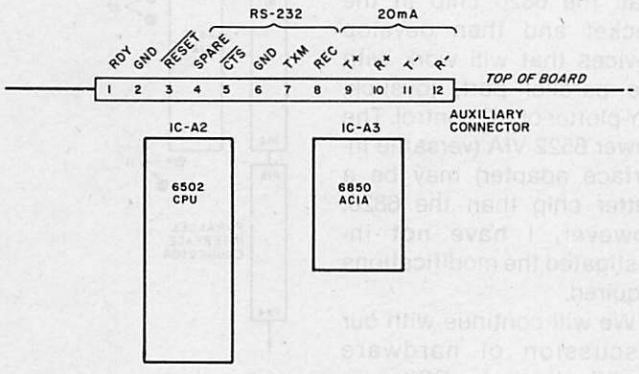


Fig. 8. Serial interface connector.

✓ Reader Service index—page 241

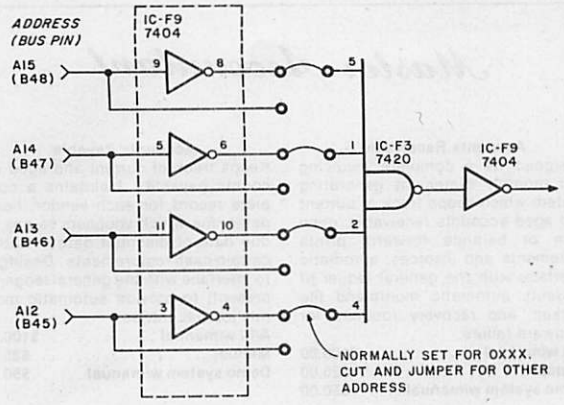


Fig. 9. 2102 RAM address decoding.

EIA Pin-out		Auxiliary Connector
1	Ground	--
2	Input to CPU	8
3	Output to CPU	7
4	Request to send	4
5	Clear to send	5
7	Signal ground	6
12	20 mA input (+)	10
13	20 mA output (+)	9
24	20 mA input (−)	12
25	20 mA output (−)	11

Note: Auxiliary connector pins 1, 2 and 3 are wired to the 6502 CPU control pins, RDY and $\overline{RES}$.

Table 5. EIA connections.

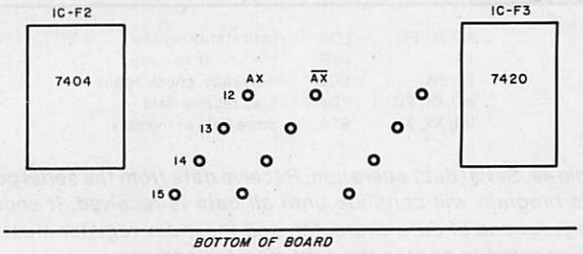


Fig. 10. Address jumpers for 2102 RAM.

are not part of the RS-232C standard.

RAM Addressing

The 2102 RAM on the 500 board is decoded for a 4K block starting at hex 0000 (page zero) (see Fig. 9). Additional memory boards, model 420 or model 527, can be added to a spare slot on the motherboard. The addition of a 420 board will not cause a problem, since the board can be decoded to occupy the next unused 4K block of memory. The addition of the 527 board, however, will create a problem, since it can only be changed to occupy an 8K block. Therefore, when installing a 527 board, you should decode it for page zero and decode the 2102 RAM for some other location. The foil trace from the $\overline{Ax}$ donut to the right-most donut (see Fig. 10) must be cut and a new jumper installed to the Ax donut.

I installed a 527 board with 16K of memory and moved the 2102 memory to location C000 hex. This memory is now used for machine-language programs. BASIC, which normally uses all consecutive memory, cannot normally access this memory; therefore, there is little danger of destroying any programs stored at that location.

ROMs and EPROMs

No changes were made to these areas, but other ROMs, PROMs or EPROMs containing a more versatile language or monitor can be substituted for the ROMs and EPROMs presently on the board. The 500 CPU manual lists other ROMs that can be used with only minor strapping changes.

Peripheral Interface

The 6820 PIA interfaces with the CPU to provide two parallel I/O ports. Although I have not implemented the parallel port fully, I have added the following components to the board (see Fig. 11):

- A 40-pin IC socket at location A1.
- J1 and J2 jumper located at the bottom and right side of socket.
- Two Molex connectors along left edge of board.

All that remains is to install the 6820 chip in the socket and then develop devices that will work with the parallel port, joystick, X-Y plotter or AC control. The newer 6522 VIA (versatile interface adapter) may be a better chip than the 6820; however, I have not investigated the modifications required.

We will continue with our discussion of hardware modifications to OSI's circuit boards next time. ■

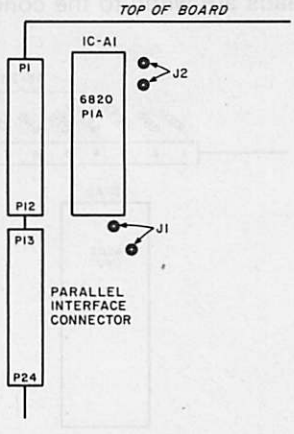


Fig. 11. PIA component location.