MODEL 2719
2 PARALLEL / 2 SERIAL I/O INTERFACE

Reference Manual

California Computer Systems

42000095-01

CCS MODEL 2719

2 PARALLEL / 2 SERIAL I/O INTERFACE

Reference Manual

Rev B
Manual #42000095-01

California Computer Systems
250 Caribbean Drive
Sunnyvale CA 94086

The information contained in this manual is believed to be correct at the time of publication. However, CCS assumes no liability resulting from the use of this publication.

Publication History:

Revision A printed December 1981
Revision B printed February 1982

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 CONFIGURING THE 2719

CHAPTER 3 PROGRAMMING INFORMATION

CHAPTER 4 HARDWARE DESIGN

APPENDIX A TECHNICAL INFORMATION

APPENDIX B SAMPLE DRIVERS

CHAPTER 1

INTRODUCTION

1.1 GENERAL DESCRIPTION

The CCS Model 2719 is capable of interfacing a wide variety of peripheral equipment to Z-8Ø-based CPUs. Software and hardware options give a high degree of flexibility to the 2719's two serial and two parallel I/O ports. The serial ports are controlled by a Z-8Ø DART (Dual Asynchronous Receiver/Transmitter), which handles asynchronous serial data transfers in all common formats. Baud rates up to 115.2K are available. The parallel ports, each controlled by a 6821 PIA (Peripheral Interface Adapter), are designed to implement Centronics interfaces, but each may also be used as two unidirectional 8-bit ports with 2-line handshaking or as one unidirectional 16-bit port.

The 2719 fully supports the three Z-8Ø interrupt modes for all ports. Separate headers allow the user to select the Mode Ø or Mode 2 interrupt priority level and the devices which will directly control the INT* line for Mode 1 and Mode 2 interrupts. (In Mode Ø the INT* line is controlled by the system's Interrupt Controller.) The 2719 also supports CCS's fast Mode 2 Interrupt Daisy Chain Look-Ahead Scheme as implemented in CCS Systems 3ØØ/4ØØ.

Though designed especially for use with CCS Systems 3ØØ/4ØØ, the 2719 is compatible with CCS System 221Ø as well as with a majority of the Z-8Ø-based S-1ØØ systems presently available. The base address of the I/O ports is jumper-selectable. Clock phase, Interrupt-Acknowledge wait, and reset options allow the user to meet the special conditions of specific systems.

1.2 USING THIS MANUAL

This manual is intended to provide information required by system integraters, troubleshooters, and programmers. Chapter 2 deals with board configuration, including hardware-configured serial and parallel interface options. Chapter 3 discusses the 2719-unique programming requirements of the DART and CTC and provides complete programming instructions for the PIAs. Chapter 4 presents a detailed discussion of the hardware design of the 2719, and is intended to be read in conjunction with frequent references to the Schematic/Logic Diagram included, along with various technical illustrations and tables, in Appendix A. Sample drivers for the serial and parallel port drivers for CP/M and OASIS operating systems are provided in Appendix B.

1.3 SPECIFICATIONS

I/O INTERFACES

SERIAL:

Two Asynchronous Ports Meet EIA RS-232-C Standard, Full or Partial Primary Channel
Synchronous Capability (SIO/Ø Plug-Compatible)
Easy DCE-to-DTE Reconfiguration
Non-Standard Handshaking Options
Programmable (Z-8Ø CTC) or External Baud Rates

PARALLEL:

Two Centronics-Type Ports
Hardware and Software Reconfiguration Options
Port Buffers Disabled When Cable Disconnected

SYSTEM INTERFACE

S-1ØØ: Complies with IEEE Task 696.1/D2
Supports All Three Z-8Ø Interrupt Modes
Supports CCS's Mode 2 Interrupt Extended Daisy Chain Look-Ahead Scheme
Jumper-Selectable Board Base Address
Full Buffering of Bus-Driving Outputs, Schmitt-Trigger Bus Receiver Inputs

POWER

+8 Volts Regulated On-Board to +5 Volts
+16 Volts Regulated On-Board to +12 Volts
-16 Volts Regulated On-Board to -12 Volts

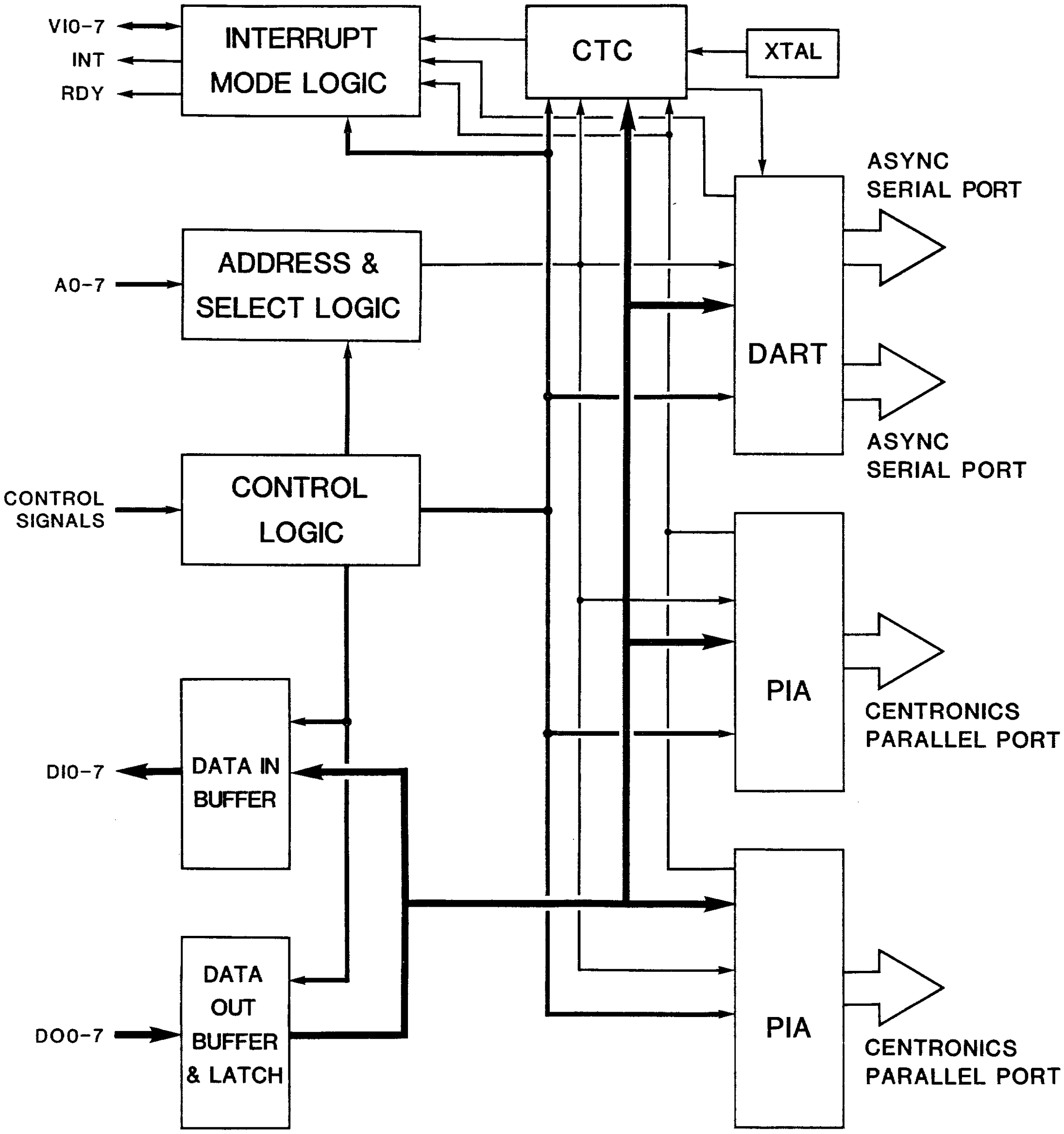
Consumption:	.75 Amps at +8 Volts
	.Ø5 Amps at +16 Volts
	.Ø5 Amps at -16 Volts

Heat Burden:	11Ø gram-calories/minute
	.45 BTU/minute

ENVIRONMENTAL REQUIREMENTS

Temperature:	Ø to 7Ø C. (32 to 155 F.)
Humidity:	Up to 9Ø% Non-Condensing

1.4 2719 BLOCK DIAGRAM



CHAPTER 2

CONFIGURING THE 2719

The 2719, while designed to be flexible, has also been designed to require as little configuration as possible in its primary environment, CCS Systems 300/400. When the 2719 is added to a System 300/400, only the IM2 Header and Protective Ground Jumper require configuration. All other headers and jumpers are shipped configured for a System 300/400. However, if the 2719 is used in a CCS System 2210 or a non-CCS system, additional configuration will be required. This chapter includes configuration instructions for all board options. Jumper and header locations are shown in Figure 2-1. Table 2-2 briefly defines each option.

2.1 BASE ADDRESS JUMPERS

The CTC channels and serial and parallel ports occupy sixteen contiguous port addresses (for relative locations see Table 3-1). The A7-A4 jumpers allow the user to select the base address of the 2719 at any multiple of 16 (10H) between 0 and 255 (00-FFH). Table 2-1 shows the jumper settings for all possible base addresses. The 2719 is configured at the factory for a base address of 50H.

Table 2-1. BASE ADDRESSES

JUMPER SETTINGS				BASE	
A7	A6	A5	A4	HEX.	DEC.
0	0	0	0	00	00
0	0	0	1	10	16
0	0	1	0	20	32
0	0	1	1	30	48
0	1	0	0	40	64
0	1	0	1	50	80
0	1	1	0	60	96
0	1	1	1	70	112
1	0	0	0	80	128
1	0	0	1	90	144
1	0	1	0	A0	160
1	0	1	1	B0	176
1	1	0	0	C0	192
1	1	0	1	D0	208
1	1	1	0	E0	224
1	1	1	1	F0	240

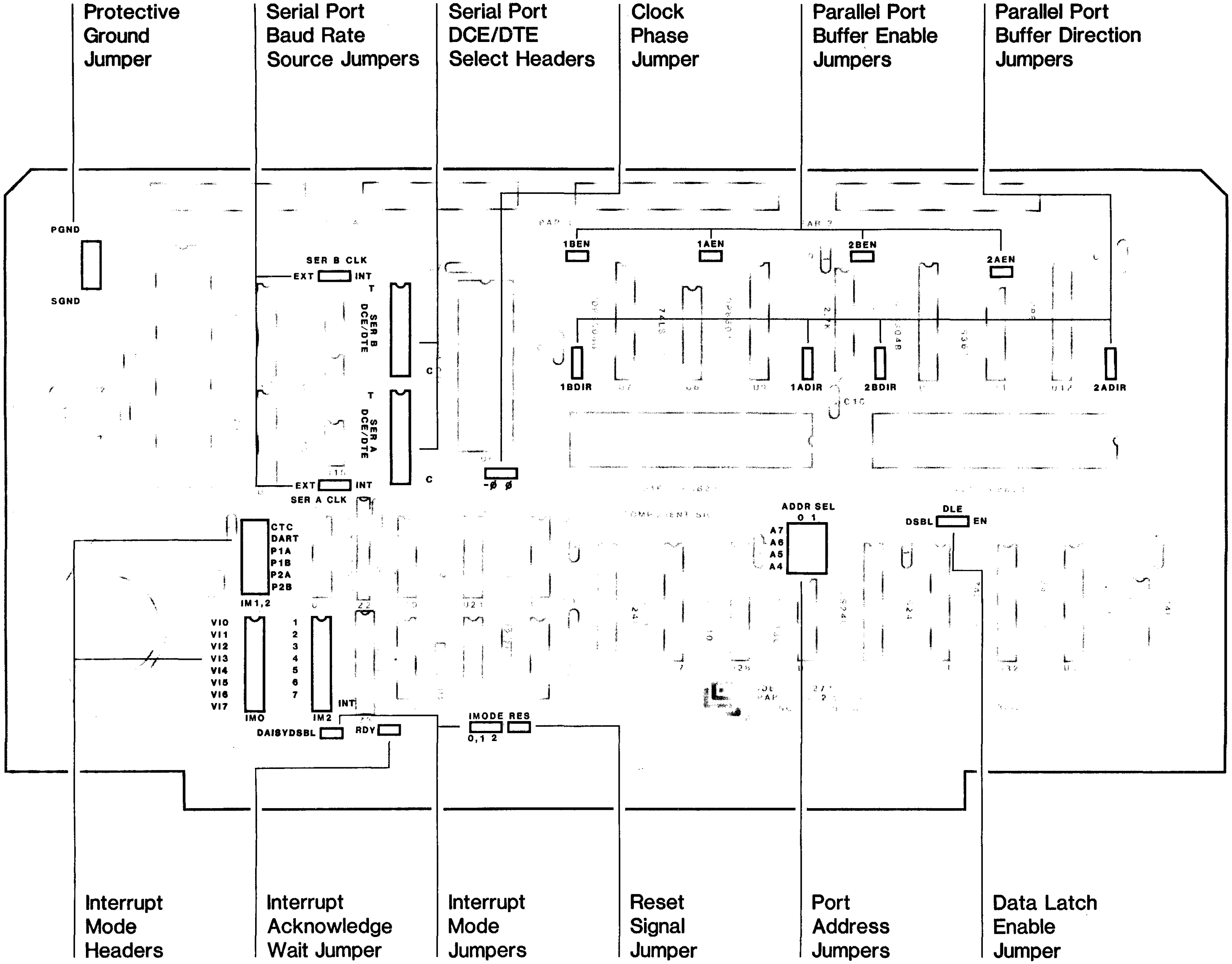


Figure 2-1. JUMPER AND HEADER LOCATIONS

Table 2-2. USER OPTIONS

OPTION LABEL	FACTORY CONFIGURATION	FUNCTION
ADDR SEL	A7=0, A6=1, A5=0, A4=1	Selects four most significant bits of 2719's base I/O address.
RES	Open	If closed, 2719 is reset by RESET* as well as by POC* and SLV CLR*.
DLE	EN	EN enables latching of all data written to 2719; DSBL disables latching.
-Ø/Ø	Ø	Selects uninverted (Ø) or inverted (-Ø) system clock for CTC and DART clocks.
SER A CLK SER B CLK	INT INT	Selects CTC Ch. 0/1 (INT) or RS-232-C line 15 (EXT) for Serial Port A/B receiver and transmitter clocks.
SER A DCE/DTE SER B DCE/DTE	DTE (standard) DTE (standard)	Serial Port A/B interfaces to DTE device if header pin 1 at T, to DCE device if pin 1 at C. May be modified for non-standard handshaking.
PGND/SGND	Configuration required	Connects Serial Interface Protective Ground to either Signal Ground or Chassis Ground.
1ADIR 1BDIR 2ADIR 2BDIR	I O I O	Conditions Parallel Port 1/2 Channel A/B for input (I) or output (O)
1AEN 1BEN 2AEN 2BEN	Open Open Open Open	If closed, permanently enables Parallel Port 1/2 Channel A/B; if open, Channels A and B enabled by lows on interface pins 30 and 19 respectively.
IM0	Unconfigured	Selects VI* line controlled by 2719 for Z-80 Mode 0 Interrupts.
IM1,2	CTC and DART closed	If circuit closed straight across, corresponding device can pull INT* low.
IM2	Unconfigured	Selects daisy-chain priority level for Z-80 Mode 2 Interrupts.
DAISYDSBL	Closed	If open, disables VI0* input, allowing use of VI0* in Interrupt Mode 0.
IMODE	2	Enables (2) or disables (0,1) output of Interrupt Vector during appropriate Interrupt Acknowledge cycles.

2.2 INTERRUPT MODE AND PRIORITY CONFIGURATION

Three header areas and two jumpers allow you to tailor 2719 interrupts to a particular system. If the system uses an interrupt controller, you will need to configure the 2719 for Mode Ø interrupts. If the system does not implement vectored interrupts, you will need to configure for Mode 1, in which an interrupt causes an automatic restart at location ØØ38H. If the system supports the powerful Mode 2 Interrupt Daisy Chain, as CCS Systems 3ØØ/4ØØ do, you should configure for Mode 2.

The three header areas are labelled IMØ, IM2, and IM1,2 in accordance with the interrupt modes to which they apply. IM1,2 is a 2x6 pad matrix, hardwired for the standard configuration, which may be altered by the installation of jumper wires or header pins and shorting plugs. IMØ and IM2 are socketed 2x8 headers. The Interrupt Mode (IMODE) and Daisy Chain Disable (DAISYDSBL) Jumpers are hardwired for Mode 2 and must be reconfigured for Mode Ø or Mode 1.

2.2.1 Mode Ø Configuration

The first task in configuring for Mode Ø is to reconfigure the IMODE jumper, cutting the trace labelled 2 and installing a jumper in the Ø,1 position. Next, cut the CTC and DART traces of the IM1,2 Jumpers and the DAISYDSBL trace. Finally, install the 16 pin DIP header in the IMØ socket and wire the header cover to select the VI* line by which each device will assert its interrupt. Remember that the lower the number of the VI* line, the higher the interrupt priority. The pins corresponding to the VI* lines are labelled Ø through 7; the six interrupt signal pins are labelled C, D, 1A, 1B, 2A, and 2B. Pins 9 and 16 are not used. More than one interrupt line may be tied to one VI* line; it will then be up to the interrupt service routine called when that VI* line is asserted to determine which device generated the interrupt.

2.2.2 Mode 1 Configuration

Configuring for Mode 1 involves: 1) leaving the IMØ and IM2 Headers unconfigured; 2) reconfiguring the IMODE Jumper by cutting the trace labelled 2 and installing a jumper in the Ø/1 position; 3) cutting the DAISYDSBL trace; and 4) either leaving the IM1,2 Jumpers as they are or installing jumpers, depending on whether or not parallel port interrupts are to be handled through the CTC. If they are (see Chapter 3 for an explanation of how this is done), the IM1,2 Jumpers should be left as they are. If parallel port interrupts are to be asserted directly by the parallel ports, jumpers should be installed in positions P1A, P1B, P2A, and P2B.

2.2.3 Mode 2 Configuration

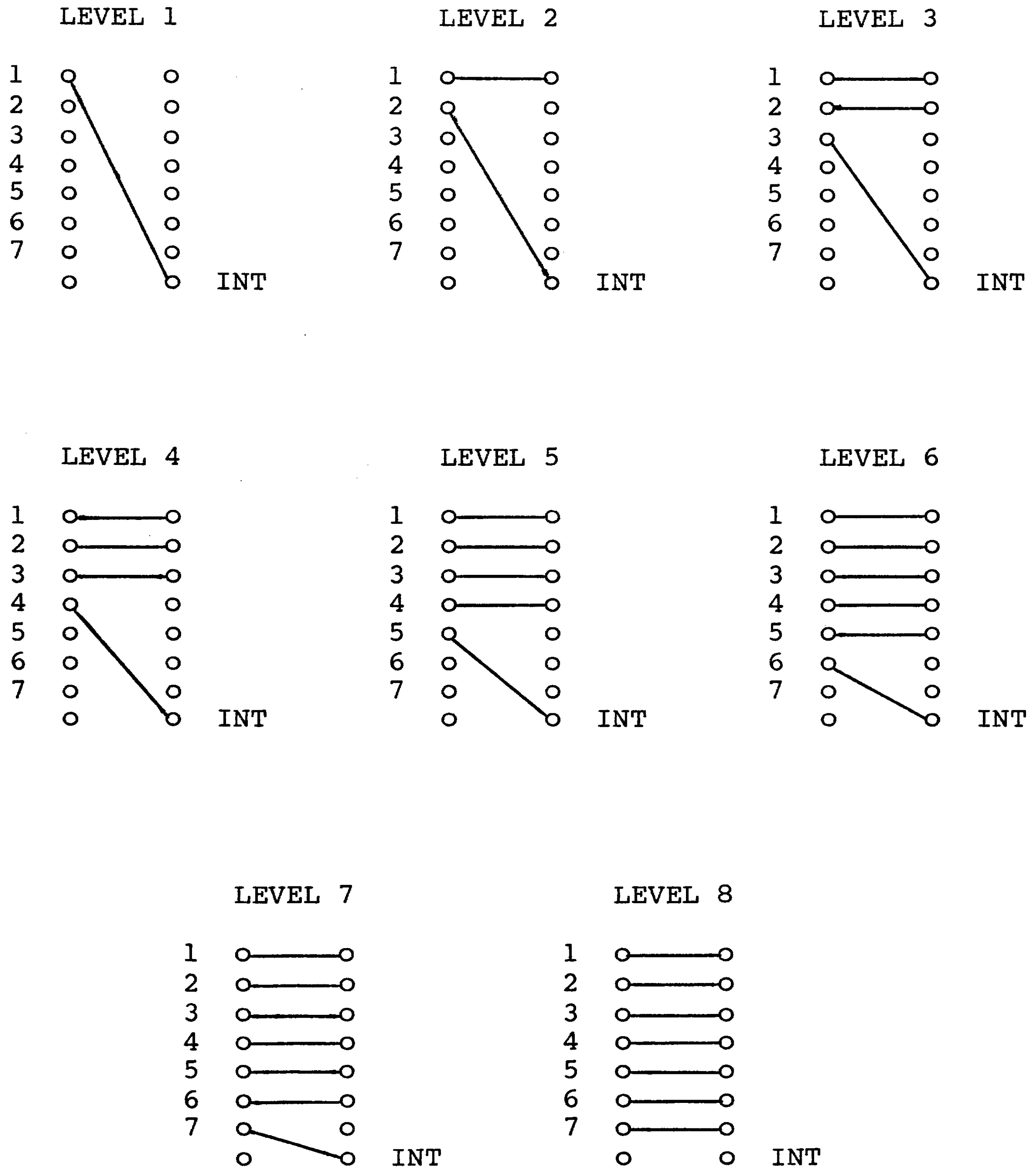
The Z-8Ø Mode 2 Interrupt Daisy Chain, when extended beyond four peripheral devices, requires look-ahead logic to ensure that all devices are properly informed of higher-priority interrupts within the allotted time. CCS Systems 2ØØ, 3ØØ, and 4ØØ implement a unique look-ahead scheme in which: 1) each board participating in the daisy chain asserts its interrupt priority by forcing a given Vectored Interrupt bus line low; and 2) a board is prevented from interrupting when a lower-numbered VI* line is low. Thus there are nine interrupt priority levels, Ø-8; the priority Ø board controls VIØ* and senses no VI* lines, while the priority 8 board senses all VI* lines and controls none. The IM2 Header allows the user to select the interrupt priority level of the 2719 by selecting which VI* line(s) the board will be sensitive to and which VI* line it will pull low.

System 3ØØ/4ØØ interrupt priorities may be determined at the system implementer's discretion, depending on system components and application. However, the priority scheme shown below should be appropriate for the majority of systems. The System Processor is hardwired for Level Ø; it is the only board whose priority is fixed.

	Level Ø:	282Ø System Processor
	Level 1:	28Ø5 Wallclock/Terminator
	Level 2:	283Ø Six-Channel Serial I/O
---->	Level 3:	2719 2 Parallel/2 Serial I/O
	Level 4:	2831 Arithmetic Processor
	Level 5:	2833 GPIB Interface
	Level 7:	2822 Floppy Disk Controller
	Level 8:	2832 Hard Disk Controller

Note that gaps are allowed in the priority structure; thus, it is not necessary to reconfigure a level 3 board to level 2 if there is no level 2 board in the system. However, no priority level may be occupied by more than one board.

Figure 2-2. IM2 HEADER CONFIGURATION



To configure the IM2 Header, determine the priority level, then: 1) tie all lower-numbered left-column pins straight across; and 2) tie the pin corresponding to the 2719's priority level to pin 9, labeled INT. Note that the lowest level to which the 2719 may be assigned is 1; the board is hardwired to sense the VIØ* line, which is reserved as the 282Ø System Processor's priority-assertion line. Figure 2-2 shows configuration of IM2 for all priority levels.

2.3 SERIAL INTERFACES

2.3.1 Synchronous Conversion

To convert the 2719 serial ports for synchronous communications, simply remove the Z-8Ø DART (U3) and replace it with a Z-8Ø SIO/Ø (neither the SIO/1 nor the SIO/2 is plug-compatible with the DART). The SIO/Ø is not supplied with the board.

2.3.2 DCE/DTE Conversion

Each serial port interface includes a 16-pin DIP header with cover for selecting whether the port will interface to a DTE (Data Terminal Equipment) or DCE (Data Communication Equipment) device. Since most peripherals act as DTE, the 2719 is shipped configured to interface to DTE devices. To reconfigure a port to interface to a DCE device, simply turn the port's DCE/DTE header so that instead of having pin 1 in pin 1 of the socket (labeled T), pin 1 is in pin 9 of the socket (labeled C).

2.3.3 Non-Standard Handshaking

Some devices using the RS-232-C interface, especially printers, use non-standard handshaking. To interface such a device, you will need to reconfigure a DCE/DTE Header according to the requirements of the peripheral. Figure 2-3 shows the header pinouts and the standard DTE wiring. Note that, besides the RS-232-C lines normally used by the 2719's serial ports, two other lines, 19 (Secondary Request To Send) and 11 (Unassigned), are also made available for special handshaking. Figure 2-4 shows a header modified for a device such as an NEC Spinwriter printer which handshakes on RS-232-C pin 19 (Sec RTS). Assuming that the model driver in Appendix B is used, the printer's handshake signal must toggle the DCD bit in the DART channel's Status Register; thus RS-232-C pin 19 must be tied to the DART's DCD* input, rather than pin 2Ø (DTR) as in the standard DTE configuration. The procedure for reconfiguring the header for a Spinwriter is as follows: 1) remove the jumper wires connecting pin 1 to pin 16 and pin 8 to pin 1Ø; 2) install a jumper wire connecting pin 1 to pin 1Ø; 3) leave all other header wires as they are.

Figure 2-3. STANDARD DTE HEADER CONFIGURATION

STD CONFIG:

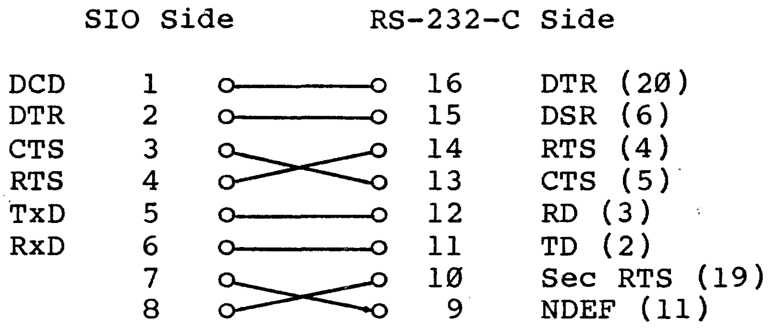
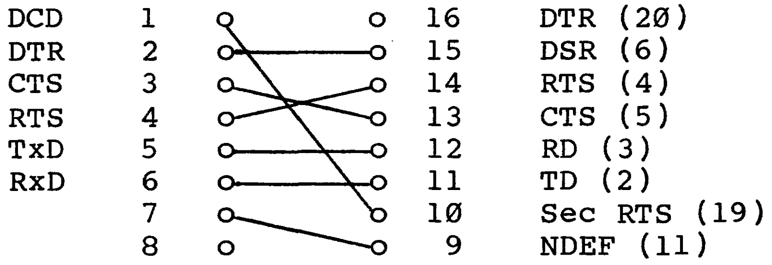


Figure 2-3. SPINWRITER DTE HEADER CONFIGURATION



2.3.4 Baud Rate Source Jumpers

If you plan to use a DART channel to interface a DCE device, you may want that channel's baud rate clock to be supplied by the DCE device. The Channel A and B Baud Rate Source Jumpers, labeled SER A CLK and SER B CLK, allow each DART channel's clocks to be controlled by either the CTC or RS-232-C line 15, TSEC(DCE), from the peripheral. The jumpers are hardwired in the INT position, selecting the internal (CTC) baud rate sources. To select the external (peripheral) baud rate source, cut the INT trace and install a jumper wire in the EXT position of the appropriate Baud Rate Source Jumper.

[Please note that because DART pins RxCA and TxCA are tied together on the PC Board, the Channel A Receiver and transmitter Clocks cannot be separately controlled.]

2.3.5 Protective Ground

Protective Ground is defined as the supply current return path, not a signal current return path. It is intended to equalize the voltage potential of the terminal and the mainframe, and should be implemented whenever the terminal and the mainframe are connected to different power sources which may have different ground potentials. If both serial terminals interfaced through the 2719 are to be connected to the same outlet as the mainframe, the protective ground feature need not be implemented.

At the upper left corner of the board are two jumper pads labeled PGND and SGND. These allow the user to select one of two implementations of the RS-232-C Protective Ground signals for both ports. The recommended implementation is to run a green wire from the PGND pad to the mainframe chassis, with an alligator clip or other convenient method for connection. This conforms to the RS-232 design specifications, ensuring that terminal and mainframe have the same potentials. Use this method with CCS-supplied terminals. Some terminals, however, tie Protective Ground and Signal Ground together or use the Protective Ground as the Signal Ground. If you are using such a terminal with the 2719, you will need to install a 1ØØ ohm, 1/2 Watt resistor (as per EIA standard RS-422-A) between the PGND and SGND pads.

2.4 PARALLEL INTERFACE CONFIGURATION

2.4.1 Buffer Direction Jumpers

Bidirectional buffers are used on the four parallel port data channels. The direction of data flow is determined by the Parallel Port Buffer Direction Jumpers 1ADIR, 1BDIR, 2ADIR, and 2BDIR, which are hardwired for the standard Centronics-type interface configuration. Channel B of each port (Centronics data bus) is hardwired in the O position for output; Channel A of each port (Centronics status bus) is hardwired in the I position for input. To change the direction of a buffer, cut the existing trace and install a jumper wire in the opposite position. [Please note that while each channel may be used for either input or output, the characteristics of the PIA handshaking signals make Channel B more suited for output and Channel A more suited for input.]

2.4.2 Buffer Enable Jumpers

The parallel interfaces are designed so that lows on interface lines 3Ø and 19 enable the Channel A and Channel B interface buffers respectively; otherwise the enable inputs to the buffers are pulled high. Thus, the buffers will not be enabled unless the interface cable is connected. On the Centronics interface lines 3Ø and 19 are defined as ground lines. However, some Centronics-type peripherals may not support the lines as defined. If this is the case, or if the interface is used in a non-Centronics configuration and the peripheral does not assert interface lines 3Ø and 19 low, jumper wires must be installed between the appropriate Buffer Enable Jumper pads to permanently enable the buffers. The four Buffer Enable Jumpers are labeled 1AEN, 1BEN, 2AEN, and 2BEN.

2.4.3 Reversing Interface Polarities

The parallel port buffers were selected to support the Centronics interface polarities: positive logic data and negative logic handshaking. However, pin-compatible buffers may be substituted for the buffers used on the parallel ports if positive-logic handshaking or negative-logic data are required. For negative-logic data, replace the appropriate 81Ø4/83Ø4 with an 81Ø3/83Ø3; for positive-logic handshaking, replace the appropriate 74LS367A with a 74LS368A.

2.5 THE RESET JUMPER

If the 2719 is used in a system which does not automatically assert SLVCLR* when RESET* is asserted, a jumper must be installed between the pads labelled RES. This is necessary primarily when the board is used in Cromemco systems. It is not necessary with CCS systems. If you are uncertain about whether the Reset Jumper is required, consult the system documentation; if you remain uncertain, call the system manufacturer.

2.6 INTERRUPT ACKNOWLEDGE WAITS

If the 2719 is used with a CCS 281Ø CPU, a wait state is required in all Interrupt Acknowledge cycles to ensure that the CTC has time to put its vector on the bus before the CPU tries to read it. To enable Interrupt Acknowledge wait states, install a jumper wire between the pads labeled RDY. This jumper is not required if the 2719 is used in a System 3ØØ/4ØØ. If the 2719 is used in a non-CCS system, this jumper may or may not be required; experiment, and install the jumper if necessary.

2.7 CLOCK PHASE

In some systems, including the CCS 221Ø (with the 281Ø CPU board), the system clock on bus pin 24 and the CPU clock are of opposite phase. Z-8Ø devices in a system must all have clocks of the same phase to work together. The Clock Phase Jumper allows the user to invert the phase of the system clock signal used on the 2719 as necessary. If the 2719 is used in a CCS System 3ØØ/4ØØ or any other system in which the bus clock is in phase with the processor clock, set the Clock Phase Jumper to the rightmost (Ø) position. If the 2719 is used in a system (including the CCS System 221Ø) featuring the CCS 281Ø CPU board or any other system in which the bus clock and processor clock are of opposite phase, set the Clock Phase Jumper to the leftmost (-Ø) position.

2.8 DATA LATCH ENABLING AND DISABLING

The DLE jumper has been included on the 2719 to allow disabling of the Data Out Latch. The jumper is hardwired in the EN position, which is required if the 2719 is used with a CCS 281Ø CPU. When the 2719 is used in a System 3ØØ/4ØØ, the jumper may be in either the EN or DSBL position. However, some CPU's may require that data coming onto the board not be latched. If you are using a non-CCS CPU, experiment, then disable the latching by cutting the EN trace and installing a jumper wire in the DSBL position if the 2719 cannot accept latched data from your CPU.

2.9 BAUD RATES IN 2 MHZ SYSTEMS

The CTC cannot accept a CLK/TRG input whose frequency is greater than half the system clock frequency. Thus, if you use the 2719 in a 2 MHz system, you must replace the 1.8432 MHz crystal pack in the lower right corner of the board with a 74LS74 dual flip-flop. The crystal pack is not socketed; you will need to unsolder its four pins and remove the solder from the other pads before installing the 74LS74. (If you install the 74LS74, then at a later date install the 2719 in a 4 MHz system, you need not replace the crystal pack unless you want to be able to select baud rates greater than 96ØØ.)

Note that installation of a 74LS74 will necessitate changes to the serial port drivers if they have been written for the 2719 with the crystal pack. See Section 3.3.

2.1Ø OPERATION IN 6MHZ SYSTEMS

To use the 2719 in a 6 MHz system, you must replace four chips: the Z-8ØA CTC with a Z-8ØB CTC; the Z-8ØA DART with a Z-8ØB DART (or Z-8ØB SIO/Ø for synchronous capability); and both 6821 PIAs with 68B21 PIAs. Some 2719 boards may be shipped with 68B21's, in which case replace ment of the PIAs will not be necessary; check the chips (not the silkscreen labels) before replacing the PIAs.

Baud rate programming will not change if the CTC is used in the counter mode with the 1.8432 MHz crystal pack. In the timer mode or with a 74LS74 installed instead of the crystal pack, multiply the 4 MHz values in Table 3-3 by 1.5 (75 baud will not be available).

CHAPTER 3

PROGRAMMING INFORMATION

This section is provided for those who wish to write their own drivers for the 2719's serial and/or parallel ports. Full instructions for programming the PIAs are given, as they may not be readily available. Programming options for the Z-8Ø SIO and CTC are quite elaborate, and are not given in this manual. Complete instructions are given in the Z-8Ø Family Programming Reference Manual included in CCS System 3ØØ/4ØØ documentation packages or available separately from CCS, as well as in a variety of other publications, a few of which are listed below. Only the programming limitations and requirements stemming from the implementation of the SIO and CTC on the 2719 are treated in this chapter.

AN INTRODUCTION TO MICROCOMPUTERS, Osborne and Associates, Inc. (Berkeley, CA: 1978).
ZILOG MICRCOMPUTER COMPONENTS DATA BOOK, Zilog, Inc. (Cupertino, CA: 198Ø).
MOSTEK MICROCOMPUTER DATA BOOK, Mostek Corporation (Carollton, TX: 1979).

3.1 PORT RELATIVE ADDRESSES

The base address of the 16 ports occupied by the 2719 is selected by the user as described in Chapter 2, but within the 16-port block the relative addresses of the ports are fixed. Table 3-1 shows the relative addresses of the CTC channels, the SIO data and command/status ports, and the PIA registers. Addresses in parentheses are the hexadecimal port addresses if the standard base address of 5ØH, required by CCS-supplied software, is used.

Table 3-1. PORT RELATIVE ADDRESSES

CTC:	CHANNEL Ø Base (5Ø)	CHANNEL 1 Base+1 (51)	CHANNEL 2 Base+2 (52)	CHANNEL 3 Base+3 (53)
DART:	A DATA Base+4 (54)	A COMMAND Base+5 (55)	B DATA Base+6 (56)	B COMMAND Base+7 (57)
PIA1:	A DATA/DIR Base+8 (58)	B DATA/DIR Base+9 (59)	A CONTROL Base+A (5A)	B CONTROL Base+B (5B)
PIA2:	A DATA/DIR Base+C (5C)	B DATA/DIR Base+D (5D)	A CONTROL Base+E (5E)	B CONTROL Base+F (5F)

3.2 DART 2719-UNIQUE PROGRAMMING

Table 3-2 shows which DART pins are connected to which RS-232-C lines. The RS-232-C signals are identified as DTE or DCE according to whether the Serial Port DCE/DTE Header is in the DTE or DCE position (see Section 2.3.2). The programmer should keep in mind that commmand and status bits in the programming guide are named for the DART pin and not the RS-232-C interface line--for example, the RTS command bit actually controls the CTS interface line when the Serial Port DCE/DTE Header is in the DTE position.

Table 3-2. DART/RS-232-C INTERFACING

SIO		RS-232-C (DTE)		RS-232-C (DCE)	
PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL
15/26	TxD	3	RD (BB)	2	TD (BA)
12/28	RxD	2	TD (BA)	3	RD (BB)
17/24	RTS	5	CTS (CB)	4	RTS (CA)
18/23	CTS	4	RTS (CA)	5	CTS (CB)
16/25	DTR	6	DSR (CC)	2Ø	DTR (CD)
19/22	DCD	2Ø	DTR (CD)	6	DSR (CC)

Programming limitations of the DART are listed below. They result from the fact that pins W/RDYA*, W/RDYB*, RIA*, and RIB* are not connected on the PC Board.

1. Bit 7 of Command Register 1 (Wait/Ready Enable) should be Ø. Bits 6 and 5 are don't-care bits.

2. Bit 4 of Status Register Ø will always be low. This will not affect External Status Interrupts.

3.3 CTC 2719-UNIQUE PROGRAMMING

The CTC on the 2719 is used to provide programmable clock signals for the DART and to generate Mode 2 interrupts for the PIAs. In the factory configuration, Channel Ø provides the clocks for Serial Port A, Channel 1 provides the clocks for Serial Port B, Channel 2 interrupts for PIA1, and Channel 3 interrupts for PIA2. Please note that, for each serial port, both the Receiver Clock and the Transmitter Clock are controlled by the same CTC signal and therefore cannot be independently programmed.

The specific programming requirements for the CTC on the 2719 are listed below.

1. Channels Ø and 1, which determine the serial port baud rates, may be programmed in either the counter mode or the timer mode. Interrupts should be disabled. Table 3-3 shows programming options for common baud rates.

2. If Mode 2 interrupts from the PIAs are desired, Channels 2 and 3 must be programmed in the counter mode with interrupts enabled, rising edges counted, and a time constant of 1. Thus the two bytes sent after the interrupt vector to initialize Channel 2 or 3 are D7H followed by Ø1H.

Table 3-3. TIME CONSTANTS FOR COMMON BAUD RATES

	TIME CONSTANTS					
	CRYSTAL		FLIP-FLOP			
BAUD RATE	4 MHZ		2 MHZ		4 MHZ	
	C	T	C	T	C	T
75		2Ø8		1Ø4		2Ø8
11Ø		142		71		142
134.5		116		58		116
15Ø		1Ø4		52		1Ø4
3ØØ		52		26		52
6ØØ	192	26	1Ø4	13	2Ø8	26
12ØØ	96	13	52		1Ø4	13
24ØØ	48		26		52	
48ØØ	24		13		26	
72ØØ	16				17	
96ØØ	12				13	
19.2K	6					
38.4K	3					
57.6K	2					
115.2K	1					

C = Counter Mode (Command Byte = 47H)
T = Timer Mode, Prescaler of 16 (Command = Ø7H)
DART Clock Rate is assumed to be 16x

3.4 PROGRAMMING THE PIA

Each PIA has six accessible (read and write) registers: two Data Registers, two Data Direction Registers, and two Command Registers. Register selection is determined by two Register Select inputs (RSØ, RS1) controlled by AØ and A1 and by Bit 2 of the Command Register. Table 3-4 shows how each register is selected.

Table 3-4. PIA REGISTER SELECTION

A1	AØ	CRA2	CRB2	REGISTER
Ø	Ø	1	X	Data Register A
Ø	Ø	Ø	X	Data Direction Reg A
Ø	1	X	1	Data Register B
Ø	1	X	Ø	Data Direction Reg B
1	Ø	X	X	Command Register A
1	1	X	X	Command Register B

3.4.1 PIA Command Registers

The Command Registers may be both read and written to. Written to, they determine all programmable operating parameters for the PIA channels except data direction. The current command and interrupt status (Bit 7) can be obtained by reading the Command Register. Command Register format is as follows:

CLR ON READ
SET ON L

	D7	D6	D5	D4	D3	D2	D1	DØ
	X	0	1	0	1	1	0	X (0
CRA	IRQA1	IRQA2	CA2 CONTROL			DDRA	CA1 CONTROL	
CRB	IRQB1	IRQB2	CB2 CONTROL			DDRB	CB1 CONTROL	

Bit Ø — This bit disables interrupts by CA1/CB1 when Ø, and enables interrupts by CA1/CB1 when 1. See Table 3-5.

Bit 1 — This bit selects the edge of CA1/CB1 (the ACKA*/ACKB* interface line) which will set Bit 7 of the Command Register, a 1 selecting the rising edge and a Ø selecting the falling edge. See Table 3-5.

Table 3-5. CA1/CA2 INPUT CONTROL

CRx-1	CRx-2	INT FLAG CRx-7	INT OUTPUT IRQx*
Ø	Ø	Set high by high-to-low transition of Cx1	Disabled; stays high
Ø	1	Set high by high-to-low transition of Cx1	Goes low when CRx-7 goes high
1	Ø	Set high by low-to-high transition of Cx1	Disabled; stays high
1	1	Set high by low-to-high transition of Cx1	Goes low when CRx7 goes high

Bit 2 If this bit is 1, a Data Register is accessed at the Data/Data Direction address; if it is Ø, a Data Direction Register is addressed.

Bits 3-4 These bits control output CA2/CB2 as indicated in Table 3-6. Channel A timing uses negative edges of E, while Channel B uses positive edges.

Table 3-6. CA2/CB2 OUTPUT CONTROL

CR4	CR3	Cx2 FUNCTION
Ø	Ø	Set by Cx1 going active; cleared by read (Channel A) or write (Channel B)
Ø	1	Pulses low immediately after read (Channel A) or write (Channel B)
1	Ø	Always low
1	1	Always high

Bit 5 This bit must be 1 to condition CA2/CB2 as an output. On the 2719 CA2/CB2 may not be used as an interrupt input.

Bit 6 This is the CA2/CB2 interrupt flag. Because on the 2719 CA2/CB2 cannot be used as an interrupt input, this bit will always be Ø. This bit is not affected by a write to the Command Register.

Bit 7 This is the CA1/CB1 interrupt flag. When this bit is 1, IRQA*/IRQB* has been asserted by the appropriate transition of CA1/CB1. This bit is cleared when the channel's Data Register is read and is not affected by a write to the Command Register.

3.4.2 PIA Data Direction Registers

Bits Ø-7 of the Data Direction Register control the direction of data lines Ø-7 respectively. If a bit is Ø, the corresponding data line is an input; if a bit is 1, the corresponding data line is an output. A Data Direction Register can be accessed only if Bit 2 of the Command Register for the same channel is Ø. Because of the way the PIA data lines are buffered on the 2719, ALL BITS OF A CHANNEL MUST BE PROGRAMMED FOR THE SAME DIRECTION and the data direction programmed for a channel must agree with the setting of the corresponding Parallel Port Data Direction Jumper as described in Section 2.4.1.

3.4.3 PIA INITIALIZATION: CENTRONICS CONFIGURATION

The following sequence initializes a parallel port in the standard Centronics configuration.

a. Output ØØH to both Ch. A and Ch. B Control Ports to select DDR.
b. Output to Data/Dir Ports (ØØH to Ch. A and ØFFH to Ch. B) to select direction.
c. Output 2CH (or 2DH for interrupts) to both Ch. A and Ch. B Control Ports to set the PIA mode.
d. Input from both Ch. A and Ch. B Data/Dir Ports to clear the status bits.

CHAPTER 4

HARDWARE DESIGN

Two 6821 PIAs and a Z-8Ø DART provide the two parallel and two serial ports of the 2719. Most of the interface functions are provided by these three chips. A Z-8Ø CTC is employed to generate baud rate clocks for the two DART channels and Mode 2 interrupts for the two PIAs. Additional logic supports interrupt capability in all three Z-8Ø modes, addresses and controls the CTC, DART, and PIAs, and controls data buffering. Thus the 2719 can be divided into six functional elements: the PIAs, the DART, the CTC, the interrupt logic, the address/control logic, and data buffering. Each element is separately described below.

4.1 THE PIA'S

Each 6821 PIA provides two parallel data channels, programmable for input or output on a bit-by-bit basis, with programmable two-line handshaking for each channel. Programming options are discussed in Chapter 3. PIA inputs and outputs are defined in Table 4-1. For additional information see a 6821 data sheet.

Each PIA data channel is buffered by an 83Ø4 bi-directional driver/receiver, the direction of data flow being jumper-selectable but pre-configured for Channel A as input and Channel B as output. The A and B data buffers are enabled by lows on interface lines 3Ø and 19 respectively, or may be permanently enabled by the installation of a jumper. Handshaking is determined to consist of one input (CA1/CB1) and one output (CA2/CB2) by the buffers on the handshake lines. The on-board reset signal is buffered onto the parallel interface to provide a reset signal for the peripheral.

When used as hardwired, the parallel ports interface with Centronics-type peripherals. Other types of parallel interface devices may be interfaced, however. Hardware and software options are discussed in Sections 2.4 and 3.5 respectively. Interface pinouts are shown in Section A.3.

TABLE 4-1. PIA SIGNALS

SIGNAL	FUNCTION
E	Enable is the PIA's timing signal.
DØ-7	The bi-directional data pins connect directly to the 2719's internal data bus.
R/W*	R/W* controls the direction of data transfer.
RESET*	This input low clears all registers.
CSØ,CS1, CS2*	The Chip Select inputs must all be active for the PIA to be selected.
RSØ RS1	These inputs determine which PIA register will be accessed.
IRQA* IRQB*	These are the Interrupt Request outputs for the two PIA channels.
PAØ-7 PBØ-7	These are the bi-directional data pins for the the two PIA channels.
CA1 CB1	These handshaking inputs set the interrupt flags.
CA2 CB2	These pins are used on the 2719 as handshaking outputs.

4.2 THE DART

A Z-8Ø DART provides two extensively programmable asynchronous serial ports. The port interface, as implemented on the 2719, consists of Transmitter Data and Receiver Data lines and four handshaking lines. The handshaking lines are connected to RS-232-C lines RTS, CTS, DTR, and DSR, as indicated in Table 3-2, the Serial Interface Headers allowing configuration of the interface as either DCE or DTE, as well as allowing for non-standard handshaking using lines 19 or 11. Table 4-2 defines the DART inputs and outputs.

4.3 THE CTC

The Z-8Ø CTC consists of four separately programmable counter/timer circuits. Each circuit includes a downcounter, a time constant register, and a prescaler. In the timer mode, the downcounter is loaded with the programmed time constant, then decremented with every 16 or 256 pulses of the 4 MHz system clock (depending on the prescaler selected). In the counter mode, the downcounter is loaded with the time constant, then decremented with every pulse of the channel's CLK/TRG input, the prescaler having no effect. Channels Ø-2 have ZC/TO (Zero Count/Time Out) outputs that pulse high when the downcounters reach zero; all four channels can be programmed to interrupt when their downcounters reach zero. In addition, downcounter contents can be read from the channel address without disturbing the counting. Table 4-3 defines the CTC inputs and outputs.

On the 2719, CTC Channels Ø and 1 are used to supply the DART Channel A and Channel B clocks respectively, while Channels 2 and 3 are used to bring the PIA interrupts into the Z-8Ø Mode 2 Interrupt Daisy Chain. Channels Ø and 1 thus may be programmed in either the counter or timer mode, as described in Section 3.3. The CLK/TRG inputs of Channels Ø and 1 are controlled by a 1.8432 MHz crystal. (Note that jumpers allow each DART channel's transmitter and receiver clocks to be controlled by RS-232-C line 15 if the DART channel functions as the DTE device and the user desires the baud rate to be controlled by the DCE device.)

Whenever either of the PIA1 interrupt request outputs IRQA* and IRQB* goes low, the CLK/TRG input to CTC Channel 2 goes high. The CLK/TRG input to Channel 3 is similarly controlled by the PIA2 interrupt request outputs. If these channels are programmed in the counter mode as described in Section 3.3, the CTC will generate the actual interrupt request whenever a PIA signals that it wants one. In Interrupt Mode 2, this means that an Interrupt Vector pointing to the appropriate PIA service routine can be gated

TABLE 4-2. DART SIGNALS

SIGNAL	FUNCTION
CE* DØ-7 IORQ* M1* RD*	See Table 4-3 for definitions of these signals.
B/A*	This input, controlled by AØ, determines whether Channel A or Channel B is selected.
C/D*	This input, controlled by A1, determines whether a control or data transfer will occur.
TxDA TxDB	Serial data at TTL levels is output to interface lines RxD.
RxDA RxDB	Serial data at TTL levels is input at these pins via the TxD interface lines.
CTSA* CTSB*	The Clear To Send inputs, connected to the RTS RS-232-C lines, may be programmed as transmitter auto-enable or general-purpose signals.
RTSA* RTSB*	The Request to Send handshaking outputs are connected to the CTS RS-232-C lines.
DCDA* DCDB*	The Data Carrier Detect inputs, connected to the DTR RS-232-C lines, may be programmed as receiver auto-enable or general purpose inputs.
DTRA* DTRB*	The Data Terminal Ready handshaking outputs are connected to the DSR RS-232-C lines.
RIA* RIB*	These pins are not connected on the 2719.
RxCA TxCA RxTxCB	The Channel A and Channel B clocks are controlled by CTC Channel Ø and Channel 1 respectively.
RESET*	A low at this pin resets both DART channels.
CLK	This is the DART's system clock input.
INT* IEI IEO	See Section 4.5 for a discussion of these interrupt daisy chain signals.

TABLE 4-3. CTC SIGNALS

SIGNAL	FUNCTION
CE*	Chip Enable is controlled by the address decode circuitry.
RD*	The Read input determines the direction of data transfer.
IORQ*	The I/O Request input enables data transfer.
M1*	Both M1* and IORQ* low indicates an Interrupt Acknowledge cycle.
CSØ,CS1	The Channel Select inputs select one of four CTC channels. They are controlled by AØ-A1.
CLK/TRG Ø-3	The Clock/Trigger inputs control downcounter decrementing in counter mode. CLK/TRGØ-1 are controlled by the crystal or the system clock divided by two. CLK/TRG2-3 are controlled by the PIA IRQ* outputs.
ZC/TO Ø-2	The Zero Count/Timeout pins pulse high when the downcounters reach zero. ZC/TOØ-1 control the DART receiver and transmitter clocks.
DØ-7	The bi-directional data pins connect directly to the 2719 internal data bus.
RESET*	Reset low terminates downcounting, disables interrupts, and tri-states DØ-D7.
CLK	This is the CTC's system clock input.
INT* IEI IEO	See Section 4.5 for a discussion of these interrupt daisy chain signals.

onto the bus during the Interrupt Acknowledge cycle. The IMØ and IM1,2 Headers allow direct assertion of PIA interrupt requests in Interrupt Modes Ø and 1, in which the special interrupt capabilities of a Z-8Ø device are not required.

4.4 ADDRESS AND CONTROL LOGIC

During I/O cycles, A7-AØ carry the I/O port address, selecting one of 256 I/O ports. Each of the four major components of the 2719 (the DART, the CTC, and the two PIAs) occupies four ports. The 2719 is designed so that the ports of the four devices occupy absolute locations relative to each other in any 16-port block whose base is a multiple of 16. For an on-board device to be selected, A7-A4 must match the settings of the Base Address Jumpers, either sINP or sOUT must be active, and sINTA must be inactive. When these conditions are met, the internal signal BDSEL (Board Select) goes active.

BDSEL is the CSØ input to each PIA and is input to the CTC and DART Select Gates (U1Øa and b), the outputs of which control the CTC and DART Chip Enable inputs. When BDSEL is active, A3 and A2 determine which device is enabled, controlling PIA inputs CS1 and CS2* and being input to the CTC and DART Select Gates. Table 4-4 shows the states of A3 and A2 required to enable the various chips.

Table 4-4. CHIP SELECTION

A3	A2	CHIP SELECTED
Ø	Ø	CTC
Ø	1	DART
1	Ø	PIA1
1	1	PIA2

Except for the Enable inputs to the PIAs, the rest of the control logic for the four devices is relatively straightforward. The R/W* inputs to the PIAs are controlled directly by pWR*, while the RD* (essentially R*/W) inputs to the DART and CTC are low when pDBIN is active during non-Interrupt-Acknowledge cycles and high at all other times. CTC and DART inputs M1* and CLK are controlled directly by the sM1 and o2 bus lines; IORQ* is active when one of sINTA, sINP, or sOUT is active. The Enable inputs to the PIAs are required to be active for 166 nanoseconds during read and write operations, but need not be synchronized to the system clock. On the 2719 they are active whenever either pWR* or pDBIN is active. With a 4 MHz or slower system clock, Enable pulses of adequate duration are guaranteed.

The 2719 is wired so that the CTC, the DART, and the PIAs are all reset when either of bus signals POC* or SLVCLR* goes active. The IEEE standards for the S-1ØØ bus specify that SLVCLR* should be asserted whenever RESET* is asserted. SLVCLR*, when asserted by RESET*, is removed slightly before RESET* to ensure that bus slaves finish the reset before the bus master comes up. Thus it is advantageous for a peripheral board to be reset by SLVCLR* if possible. However, in some systems SLVCLR* is not automatically asserted when RESET* is asserted. When the 2719 is used in such systems it will be necessary to install a jumper between the pads labelled RES to cause the 2719 to be reset directly by RESET* active.

4.5 INTERRUPT LOGIC

The Z-8Ø CPU is capable of three modes of maskable interrupt response, the mode in which the CPU operates at a given time being determined by software. The three modes are defined in the CPU section of the Z-8Ø Family Programming Reference Manual. Mode Ø is the 8Ø8Ø interrupt mode, which requires that an interrupt controller be part of the system. An interrupting device asserts one of the VI* lines, which is sensed by the Interrupt Controller; the Interrupt Controller then asserts INT* and puts out the programmed instruction (usually a restart or call). In Mode 1 the interrupting device asserts INT* directly, causing the CPU to execute a restart at location ØØ38H. In Mode 2, the special Z-8Ø mode, the highest-priority interrupting device puts the vector for its interrupt service routine onto the bus during the Interrupt Acknowledge cycle.

In support of Mode 2 interrupts, the Z-8Ø peripherals have IEI and IEO (Interrupt Enable In and Out) pins which allow them to be linked in a hardware-prioritizing interrupt daisy chain. The highest-priority device's IEO is connected to the next-highest-priority device's IEI. If a device's IEI input is high, it may generate an interrupt request by forcing INT* low. A device's IEO output is forced low if either its IEI pin or its INT* pin is low. Thus a device generating an interrupt request disables the interrupt request logic of all lower-priority devices in the daisy chain. Higher-priority devices are unaffected, however, and may interrupt at any time, providing that CPU interrupts are enabled.

If more than four devices are connected in a simple daisy chain, a low-priority interrupt request may not be disqualified by a higher-priority interrupt request soon enough to prevent the low-priority device from thinking its interrupt is being acknowledged and outputting its interrupt vector. The 2719 supports an extended Z-8Ø Mode 2 Interrupt Daisy Chain with a look-ahead scheme implemented on all CCS System 3ØØ/4ØØ peripheral boards. The VI* bus lines are used to prioritize the boards participating in the chain. Each board's look-ahead logic guarantees that an interrupt request by an on-board device or by a device on a higher-priority board will be passed on to lower-priority boards rapidly enough to be recognized before the interrupt is acknowledged, preventing two devices from putting their interrupt vectors on the bus at the same time.

On the 2719, if the DART requests an interrupt, the CTC is prevented from interrupting by the normal chip-to-chip IEI-IEO daisy chain. An interrupt by a higher-priority board does not ripple through the DART, however; the interrupt request logic of both the DART and the CTC is immediately disabled when the 2719 interrupt logic senses any higher-priority VI* line going low. Whenever an on-board device interrupts or the 2719 senses a higher-priority VI* line low, the 2719 forces its priority-assertion VI* line low to immediately disable lower-priority boards. The particular VI* lines the 2719 senses and the VI* line it asserts are determined by the configuration of the IM2 Header (see Section 2.2.3).

In some environments it will be desirable for the 2719 to generate Mode Ø or Mode 1 interrupts. The Mode Ø Header allows the user to select which VI* line each on-board device will use to communicate its interrupt request to the Interrupt Controller. The Mode 1,2 Header allows each device to directly assert INT*. Configuration instructions for the interrupt headers are given in Section 2.2.

In some systems, including those using a CCS 281Ø CPU, a wait may be required during Interrupt Acknowledge cycles to ensure that the CTC has time to get its interrupt vector onto the bus. For this reason, jumper-enabled circuitry has been provided to force bus line RDY low while pSYNC is active during Interrupt Acknowledge cycles in which the 2719 has the highest-priority interrupt pending.

4.6 DATA BUFFERING AND LATCHING

All system bus inputs and outputs are fully buffered, as are the serial and parallel port interface lines. Hysteresis drivers and receivers are used for system bus interfacing, ensuring minimum noise on the bus. No load of more than one Low-Power Schottky TTL level is placed on any system bus input. INT* and VI* line drivers are open-collector. Serial port interface drivers and receivers meet EIA RS-232-C specifications.

Except for the parallel port buffers, discussed in Section 4.1, and the Data In Buffer, all buffers are permanently enabled. The Data In Buffer and Data Out Latch (described below) are alternately tri-stated except during Interrupt Acknowledge cycles. The Data Out Latch is always disabled during Interrupt Acknowledge cycles. The Data In Buffer is enabled during Interrupt Acknowledge cycles if the board is configured for Mode 2 interrupts and the 2719 has the highest-priority request pending; this is necessary to allow the Interrupt Vector onto the bus. However, in Interrupt Mode Ø or 1 it is necessary to keep the Interrupt vector off the bus. If the IMS Jumper is wired for the Ø/1 position, the Data In Buffer is disabled during all Interrupt Acknowledge cycles. In all non-Interrupt-Acknowledge cycles, the Data Out Latch is disabled and the Data In Buffer enabled for the duration of pDBIN active when the board is addressed during an I/O Read cycle; otherwise the Data In Buffer is tri-stated and the Data Out Latch is enabled.

The Data Out Latch has been included to defeat the multiplexing of status and data on the DO lines by the CCS 281Ø CPU, thus allowing the 2719 to' be used in the CCS System 221Ø. The CTC cannot handle the 281Ø's multiplexed status and data. When bus signal sOUT goes inactive, the data is latched (provided that the latching signal is not jumper-disabled) until one of three things occurs: 1) sINTA goes active; 2) an I/O Read cycle occurs, enabling the Data In Buffer; or 3) another I/O Write cycle occurs, with sOUT going active, gating new data onto the bus. Because the CTC's RD* input is always low unless pDBIN is active and sINTA is inactive, the CTC may read valid data for as long as it is enabled during an I/O Write cycle.

When a device has interrupted, it monitors the data bus during all instruction fetches, looking for the RETI instruction which signals that servicing of the interrupt has been completed and that lower-priority interrupts may now be asserted. To ensure proper monitoring of the CPU instruction fetches by the 2719, the CPU must internally connect the DI and DO buses. This is necessary because, while the RETI instruction is input to the CPU on the DI lines, the 2719 looks for RETI on the DO lines. The CCS 282Ø System Processor Board meets this requirement as long as no other board asserts DODSB during instruction fetch cycles.

APPENDIX A

TECHNICAL INFORMATION

A.1 USER-REPLACEABLE PARTS

QTY	REF	DESCRIPTION	CCS PART #
Integrated Circuits			
1	U3	Z-8ØA DART	482ØØ119-Ø1
1	U6	Z-8ØA CTC	482ØØØ84-Ø1
2	U16,17	6821 PIA	482ØØØ76-Ø1
3	U24,28,29	74LS136 quad EX-OR	482ØØØ21-Ø1
1	U23	74LS13 dual 4-in NAND	482ØØ12Ø-Ø1
2	U26,3Ø	74LS24Ø oct buffers, inv	482ØØØ34-Ø1
1	U22	74LSØ8 quad 2-in AND	482ØØØØ6-Ø1
3	U27,31,33	74LS244 octal buffers	482ØØØ35-Ø1
1	U21	74LSØ2 quad 2-in NOR	482ØØØØ2-Ø1
1	U2Ø	74LSØØ quad 2-in NAND	482ØØØØ1-Ø1
4	U7,9,1Ø,12	83Ø4B transceiver	482ØØØ68-Ø1
1	U19	74Ø7 hex buffer, OC	482ØØØ51-Ø1
1	U28	74LS1Ø tri 3-in NAND	482ØØØØ8-Ø1
1	U25	74LS11 tri 3-in AND	482ØØ125-Ø1
3	U5,14,15	7515Ø line driver	482ØØØ55-Ø1
2	U4,13	75154 line receiver	482ØØØ56-Ø1
1	U1	79L12 -12V regulator	482ØØ115-Ø1
1	U2	78L12 +12V regulator	482ØØ127-Ø1
2	U8,11	74LS367A hex drivers,	482ØØØ39-Ø1
1	U32	74LS373 octal D latches	482ØØØ41-Ø1
1	U18	LM323K +5 V regulator	482ØØ1Ø7-Ø1
1	Y1	Oscillator, 1.8432 MHz	482ØØ118-Ø1
Capacitors			
13	C1,7-12,14-19	.1uf Mono, 5ØVDC, 2Ø%	159ØØØØ1-Ø1
6	C2-6,13	4.7uf Tant, 35VDC, 2Ø%	155ØØØØ3-Ø1
Resistors			
2	R1,2	2.7K ohm, 1/4 Watt, 5%	47ØØØØ23-Ø1
3	Z1-3	SIP Network, 2.7K x 7	474ØØØØ2-Ø1

QTY	REF	DESCRIPTION	CCS PART #
Miscellaneous			
6		Socket, IC, 16 pin	214ØØØ15-Ø1
4		Socket, IC, 2Ø pin	214ØØØ17-Ø1
1		Socket, IC, 28 pin	214ØØØ18-Ø1
3		Socket, IC, 4Ø pin	214ØØØ2Ø-Ø1
2		Header, 2 x 13 rt angle	21ØØØØ17-Ø1
2		Header, 2 x 17 rt angle	21ØØØØ18-Ø1
1		Header, 1 x 4	21ØØØØØ9-Ø1
1		Header, 1 x 3	21ØØØØØ7-Ø1
1		Header, 2 x 4	21ØØØØ23-Ø1
5		Jumper Plug	213ØØØ21-Ø1
3		Header, DIP, 16 pin	216ØØØØ1-Ø1
3		Header Cover, 16 pin	141ØØØØ1-Ø1
1		Heatsink, TO-3	76ØØØØØ3-Ø1
1		Heatsink Insulator	76ØØØØØ4-Ø1
2		Screw, 6-32 x 3/8	28ØØØØØ6-Ø1
2		Nut, 6-32 KEP	281ØØØØ4-Ø1
2		Board extractor	74ØØØØØ1-Ø1
2		Roll pin	283ØØØØ1-Ø1
2		Cable, 26 pin, 24" DB25	6Ø9ØØØØ5-Ø1
2		Cable, 34 pin, 24"	6Ø9ØØØØ9-Ø1

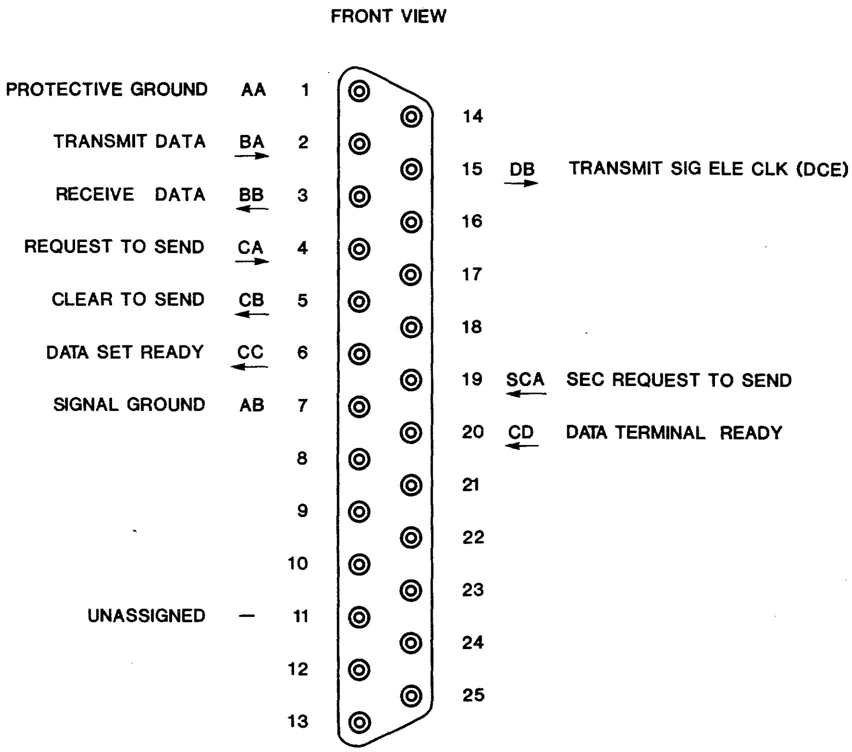
A.2 SERIAL CONNECTOR PINOUTS

PGND	AA	1	o	o	2		
TxD	BA	3	o	o	4	CB	TSET (DCE)
RxD	BB	5	o	o	6		
CTS	CB	7	o	o	8		
RTS	CA	9	o	o	1Ø		
DSR	CC	11	o	o	12	SCA	SRTS
SGND	AB	13	o	o	14	CD	DTR
		15	o	o	16		
		17	o	o	18		
		19	o	o	2Ø		
	NDEF	21	o	o	22		
		23	o	o	24		
		25	o	o	26		

A.3 PARALLEL CONNECTOR PINOUTS

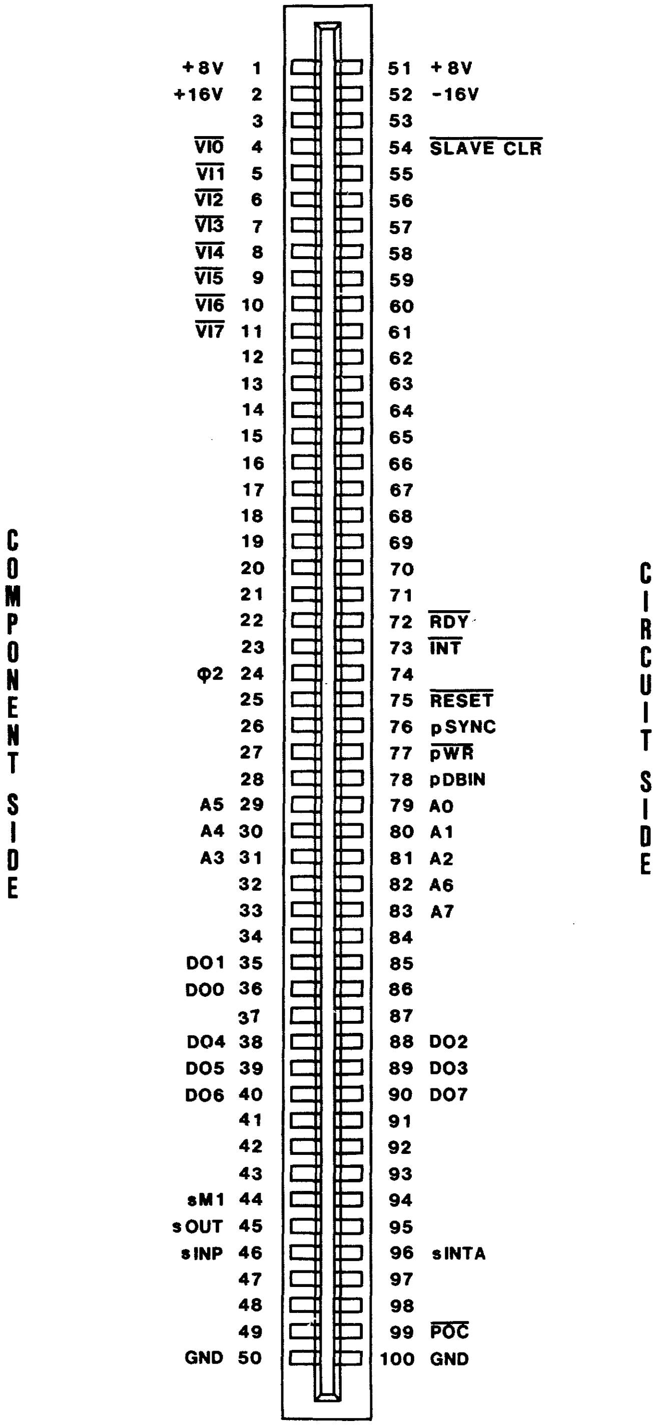
BSTB	1	o	o	2	BEN
BDØ	3	o	o	4	SIG GND
BD1	5	o	o	6	SIG GND
BD2	7	o	o	8	SIG GND
BD3	9	o	o	1Ø	SIG GND
BD4	11	o	o	12	SIG GND
BD5	13	o	o	14	SIG GND
BD6	15	o	o	16	SIG GND
BD7	17	o	o	18	SIG GND
BACK	19	o	o	2Ø	SIG GND
ADØ	21	o	o	22	SIG GND
AD1	23	o	o	24	AEN
AD2	25	o	o	26	RESET
AD3	27	o	o	28	AD4
ASTB	29	o	o	3Ø	AD5
AACK	31	o	o	32	AD6
	33	o	o	34	AD7

A.4 RS-232-C CONNECTOR PINOUTS



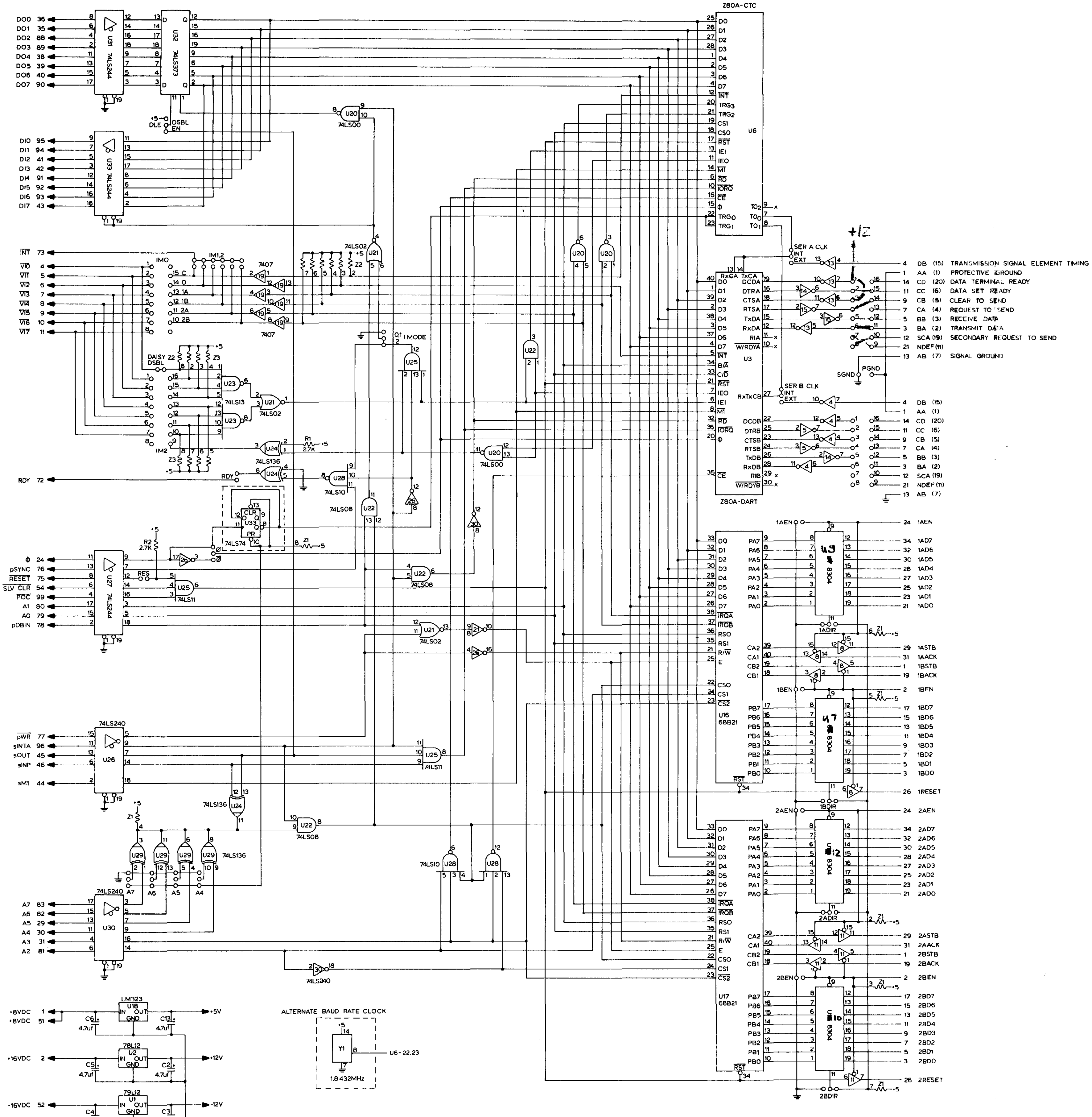
DB-25S (FEMALE)

A.5 BUS CONNECTOR PINOUTS



TOP VIEW

A.6 SCHEMATIC/LOGIC DIAGRAM



MODEL 2719
©CCS 1981
2 PARALLEL / 2 SERIAL
REV B

APPENDIX B

SAMPLE DRIVERS

Drivers for the 2719 parallel and serial ports will be included with future releases of CP/M with the CCS System 221Ø and OASIS with the CCS Systems 3ØØ and 4ØØ. If your operating system does not include drivers for the 2719, you will need to add them. The sample drivers that follow may be used as they are or modified as desired. For instructions on adding the drivers to your operating systems, see the relevant operating system documentation. For CP/M on the System 3ØØ or 4ØØ, see the System 3ØØ or System 4ØØ CP/M Supplementary Manual; for OASIS, see Phase One's Macro Assembler Language Reference Manual.

Please note that the ORG addresses used in the sample driver listings were chosen for assembly purposes and are not the addresses at which your drivers will reside.

B.1 CP/M DRIVER

```
0100                 ORG     100H
             ;
             ; SAMPLE DRIVER CODE FOR THE CCS MODEL 2719
             ;       2 SERIAL/ 2 PARALLEL INPUT/OUTPUT INTERFACE BOARD
             ;
             ; The input/output drivers shown below work in a polled
             ;       environment.  They conform to the CP/M interface
             ;       specifications as documented in the CP/M Alteration
             ;       guide, and are intended for CP/M or MP/M applications.
             ;
             ; Port Address Assignments
             ;
0050 =       BASE19: EQU     50H      ; Base address of the 2719 board
             ;
0050 =       CTC0:   EQU     BASE19   ; CTC base address
0051 =       CTC1:   EQU     CTC0+1   ; CTC #1
0052 =       CTC2:   EQU     CTC0+2   ; CTC #2
0053 =       CTC3:   EQU     CTC0+3   ; CTC #3
             ;
0054 =       SIOAD:  EQU     BASE19+4  ; SIO Channel A Data Register
0055 =       SIOAC:  EQU     SIOAD+1  ; SIO Channel A Command Register
0055 =       SIOAS:  EQU     SIOAC    ; SIO Channel A Status Register
0056 =       SIOBD:  EQU     BASE19+6  ; SIO Channel B Data Register
0057 =       SIOBC:  EQU     SIOBD+1  ; SIO Channel B Command Register
0057 =       SIOBS:  EQU     SIOBC    ; SIO Channel B Status Register
             ;
0058 =       PIA1AD: EQU     BASE19+8  ; PIA1 Channel A Data Register
0058 =       PIA1ADD:  EQU   PIA1AD   ; PIA1 Channel A Data Direction Register
0059 =       PIA1BD: EQU     BASE19+9  ; PIA1 Channel B Data Register
0059 =       PIA1BDD:  EQU   PIA1BD   ; PIA1 Channel B Data Direction Register
005A =       PIA1AC: EQU     PIA1AD+2  ; PIA1 Channel A COMMAND Register
005A =       PIA1AS: EQU     PIA1AC   ; PIA1 Channel A STATUS Register
005B =       PIA1BC: EQU     PIA1BD+2  ; PIA1 Channel B Command Register
005B =       PIA1BS: EQU     PIA1BC   ; PIA1 Channel B Status Register
             ;
005C =       PIA2AD: EQU     BASE19+12  ; PIA2 Channel A Data Register
005C =       PIA2ADD:  EQU   PIA2AD   ; PIA2 Channel A Data Direction Register
005D =       PIA2BD: EQU     BASE19+13  ; PIA2 Channel B Data Register
005D =       PIA2BDD:  EQU   PIA2BD   ; PIA2 Channel B Data Direction Register
005E =       PIA2AC: EQU     PIA2AD+2  ; PIA2 Channel A Command Register
005E =       PIA2AS: EQU     PIA2AC   ; PIA2 Channel A Status Register
005F =       PIA2BC: EQU     PIA2BD+2  ; PIA2 Channel B Command Register
005F =       PIA2BS: EQU     PIA2BC   ; PIA2 Channel B Status Register
             ;
002C =       PIAMOD: EQU     00101100B  ; PIA operating mode byte
```

```
00FF =          DATOUT: EQU     11111111B  ; PIA Data Direction byte for output
                ;
                ; The following equates establish the baud rates for the serial
                ;       channels.  Three sets of values are identified, and must
                ;       be set to match the specific board configuration.
                ;
                ;       The normal configuration is for 4 mhz operation with a
                ;       crystal oscillator.  The other two are for deriving the
                ;       baud rate clock from the system clock divided by 2 by
                ;       U34 (74LS74).  See the Manual text for further detail.
                ;
                ;       BAUD    CTCMOD          CTCDIV
                ;                       XTAL    2mhz    4mhz
                ;       75      07H     208     104     208
                ;       110     07H     142     71      142
                ;       134.5   07H     116     58      116
                ;       150     07H     104     52      104
                ;       300     07H      52     26       52
                ;       600     47H     192     104     208
                ;       1200    47H      96      52     104
                ;       1800    47H      64      35      69
                ;       2000    47H      58      31      63
                ;       2400    47H      48      26      52
                ;       3600    47H      32      17      35
                ;       4800    47H      24      13      26
                ;       7200    47H      16       9      17
                ;       9600    47H      12     ---      13
                ;       19200   47H       6     ---     ---
                ;       38400   47H       3     ---     ---
                ;       57600   47H       2     ---     ---
                ;       115200  47H       1     ---     ---
                ;
0047 =          CTCMOD: EQU     47H     ; CTC mode for 9600 baud
000C =          CTCDIV: EQU     12      ; CTC divisor for 9600 baud (XTAL)
                ;
                ; for the SIO command byte definitions, see the text.
                ;
                ;----------------------------------------------------------------------
                ;
                ; The following code segment initializes the SIO channels for
                ;       asynchronous operation with auto-enables.
                ;
                ;       It must be placed in the cold boot code path.
                ;
                ;        .
                ;        .
                ;        .
```

```
0100 211B01     SIOINIT:  LXI    H,SIOCMD  ; point to the SIO Init Data String
0103 0606                 MVI    B,SIOLGTH ; get the String length
0105 7E         SIOI1:    MOV    A,M       ; get the next init command
0106 D355                 OUT    SIOAC     ; output to SIO Channel A
0108 D357                 OUT    SIOBC     ;   and to Channel B
010A 23                   INX    H         ; advance string pointer
010B 05                   DCR    B         ; check the loop control
010C C20501               JNZ    SIOI1     ; jump if more to do
010F 3E47                 MVI    A,CTCMOD  ; now, set the baud rate generator
0111 D350                 OUT    CTC0      ;  Channel A
0113 D351                 OUT    CTC1      ;  Channel B
0115 3E0C                 MVI    A,CTCDIV  ; baud rate divisor
0117 D350                 OUT    CTC0      ;  Channel A
0119 D351                 OUT    CTC1      ;  Channel B
                ;          .
                ;          .
                ;          .
                ;
                ; The SIO Initialization Data String should be put in
                ;         the data area of the BIOS or XIOS.
                ;
                SIOCMD:
011B 04                   DB     4         ; Access Write Register 4
011C 46                   DB     01000110B  ; x16 clock, 1 stop bit, no parity
011D 05                   DB     5         ; Access Write Register 5
011E EA                   DB     11101010B  ; DTR, Tx 8 bits, Tx Enable, RTS
011F 03                   DB     3         ; Access Write Register 3
0120 E1                   DB     11100001B  ; Rx 8 bits, Auto enables, Rx enable
0006 =          SIOLGTH:  EQU    $-SIOCMD  ; Init Command string length
                ;
                ; The following code segment initializes both PIAs for
                ;         Centronics-compatable printer operation
                ;
                ;         It also must be put into the cold boot code.
                ;
                ;          .
                ;          .
                ;          .
                PIAINIT:
0121 AF                   XRA    A         ; get a zero into (A)
0122 D35A                 OUT    PIA1AC    ;  allows access to the data direction register
0124 D35B                 OUT    PIA1BC
0126 D35E                 OUT    PIA2AC
0128 D35F                 OUT    PIA2BC
012A D358                 OUT    PIA1AD    ;   sets the A side for input
012C D35C                 OUT    PIA2AD
012E 3EFF                 MVI    A,DATOUT  ; direction control byte for output
```

```
0130 D359              OUT     PIA1BD  ;   sets the B side for output
0132 D35D              OUT     PIA2BD
0134 3E2C              MVI     A,PIAMOD  ; PIA mode control byte
0136 D35A              OUT     PIA1AC  ; sets the PIA operating mode
0138 D35B              OUT     PIA1BC
013A D35E              OUT     PIA2AC
013C D35F              OUT     PIA2BC
013E AF                XRA     A       ; get a null character
013F D359         IN ~~OUT~~     PIA1BD  ;  output it to prime the handshake lines
0141 D35D         IN ~~OUT~~     PIA2BD
                ;       .
                ;       .
                ;       .
                ;
                ; Serial Driver Routines
                ;
                SIOAST:                ; SIO Channel A input status routine
0143 DB55              IN      SIOAC   ; read the SIO status byte
0145 E601              ANI     00000001B  ; see if Rx Character available
0147 C8                RZ              ; done if not
0148 F6FF              ORI     0FFH    ; else, flag the ready condition
014A C9                RET
                ;
                SIOAIN:                ; SIO Channel A Input Routine
014B CD4301            CALL    SIOAST  ; check the port status
014E C24B01            JNZ     SIOAIN  ; try again if not ready
0151 DB54              IN      SIOAD   ; get the data byte
0153 E67F              ANI     7FH     ; strip off bit 7
0155 C9                RET
                ;
                SIOAOST:               ; SIO Channel A Output Status Routine
0156 DB55              IN      SIOAC   ; read the SIO status byte
0158 E608              ANI     00001000B  ; check the DCD bit (handshake)
015A C8                RZ              ; return if not ready
015B DB55              IN      SIOAC   ; reget the SIO status byte
015D E604              ANI     00000100B  ; see if Tx Buffer is empty
015F C8                RZ              ; no, still busy
0160 F6FF              ORI     0FFH    ; else, flag the ready condition
0162 C9                RET
                ;
                SIOAOUT:               ; SIO Channel A output routine
0163 CD5601            CALL    SIOAOST ; see if port ready for output
0166 CA6301            JZ      SIOAOUT ; try again if not
0169 79                MOV     A,C     ; else, get the data for output
016A D354              OUT     SIOAD   ; output it
016C C9                RET
                SIOBST:                ; SIO Channel B input status routine
```

```
016D DB57               IN      SIOBC   ; read the SIO status byte
016F E601               ANI     00000001B  ; see if Rx Character available
0171 C8                 RZ              ; done if not
0172 F6FF               ORI     0FFH    ; else, flag the ready condition
0174 C9                 RET
                ;
                SIOBIN:                 ; SIO Channel B Input Routine
0175 CD6D01             CALL    SIOBST  ; check the port status
0178 C27501             JNZ     SIOBIN  ; try again if not ready
017B DB56               IN      SIOBD   ; get the data byte
017D E67F               ANI     7FH     ; strip off bit 7
017F C9                 RET
                ;
                SIOBOST:                ; SIO Channel B Output Status Routine
0180 DB57               IN      SIOBC   ; read the SIO status byte
0182 E608               ANI     00001000B  ; check the DCD bit (handshake)
0184 C8                 RZ              ; return if not ready
0185 DB57               IN      SIOBC   ; reget the SIO status byte
0187 E604               ANI     00000100B  ; see if Tx Buffer is empty
0189 C8                 RZ              ; no, still busy
018A F6FF               ORI     0FFH    ; else, flag the ready condition
018C C9                 RET
                ;
                SIOBOUT:                ; SIO Channel B output routine
018D CD8001             CALL    SIOBOST ; see if port ready for output
0190 CA8D01             JZ      SIOBOUT ; try again if not
0193 79                 MOV     A,C     ; else, get the data for output
0194 D356               OUT     SIOBD   ; output it
0196 C9                 RET
                ;
                ; Centronics Printer Output Drivers
                ;
                PIA1ST:                 ; PIA1 Status Routine
0197 DB58               IN      PIA1AD  ; check for Printer status
0199 E617               ANI     00010111B ; isolate the bits of interest
                                        ;       Fault (bit 4)
                                        ;       Select (bit 2)
                                        ;       Paper Empty (bit 1)
                                        ;       Busy (bit 0)
019B EE14               XRI     00010100B  ; invert the -Fault and Select signals
019D CAA201             JZ      PIA1ST1 ;  all must be zero for ready condition
01A0 AF                 XRA     A       ;  else, show busy
01A1 C9                 RET
01A2 DB5B       PIA1ST1:  IN    PIA1BC  ; read the B Side Status Register
01A4 E680               ANI     10000000B  ; check if last byte was accepted
01A6 C8                 RZ              ; busy if zero
01A7 F6FF               ORI     0FFH    ;  show the ready condition
```

```
01A9 C9                 RET
                ;
                PIA1OUT:                    ; Printer Output entry point
01AA CD9701             CALL    PIA1ST  ; See if ready for data out
01AD CAAA01             JZ      PIA1OUT ;  try again if not
01B0 DB59               IN      PIA1BD  ; Reset the data accepted bit
01B2 79                 MOV     A,C     ; get the data for output
01B3 D359               OUT     PIA1BD  ; output it
01B5 C9                 RET
                ;
                PIA2ST:                     ; PIA2 Status Routine
01B6 DB5C               IN      PIA2AD  ; check for Printer status
01B8 E617               ANI     00010111B ; isolate the bits of interest
                                        ;       Fault (bit 4)
                                        ;       Select (bit 2)
                                        ;       Paper Empty (bit 1)
                                        ;       Busy (bit 0)
01BA EE14               XRI     00010100B  ; invert the -Fault and Select signals
                  JZ         510C01
~~01BC CAC101             JZ      PIA2ST1 ;  all must be zero for ready condition~~
01BF AF                 XRA     A       ;  else, show busy
01C0 C9                 RET
01C1 DB5F       PIA2ST1:  IN    PIA2BC  ; read the B Side Status Register
01C3 E680               ANI     10000000B  ; check if last byte was accepted
01C5 C8                 RZ              ; busy if zero
01C6 F6FF               ORI     0FFH    ;  show the ready condition
01C8 C9                 RET
                ;
                PIA2OUT:                    ; Printer Output entry point
01C9 CDB601             CALL    PIA2ST  ; See if ready for data out
01CC CAC901             JZ      PIA2OUT ;  try again if not
01CF DB5D               IN      PIA2BD  ; Reset the data accepted bit
01D1 79                 MOV     A,C     ; get the data for output
01D2 D35D               OUT     PIA2BD  ; output it
01D4 C9                 RET
                ;
```

B.2 OASIS DRIVER

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

                 2           COPY     CCS
0000            110          CCSSIOA  PORT=SIO4AD,CTC=CTC20,VECT=SIO4AV*2
0050            112+LINE:    EQU      80          ; line length
                113+PORTSIO4AD:       REL          ; relocatable
                114+
                115+
0000 C30F00     116+         JP       ST          ; get status
                117+
0003 C37000     118+         JP       IN          ; get byte
                119+
0006 C39200     120+         JP       OUT         ; put byte
                121+
0009 C3B400     122+         JP       INIT        ; initialize
                123+
000C C35901     124+         JP       UNIN        ; un-initialize
                125+
                126+
000F            127+ST:
                128+;
                129+; get SIO status
                130+;
000F F3         131+         DI                   ; no ints
0010 3A7A01     132+         LD       A,(BUFI)    ; get count
0013 B7         133+         OR       A           ; test if any
0014 F5         134+         PUSH     AF          ; save
0015 DB55       135+         IN       A,(DA+1)    ; get port status
0017 CB57       136+         BIT      2,A         ; test txrdy
0019 2852       137+         JR       Z,.NOTRDY   ; brif not ready
001B FD7E1C     138+         LD       A,(IY+28)   ; get enab type
001E CB47       139+         BIT      0,A         ; dtr
0020 2031       140+         JR       NZ,.ENAB1
0022 CB5F       141+         BIT      3,A         ; cts
0024 2039       142+         JR       NZ,.ENAB4
0026 CB4F       143+         BIT      1,A
0028 2021       144+         JR       NZ,.ENAB2   ; brif dc1/dc3
002A CB57       145+         BIT      2,A         ; test
002C 283B       146+         JR       Z,.RDY      ; brif not etx/ack
002E            147+.ENAB3:
002E F1         148+         POP      AF          ; get in flags
002F F5         149+         PUSH     AF          ; re-save
0030 2810       150+         JR       Z,.TEST3    ; brif no char rdy
0032 F1         151+         POP      AF          ; else, throw away
0033 CD7900     152+         CALL     IN1         ; get char
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

0036 E67F       153+           AND      7FH          ; mask
0038 FE06       154+           CP       ACK          ; test ack
003A 20D3       155+           JR       NZ,ST        ; brif not
003C FD361D00   156+           LD       (IY+29),0    ; store
0040 18CD       157+           JR       ST           ; go around
0042            158+.TEST3:
0042 FD7E1D     159+           LD       A,(IY+29)    ; get busy
0045 FE80       160+           CP       128          ; wait for ack?
0047 2020       161+           JR       NZ,.RDY      ; brif ready
0049 1822       162+           JR       .NOTRDY      ; else, busy
004B            163+.ENAB2:
004B FD7E1D     164+           LD       A,(IY+29)    ; get busy flag
004E B7         165+           OR       A            ; test
004F 201C       166+           JR       NZ,.NOTRDY   ; brif busy
0051 1816       167+           JR       .RDY
0053            168+.ENAB1:
0053 3E10       169+           LD       A,10H
0055 D355       170+           OUT      (DA+1),A     ; reset ext/status int
0057 DB55       171+           IN       A,(DA+1)     ; get reg 0
0059 CB5F       172+           BIT      3,A          ; test dtr
005B 2810       173+           JR       Z,.NOTRDY
005D 180A       174+           JR       .RDY
005F            175+.ENAB4:
005F 3E10       176+           LD       A,10H
0061 D355       177+           OUT      (DA+1),A     ; reset
0063 DB55       178+           IN       A,(DA+1)     ; get reg 0
0065 CB6F       179+           BIT      5,A          ; test cts
0067 2804       180+           JR       Z,.NOTRDY
0069            181+.RDY:
0069 F1         182+           POP      AF           ; get input status
006A 37         183+           SCF                   ; turn on cy
006B FB         184+           EI
006C C9         185+           RET                   ; return
006D            186+.NOTRDY:
006D F1         187+           POP      AF           ; get input status
006E FB         188+           EI
006F C9         189+           RET                   ; return
                190+
                191+
0070            192+IN:
                193+;
                194+; get byte from SIO
                195+;
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

0070 CD0F00     196+          CALL    ST            ; get status
0073 2004       197+          JR      NZ,IN1        ; yes, ready
0075 CF6B       198+          SC      107           ; else, wait for int
0077 18F7       199+          JR      IN            ; loop
0079            200+IN1:
0079 F3         201+          DI                    ; ints off
007A C5         202+          PUSH    BC            ; save regs
007B D5         203+          PUSH    DE
007C E5         204+          PUSH    HL
007D 217A01     205+          LD      HL,BUFI       ; point buffer
0080 35         206+          DEC     (HL)          ; decr length
0081 4E         207+          LD      C,(HL)        ; get length
0082 0600       208+          LD      B,0           ; zero msb
0084 23         209+          INC     HL            ; point first char
0085 7E         210+          LD      A,(HL)        ; load it
0086 2805       211+          JR      Z,.MT         ; brif buffer now empty
0088 54         212+          LD      D,H           ; copy register
0089 5D         213+          LD      E,L
008A 23         214+          INC     HL
008B EDB0       215+          LDIR                  ; compress the buffer
008D            216+.MT:
008D FB         217+          EI                    ; turn on ints
008E E1         218+          POP     HL            ; restore regs
008F D1         219+          POP     DE
0090 C1         220+          POP     BC
0091 C9         221+          RET                   ; return
                222+
                223+
0092            224+OUT:
                225+;
                226+; put byte to device
                227+;
0092 CD0F00     228+          CALL    ST            ; get status
0095 30FB       229+          JR      NC,OUT        ; loop till ready
0097 F3         230+          DI                    ; ints off
0098 FDCB1C56   231+          BIT     2,(IY+28)     ; enab 3
009C 2811       232+          JR      Z,OUT2        ; no
009E FD341D     233+          INC     (IY+29)       ; bump count
00A1 FD7E1D     234+          LD      A,(IY+29)     ; load count
00A4 FE80       235+          CP      128           ; full now?
00A6 2007       236+          JR      NZ,OUT2       ; no
00A8 3E03       237+          LD      A,ETX         ; else, send etx
00AA FB         238+          EI                    ; turn on ints
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

00AB D354        239+          OUT      (DA),A
00AD 18E3        240+          JR       OUT          ; loop
00AF             241+OUT2:
00AF 79          242+          LD       A,C          ; get char
00B0 FB          243+          EI                    ; turn on ints
00B1 D354        244+          OUT      (DA),A       ; write
00B3 C9          245+          RET                   ; return
                 246+
00B4             247+INIT:
00B4 FD22DE01    248+          LD       (UCB),IY     ; save ucb addr
00B8 FDCB08A6    249+          RES      4,(IY+8)     ; no sync mode
00BC FDCB0886    250+          RES      0,(IY+8)     ; or sdlc
00C0 FD7E05      251+          LD       A,(IY+5)     ; get baud rate
00C3 E60F        252+          AND      0FH          ; any
00C5 200A        253+          JR       NZ,.SOMEB    ; brif some
00C7 FD7E05      254+          LD       A,(IY+5)     ; get prev
00CA F60B        255+          OR       11           ; default to 9600
00CC FD7705      256+          LD       (IY+5),A     ; store
00CF E60F        257+          AND      0FH          ; mask
00D1             258+.SOMEB:
                 259+;         CP       12           ; too big?
                 260+;         JR       C,.OKB       ; brif ok
                 261+;         LD       A,(IY+5)     ; else, get enab
                 262+;         AND      0F0H         ; mask
                 263+;         OR       11           ; merge 9600
                 264+;         LD       (IY+5),A
                 265+;         AND      0FH          ; mask
00D1             266+.OKB:
00D1 3D          267+          DEC      A            ; less one
00D2 87          268+          ADD      A            ; times two
00D3 5F          269+          LD       E,A
00D4 1600        270+          LD       D,0          ; zero high
00D6 21E001      271+          LD       HL,BAUD      ; point table
00D9 19          272+          ADD      HL,DE        ; offset
00DA 0E50        273+          LD       C,CTC
00DC 0602        274+          LD       B,2          ; two bytes
00DE EDB3        275+          OTIR                  ; program it
00E0 3E3C        276+          LD       A,SIO4AV*2/2 ; vector number
00E2 114102      277+          LD       DE,RETI      ; dummy
00E5 CF67        278+          SC       103
00E7 3C          279+          INC      A
00E8 CF67        280+          SC       103
00EA 11FE01      281+          LD       DE,INTI      ; input interrupt
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

00ED 3C         282+         INC     A
00EE CF67       283+         SC      103
00F0 3C         284+         INC     A
00F1 CF67       285+         SC      103
00F3 F3         286+         DI                      ; turn off ints
00F4 3E02       287+         LD      A,2             ; reg 2
00F6 D357       288+         OUT     (PORTB),A
00F8 3E70       289+         LD      A,SIO4AV*2.AND.0F0H ; int vector
00FA D357       290+         OUT     (PORTB),A
00FC 3E04       291+         LD      A,4             ; wr 4
00FE D355       292+         OUT     (DA+1),A
0100 FDCB087E   293+         BIT     7,(IY+8)        ; parity enable?
0104 280C       294+         JR      Z,.NOPAR        ; brif none
0106 FDCB0876   295+         BIT     6,(IY+8)        ; test even/odd
010A 3E4D       296+         LD      A,01001101B ; even
010C 2006       297+         JR      NZ,.OUT
010E 3E4F       298+         LD      A,01001111B ; odd
0110 1802       299+         JR      .OUT
0112            300+.NOPAR:
0112 3E4C       301+         LD      A,01001100B ; noparity
0114            302+.OUT:
0114 D355       303+         OUT     (DA+1),A
0116 3E03       304+         LD      A,3             ; wr 3 (rcv logic)
0118 D355       305+         OUT     (DA+1),A
011A FDCB087E   306+         BIT     7,(IY+8)        ; parity?
011E 3EC1       307+         LD      A,11000001B ; default
0120 2802       308+         JR      Z,.NP           ; brif ok
0122 3E41       309+         LD      A,01000001B ; else, 7 bits
0124            310+.NP:
0124 D355       311+         OUT     (DA+1),A
0126 3E01       312+         LD      A,1             ; wr 1 (control)
0128 D355       313+         OUT     (DA+1),A
012A 3E18       314+         LD      A,00011000B ; int mask
012C D355       315+         OUT     (DA+1),A
012E 3E01       316+         LD      A,1
0130 D357       317+         OUT     (PORTB),A
0132 3E1C       318+         LD      A,00011100B
0134 D357       319+         OUT     (PORTB),A
0136 3E05       320+         LD      A,5             ; wr 5 (trns)
0138 D355       321+         OUT     (DA+1),A
013A FDCB087E   322+         BIT     7,(IY+8)        ; test parity
013E 3EEA       323+         LD      A,11101010B ; default
0140 2802       324+         JR      Z,.NTP          ; brif ok
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

0142 3EAA       325+          LD        A,10101010B ; else parity = 7 bits
0144            326+.NTP:
0144 D355       327+          OUT       (DA+1),A
0146 E5         328+          PUSH      HL
0147 21CE01     329+          LD        HL,TBUF
014A 22CC01     330+          LD        (TBUFP),HL   ; set up the overflow buffer
014D 3EFF       331+          LD        A,0FFH
014F 32CB01     332+          LD        (OVFL),A
0152 E1         333+          POP       HL
0153 FB         334+          EI                     ; turn on ints
0154 AF         335+          XOR       A            ; clear busy
0155 FD771D     336+          LD        (IY+29),A    ; store
0158 C9         337+          RET                    ; return
                338+
0159            339+UNIN:
0159 F3         340+          DI                     ; turn off ints
015A 015506     341+          LD        BC,6*256+DA+1
015D 217401     342+          LD        HL,UCMD      ; reset interrupts
0160 EDB3       343+          OTIR
0162 3E3C       344+          LD        A,SIO4AV*2/2 ; clear the vectors
0164 110000     345+          LD        DE,0
0167 CF67       346+          SC        103
0169 3C         347+          INC       A
016A CF67       348+          SC        103
016C 3C         349+          INC       A
016D CF67       350+          SC        103
016F 3C         351+          INC       A
0170 CF67       352+          SC        103
0172 FB         353+          EI                     ; turn on ints
0173 C9         354+          RET                    ; return
                355+
0174 01040300   356+UCMD:     DC        1,4,3,0,5,0
                357+
017A 00         358+BUFI:     DC        0            ; buffer length
017B            359+          DS        LINE         ; the buffer itself
01CB            360+OVFL:     DS        1            ; overflow byte count
01CC            361+TBUFP:    DS        2            ; pointer into overflow buffe
01CE            362+TBUF:     DS        16           ; the overflow buffer itself
                363+
01DE            364+UCB:      DS        2
0054            365+DA:       EQU       SIO4AD       ; port address
0057            366+PORTB:    EQU       [DA.AND.0FCH]+3 ; cmd port b
0050            367+CTC:      EQU       CTC20
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

0011           368+DC1:      EQU     11H
0013           369+DC3:      EQU     13H
0003           370+ETX:      EQU     03H
0006           371+ACK:      EQU     06H
               372+
01E0           373+BAUD:
01E0 07D0      374+          DC      7,208        ; 75 baud
01E2 078E      375+          DC      7,142        ; 110
01E4 0774      376+          DC      7,116        ; 134.5
01E6 0768      377+          DC      7,104        ; 150
01E8 0734      378+          DC      7,52         ; 300
01EA 47C0      379+          DC      47H,192      ; 600
01EC 4760      380+          DC      47H,96       ; 1200
01EE 4730      381+          DC      47H,48       ; 2400
01F0 4718      382+          DC      47H,24       ; 4800
01F2 4710      383+          DC      47H,16       ; 7200
01F4 470C      384+          DC      47H,12       ; 9600
01F6 4706      385+          DC      47H,6        ; 19200
01F8 4703      386+          DC      47H,3        ; 38400
01FA 4702      387+          DC      47H,2        ; 57600
01FC 4701      388+          DC      47H,1        ; 115,200
               389+
01FE           390+INTI:
               391+;
               392+; service receiver interrupt
               393+;
01FE FB        394+          EI                   ; allow nested interrupts
01FF F5        395+          PUSH    AF           ; save reg a,f
0200 FDE5      396+          PUSH    IY           ; save iy
0202 FD2ADE01  397+          LD      IY,(UCB)     ; point to ucb
0206 C5        398+          PUSH    BC           ; save b,c
0207 3E01      399+          LD      A,1          ; read reg 1
0209 D355      400+          OUT     (DA+1),A
020B DB55      401+          IN      A,(DA+1)     ; get second status
020D 47        402+          LD      B,A          ; save it
020E DB54      403+          IN      A,(DA)       ; get char
0210 FDCB086E  404+          BIT     5,(IY+8)     ; 8 bit code?
0214 2002      405+          JR      NZ,.EIGHT    ; yes
0216 CBBF      406+          RES     7,A          ; turn off parity
0218           407+.EIGHT:
0218 4F        408+          LD      C,A          ; save char
               409+;
               410+; test parity
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

                411+;
0219 CB60       412+        BIT     4,B          ; pe?
021B 2806       413+        JR      Z,.NOPE
021D 0E3F       414+        LD      C,'?'        ; replace char
021F 3E30       415+        LD      A,30H
0221 D355       416+        OUT     (DA+1),A     ; reset parity error
0223            417+.NOPE:
0223 3ACB01     418+        LD      A,(OVFL)     ; check for nested overflow
0226 3C         419+        INC     A
0227 281E       420+        JR      Z,.NOPE1     ; brif no overflow
0229 FA4602     421+        JP      M,.NOPE2     ; clear any negative value
022C FE10       422+        CP      16           ; insure this buffer not full
022E 300D       423+        JR      NC,.IGNOR    ; brif full (ignore character
0230 32CB01     424+        LD      (OVFL),A     ; update the overflow count
0233 E5         425+        PUSH    HL           ; put character into overflow
0234 2ACC01     426+        LD      HL,(TBUFP)   ; get the pointer
0237 71         427+        LD      (HL),C       ; save the character
0238 23         428+        INC     HL           ; advance the pointer
0239 22CC01     429+        LD      (TBUFP),HL   ; save the pointer
023C E1         430+        POP     HL           ; restore the registers
023D            431+.IGNOR:
023D C1         432+        POP     BC
023E FDE1       433+        POP     IY
0240 F1         434+        POP     AF
0241            435+RETI:
0241 FB         436+        EI                   ; re-allow the interrupts
0242 00         437+        NOP
0243 00         438+        NOP
0244 ED4D       439+        RETI
                440+
0246            441+.NOPE2:
0246 AF         442+        XOR     A            ; set to zero if negative
0247            443+.NOPE1:
0247 32CB01     444+        LD      (OVFL),A     ; update the overflow count
024A CD4102     445+        CALL    RETI         ; reset the interrupts
024D            446+REPT:
024D F3         447+        DI                   ; disable interrupts for dura
024E FDCB057E   448+        BIT     7,(IY+5)     ; console
0252 282B       449+        JR      Z,.NOTR
0254 79         450+        LD      A,C          ; get char
0255 FDCB0576   451+        BIT     6,(IY+5)     ; is an ESC in progress?
0259 281C       452+        JR      Z,.CONVRT    ; brif not
025B CBAF       453+        RES     5,A          ; convert to upper case
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

025D FE51       454+          CP      'Q'          ; is it ESC Q?
025F 2808       455+          JR      Z,.RSTFIFO   ; brif so
0261 FE44       456+          CP      'D'          ; is it ESC D?
0263 2804       457+          JR      Z,.RSTFIFO   ; brif so
0265 FE53       458+          CP      'S'          ; is it ESC S?
0267 200D       459+          JR      NZ,.CONVRT1  ; brif not
0269            460+.RSTFIFO:
0269 3EFF       461+          LD      A,0FFH       ; reset the FIFO pointers
026B 32CB01     462+          LD      (OVFL),A
026E E5         463+          PUSH    HL           ; save hl for moment
026F 21CE01     464+          LD      HL,TBUF
0272 22CC01     465+          LD      (TBUFP),HL
0275 E1         466+          POP     HL           ; restore hl
0276            467+.CONVRT1:
0276 79         468+          LD      A,C          ; reget the character
0277            469+.CONVRT:
0277 CF66       470+          SC      102          ; translate
0279 3003       471+          JR      NC,.OK       ; brif no ignore
027B C1         472+          POP     BC           ; else, adjust
027C 184C       473+          JR      SIORET       ; return
027E            474+.OK:
027E 4F         475+          LD      C,A          ; else, repl
027F            476+.NOTR:
027F 3A7A01     477+          LD      A,(BUFI)     ; get prev count
0282 FE50       478+          CP      LINE         ; test full
0284 2011       479+          JR      NZ,ROC       ; brif not
0286 C1         480+          POP     BC           ; adjust
0287 FDCB057E   481+          BIT     7,(IY+5)     ; conin?
028B 283D       482+          JR      Z,SIORET     ; no, return
028D E5         483+          PUSH    HL           ; save h,l
028E 21C202     484+          LD      HL,R10       ; get jmp addr
0291 E5         485+          PUSH    HL           ; put on stack
0292 F5         486+          PUSH    AF           ; save reg af
0293 FDE5       487+          PUSH    IY           ; save iy
0295 1833       488+          JR      SIORET       ; clear interrupt
0297            489+ROC:
0297 FDCB1C4E   490+          BIT     1,(IY+28)    ; enab 2?
029B 2814       491+          JR      Z,.NOENAB    ; not
029D 79         492+          LD      A,C
029E E67F       493+          AND     7FH
02A0 FE11       494+          CP      DC1
02A2 280A       495+          JR      Z,.CTLQ
02A4 FE13       496+          CP      DC3
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

02A6 2009       497+          JR        NZ,.NOENAB
02A8            498+.CTLS:
02A8 C1         499+          POP       BC
02A9 FD771D     500+          LD        (IY+29),A     ; set the busy sw
02AC 181C       501+          JR        SIORET
02AE            502+.CTLQ:
02AE AF         503+          XOR       A             ; reset
02AF 18F7       504+          JR        .CTLS         ; turn off busy sw
02B1            505+.NOENAB:
02B1 79         506+          LD        A,C           ; get this char
02B2 C1         507+          POP       BC            ; adjust stack
02B3            508+R2:
02B3 D5         509+          PUSH      DE            ; save de and hl regs
02B4 E5         510+          PUSH      HL
02B5 217A01     511+          LD        HL,BUFI       ; point buffer
02B8 34         512+          INC       (HL)          ; incr count
02B9 5E         513+          LD        E,(HL)        ; load it
02BA 1600       514+          LD        D,0           ; zero high
02BC 19         515+          ADD       HL,DE         ; point next
02BD 77         516+          LD        (HL),A        ; store the character
02BE E1         517+          POP       HL            ; restore regs
02BF D1         518+          POP       DE
02C0 1808       519+          JR        SIORET        ; return
02C2            520+R10:
02C2 C5         521+          PUSH      BC            ; save b,c
02C3 0E07       522+          LD        C,7           ; get bell code
02C5 CD9200     523+          CALL      OUT           ; write the bell
02C8 C1         524+          POP       BC            ; restore b,c
02C9 E1         525+          POP       HL            ; restore h,l
02CA            526+SIORET:
02CA 3ACB01     527+          LD        A,(OVFL)      ; see if overflow occurred
02CD 3D         528+          DEC       A
02CE 32CB01     529+          LD        (OVFL),A      ; update the character count
02D1 FAF002     530+          JP        M,SIORET1     ; brif no overflow
02D4 C5         531+          PUSH      BC            ; else, empty the overflow
02D5 D5         532+          PUSH      DE
02D6 E5         533+          PUSH      HL
02D7 21CF01     534+          LD        HL,TBUF+1     ; squeeze the buffer down
02DA 11CE01     535+          LD        DE,TBUF
02DD 010F00     536+          LD        BC,15         ; byte count
02E0 1A         537+          LD        A,(DE)        ; get the next character
02E1 EDB0       538+          LDIR                    ; do the squeeze
02E3 2ACC01     539+          LD        HL,(TBUFP)    ; update the pointer
```

DEV25: CCS 2719 SIO Port A Driver

```
Addr Obj-Code  Line *** Source Statement ***

02E6 2B         540+          DEC      HL
02E7 22CC01     541+          LD       (TBUFP),HL
02EA 4F         542+          LD       C,A          ; move character over
02EB E1         543+          POP      HL
02EC D1         544+          POP      DE
02ED C34D02     545+          JP       REPT         ; go process this character
                546+
02F0            547+SIORET1:
02F0 FB         548+          EI
02F1 FDE1       549+          POP      IY           ; restore iy
02F3 F1         550+          POP      AF           ; restore a,f
02F4 C9         551+          RET                   ; return from interrupt
                552+
02F5            554           END
```

No assembly errors.

DEV14: CCS 2719 PIA1 Centronics Printer Driver

```
Addr Obj-Code  Line *** Source Statement ***

                 2            COPY     CCS
0000             110          CCSPIA   PORT=PIA1AD,CTC=CTC22,VECT=CTC22V
                 112+PORTPIA1AD:       REL           ; relocatable
0000 C30F00      113+         JP       ST            ; get status
0003 C31700      114+         JP       IN            ; get byte
0006 C31B00      115+         JP       OUT           ; put byte
0009 C33300      116+         JP       INIT          ; initialize
000C C35E00      117+         JP       UNINIT        ; un-initialize
                 118+;
                 119+;
000F             120+ST:
                 121+;
                 122+; get the PIA status
                 123+;
000F DB58        124+         IN       A,(PIA1AD)    ; read the printer status
0011 E617        125+         AND      00010111B     ; isolate the bits of interes
0013 EE14        126+         XOR      00010100B     ; invert the positive logic b
0015 2802        127+         JR       Z,.OK         ; brif if all normal
0017             128+IN:                             ; ignore input requests
0017 AF          129+         XOR      A             ; else, show busy
0018 C9          130+         RET
                 131+;
0019             132+.OK
0019 37          133+         SCF                    ; show ready
001A C9          134+         RET
                 135+;
001B             136+OUT:
001B F5          137+         PUSH     AF            ; save the data output byte
001C             138+OUT1:
001C DB58        139+         IN       A,(PIA1AD)    ; read the printer status bit
001E E617        140+         AND      00010111B     ; isolate the bits of interes
0020 EE14        141+         XOR      00010100B     ; set positive logic bits neg
0022 2006        142+         JR       NZ,.WAIT      ; wait if not ready
0024 DB5B        143+         IN       A,(PIA1AD+3)  ; see if last byte was accep
0026 CB7F        144+         BIT      7,A
0028 2804        145+         JR       Z,.DOIT       ; brif if idle
002A             146+.WAIT:
002A CF6B        147+         SC       107           ; else wait for interrupt
002C 18EE        148+         JR       OUT1          ; try again
                 149+;
002E             150+.DOIT:
002E 79          151+         LD       A,C           ; get the byte for output
002F D359        152+         OUT      (PIA1AD+1),A  ; output the next byte
```

DEV14: CCS 2719 PIA1 Centronics Printer Driver

```
Addr Obj-Code  Line *** Source Statement ***

0031 F1         153+         POP     AF          ; restore accumulator
0032 C9         154+         RET
                155+;
                156+; Initialize the PIA for action
                157+;
0033            158+INIT:
0033 F3         159+         DI                  ; no interrupts for duration
0034 3E06       160+         LD      A,CTC22V    ; set up the vector
0036 117000     161+         LD      DE,PIAINT   ; point to ISR
0039 CF67       162+         SC      103         ; log in the interrupt
003B 3E04       163+         LD      A,CTC22V.AND.OFCH ; set the ctc
003D B7         164+         OR      A           ; insure the carry is clear
003E 07         165+         RLCA                ; convert to vector
003F D350       166+         OUT     (CTC22.AND.OFCH),A
0041 AF         167+         XOR     A           ; get a zero
0042 D35B       168+         OUT     (PIA1AD+3),A ; enable data direction reg
0044 D35A       169+         OUT     (PIA1AD+2),A
0046 D358       170+         OUT     (PIA1AD),A  ; A side is input
0048 3D         171+         DEC     A
0049 D359       172+         OUT     (PIA1AD+1),A ; B side is output
004B 3E2C       173+         LD      A,00101100B ; set the PIA operating condi
004D D35A       174+         OUT     (PIA1AD+2),A ; no interrupts on A side
004F 3C         175+         INC     A
0050 D35B       176+         OUT     (PIA1AD+3),A ; interrupts from B side
0052 3ED7       177+         LD      A,11010111B ; CTC triggers on positive ed
0054 D352       178+         OUT     (CTC22),A
0056 3E01       179+         LD      A,1         ; use as IM2 generator
0058 D352       180+         OUT     (CTC22),A
005A FB         181+         EI                  ; re-enable interrupts
005B DB59       182+         IN      A,(PIA1AD+1) ; prime the PIA
005D C9         183+         RET
                184+;
                185+; Un-initialize the PIA and the CTC
                186+;
                187+
005E            188+UNINIT:
005E F3         189+         DI                  ; disable interrupts
005F 3E2C       190+         LD      A,00101100B ; disable PIA interrupts
0061 D35B       191+         OUT     (PIA1AD+3),A
0063 3E03       192+         LD      A,3         ; disable the CTC
0065 D352       193+         OUT     (CTC22),A
0067 3E06       194+         LD      A,CTC22V    ; log out the vector
0069 110000     195+         LD      DE,0
```

DEV14: CCS 2719 PIA1 Centronics Printer Driver

```
Addr Obj-Code  Line *** Source Statement ***

006C CF67       196+          SC        103          ; unschedule
006E FB         197+          EI
006F C9         198+          RET
                199+;
                200+; Interrupt service routine
                201+;
                202+
0070            203+PIAINT:
0070 FB         204+          EI                     ; allow nested interrupts
0071 F5         205+          PUSH      AF           ; save the accumulator
0072 DB59       206+          IN        A,(PIA1AD+1) ; clear the interrupt flag
0074 F1         207+          POP       AF           ; restore the accumulator
0075 ED4D       208+          RETI                   ; done
0077            209           END

No assembly errors.
```