Hardware Reference Manual

Quad Serial Interface

Central Data Corporation

713 Edgebrook Drive, Champaign, IL 61820 Phone: (217) 359-8010

Quad Serial Interface Manual

Central Data Corporation
P.O. Box 2530, Station A
Champaign, Il 61820
Phone: (217) 359-8010

TABLE OF CONTENTS

1. General Information

The Central Data Quad Serial Interface board is designed to expand the serial I/O capacity of any Multibus* system. The board uses the industry standard 8251 USART (Universal Synchronous/Asynchronous Receiver/Transmitter) as the parallel to serial converter, and allows three independent baud rates (transmission speeds) to be set for the board's four channels.

The board supports standard EIA RS-232 interfaces, with the following pins used: TxD, RxD, DSR, CTS, DTR, and RTS. The board is capable of operating at baud rates ranging from 75 to 19,200 baud, and can be interrupt driven if so desired. Also, the board supports 16-bit I/O addressing as called for in the Multibus specification, with an option to use only 8-bit I/O addresses.

The board drives both the XACK and AACK lines of the Multibus to allow for the greatest flexibility. It can return either signal from 0-800ns after the receipt of a command, in 100ns increments. It is suggested that XACK be strapped to be equal to the access time of the board, while AACK can be strapped to allow the fastest possible system operation.

* Multibus is a trademark of Intel Corporation and is used throughout this manual.

2. Functional Description

The Quad Serial Interface board is divided into several major sections, which are described briefly below. For more detailed information, refer to the Principles of Operation section of this manual.

The bus interface of the board consists of the I/O address comparator, data bus buffers, interrupt control cicruitry, and XACK/AACK generation logic. The board requires 16 I/O ports, which can be located on any 16-port boundary. Dip-switch I/O addressing allows the user to select the address of the board using either an 8- or 16-bit I/O address. With this ability, the board can work equally well in systems which generate a full 16-bit I/O address as well as in older 8-bit systems. The data bus is buffered into and out of the board, and interrupts can be generated on the occurance of any "receiver full" or "transmitter empty" condition. Finally, the XACK/AACK generation circuitry acknowledges all commands to the board and allows the system to run at the maximum possible speed.

The timer section of the board consists of a crystal oscillator used to generate a time-base on which all baud rates are based. This clock is used to drive a programmable divider circuit which has three independent outputs. This divider can be setup by the processor to give output frequencies corresponding to transmission rates of 75 to 19,200 baud.

Finally, the USART section is the actual interface to the external devices. This section is repeated on the board four times, which gives four totally independent channels. The interface between the on-board circuitry and the external connector is made through industry standard drivers and receivers, which guarantee proper RS-232 specifications. Options are provided at each port to select if interrupts will be generated on its "receiver full" and "transmitter empty" conditions.

3. Principles of Operation

This chapter details the operation of the entire interface board. Any signal names in this text followed by a slash (/) indicate that the signal is active-low.

As in all Central Data schematics, a grid system is provided to help locate sources and destinations of signals. The source of any named signal will have references to all locations on the schematics where the signal is used. At each location where a signal is used a reference is given to where it was generated.

If the location is on the same sheet as it is being referenced, it will show only a grid location (i.e. D2). If, however, the referenced signal appears on a seperate page, it will have the grid location preceded by the sheet number (i.e. 2-B5).

Bus Interface

Sheet 1 of the schematics shows most of the Multibus interfacing logic. Four functions must be provided to interface to the bus: address decoding, signal buffering, interrupt generation, and XACK/AACK generation.

The board requires the use of 16 I/O ports on the bus. These ports can be started on any 16-port boundary, using either 8- or 16-bit addresses. The address decoding circuitry for the upper address lines (A8-A15) is found on sheet 2 of the schematics.

All of the address lines from the Multibus are buffered through 74LS04 gates. The buffered address lines are then routed to chip selection circuitry (for A0-A3) and to address decoding circuitry (A4-A15).

The address decoding circuit consists of twelve 74LS266 open collector exclusive-NOR gates. All of the outputs of the gates are tied together, allowing any of the gates to pull the output low if its inputs do not match. If all of the pairs of inputs match, the common output is pulled high by a resistor to +5V.

One input from each of the gates goes to a buffered address

line, with the other going to a dip-switch. This dip-switch, when closed, causes the corresponding gate input to become grounded. Under this circumstance, the address line leading to the same gate must also be low for the board to be addressed. If the switch position is left open, the input to the gate goes to a high state, thus comparing for a high address line.

To allow the selection between 8- and 16-bit I/O addressing, the outputs of the gates related to A8-A15 are connected through a shorting plug to the other four outputs. If the shorting plug is installed, then the board decodes the full 16-bit address bus. If the shorting plug is removed, then the upper eight gates will not drive the common output, and thus only the lower four lines (A4-A7) are used for addressing.

When the address comparator is equal, pin 6 of IC19 will go high. This line is used in conjunction with pin 5 to enable the 74LS138 decoder. Pin 5 is low whenever the first eight addresses of the board are being accessed; when these ports are being used, one of the four USARTs is enabled. The other eight addresses for the board are used by the programmable timer circuit (IC24).

When the 74LS138 is enabled, it uses the next lower two address lines (A2 and A1) to select which USART should be enabled. In this manner, each USART has two consecutive addresses, for both reading and writing. The output of the decoder drives the chip-select pins of the USARTs. The RD/ and WR/ pins of the USARTs cause a transfer to take place only when the chip select pin is low.

When the other eight addresses are being selected, it will cause pin 10 of IC18 to go high. Pin 11 is the BOARD ADDRESSED line, as was used by the 74LS138. The output of this gate is used to drive the chip-select pin of the 8253 timer. Since this timer only requires four I/O addresses, and eight are available, pin 9 of IC18 is used to enable the chip during the lower four addresses only.

In summary, the board uses 16 of the I/O ports on the system. The base address for these ports is selected with dip-switches, and the 16 ports are divided internally for the following devices:

Port Offset	Device Selected
0- 1	USART 0
2- 3	USART 1
4- 5	USART 2
6- 7	USART 3

8-11	8253 TIMER
12-15	NOT USED

The two I/O command lines (IORC/ and IOWC/) are also buffered from the Multibus, and run to the I/O chips on the board. The INIT/ signal is buffered and used to initialize the USARTs to a known state when power is turned on or when the system is reset.

The data bus buffers consist of two 74LS242s, each one buffering four data lines. These are inverting buffers, thus immediately correcting for the inverted data on the bus. Since the directional enable pins of the buffers are of opposite polarity, they can be tied together, and are driven by the DBIN signal. This signal goes high whenever the board is addressed and a input command is in progress. During all other conditions, DBIN is low, thus sending data from the Multibus into the board.

The interrupt circuitry takes the two outputs available from each USART (explained later) and gates them together to form a common interrupt signal to the processor. If any of the SIOT or SIOR lines goes high, it will cause pin 6 of IC23 to go low. If the board is connected to a Multibus interrupt line, this will cause a vectored interrupt.

The board generates two command acknowledge signals. The first, XACK, indicates when a data transfer is complete and the processor can go to the next cycle. The other, AACK, gives the processor advance information concerning when a transfer will be complete.

The circuit which generates the acknowledge signals consists of a shift register (74LS164, IC13) which is kept cleared when the board is not active. When an I/O command occurs, the clear input goes high, allowing the register to shift 1's through at a 5MHz rate. The eight outputs of the shift register, which go high from 200-1600ns after the time a command starts, can be jumpered to the XACK and AACK drivers (IC83). Note that since the command is asynchronous with respect to the clock the outputs may vary up to one clock cycle (i.e. the second output can occur anywhere from 200-400ns after command initiation).

The user can also select either acknowledge signal to be returned as soon as the board is selected by tying the driver's input high. The drivers are enabled whenever a command is occuring to this board, thus gating the proper timing onto the bus.

Timer Section

Sheet 2 of the schematics contains the crystal oscillator and the programmable timer circuits. The oscillator is a simple feedback network, with the resistors used to bias the 7404 gates into their linear region, and the 100pf capacitor used to block any DC voltage to the crystal, and to stabilize operation. After buffering, this 4.9152MHz signal is sent to the XACK/AACK generating circuitry as well as a 74LS163 divider (IC27). This divider provides a clock for the internal operation of the 8251s (not tied to the baud rate) on pin 13, which is 1.2288MHz.

The divider also drives the programmable timer circuit (IC24) with a 614.4KHz signal, which is 32x the maximum baud rate for the board. Since the minimum divisor for the programmable timer is 2, and the USARTs normally run in 16x baud rate mode, this limits the maximum baud rate of the board to 19,200. Note that this baud rate can be exceeded (assuming the USART specifications are kept) if the USART is run in 1x baud rate mode.

The 8253 timer is used to generate the three independent baud rate clocks which are used by the USARTs. The divider will only be operated in mode 3, to generate square wave outputs. The Intel 8253 data sheet should be consulted for complete programming details on the timer.

The timer has three outputs, labeled OUT0, OUT1, and OUT2. The first output is run to the first two USARTs. The other two outputs drive the third and fourth USARTs, respectively. Since the first two USARTs (marked port 0 and port 1 on the board) have their baud rate clocks tied together, they cannot have different baud rates. They can be independent of the other two ports, which are both independent of the others.

The programming of the timer is very simple, with the following procedure followed:

1) Write port (base+11) with 36, 76, or B6, to prepare the timer to receive the divisor for output 0, 1, or 2, respectively.

2) Write the least significant byte of the divisor to the timer, followed by the most significant byte. The port that should be written with these bytes is (base+8) for output 0, (base+9) for output 1, and (base+10) for output 2.

Note that the base address referenced above refers to the address selected by the dip-switches on the board. Also, the divisor is simply the number that the input frequency (614.4KHz) must be divided by in order to obtain the proper transmit and receive clocks on the USARTs. For example, if

the USART is run in 16x baud rate mode, the proper divisor for 9600 baud would be 4, and for 150 baud it would be 256 (H100).

USART Section

Sheet 3 of the schematics shows the actual interface to the external devices. Note that this sheet is repeated four times on the board, with the IC numbers listed for ports 0-3, in that order. Also, the signals CLKX, SIOTX, and SIORX are referenced with the numbers 0-3 instead of the trailing "X" to indicate which USART is being used.

The format of the characters being sent and received is determined entirely by the USART and how it is programmed. Details on the programming of the 8251 are provided in the AMD 8251 data sheet. In addition, example software drivers are printed in this manual to guide the user in how an interrupt-based driver could be written.

Note that there are several strap selections available for each USART. The first, the CTS selection, is required because the USART will not transmit any characters unless its CTS/ pin is low. Since many serial devices do not drive this line, the strap labeled CTSINT allows the user to drive it from the RTS/ output of the USART. With this arrangement, whenever the RTS/ signal from a USART is low, it will be allowed to transmit. In the other mode, with the CTSEXT strap in place, the external device must drive CTS in order for the board to operate properly.

Secondly, there are various selection options for interrupt generation. Each USART drives two interrupt lines, one for transmitter interrupts, and one for receiver interrupts. The first, SIOTX, can be driven by the TxRDY or TxEMPTY pins or the USART or tied to ground. When tied to ground, an interrupt will never be generated for that USART's transmitter section. Normally, in interrupt driven mode, the TxRDY position is strapped. The other interrupt line, SIORX, can be tied to the 8251's RxRDY output or disabled by tying it to ground.

Note that one strap must be inserted for each of the lines, or the corresponding line will float high, always causing an interrupt.

All of the RS-232 signals from the external connector are buffered by 1489s. Note that a capacitor can be added to slow the rise and fall times on the RxD input of the USART. Also, the USART outputs to the connector are buffered through 1488s, with each output optionally tied to a capacitor to slow rise and fall times. Normally, these

additional capacitors are not needed.

4. Installation/User Selectable Options

The Quad Serial Interface is designed to operate in any standard Multibus system. The board can occupy any card position of the system, since it does not operate as a bus master.

Addressing

The board has a 12-position dip-switch to select the port addresses it will respond to. Each position of the switch corresponds to one address line, from A4 to A15. As marked on the board, A15 is selected by the left-most switch, while A4 is selected by the right-most. An address line is compared for "0" if the switch is closed (up), as printed on the board. With the switch left open (down), the corresponding address line is compared for "1".

If 16-bit I/O addressing is to be used, a shorting plug must be placed over the two wire-wrap pins marked EXTENDED I/O. For systems where only 8-bit I/O addressing is used, this shorting plug should be left off. Also, for 8-bit systems, the upper eight address switches are not used.

CTS Selection

Since the USART will not transmit any data unless the CTS signal is active, the board allows the user to jumper it to a known state. This option can be used when the board is being connected to a simple device which does not generate this signal.

When the user wants the USART's RTS output to drive its CTS input, then a shorting plug should be placed in the USART's CTSINT position. This will allow the USART to transmit regardless of the state of the CTS signal from the external connector. If the user wishes CTS to be monitored from the device, then the CTSEXT position should be shorted. This will cause the output of the CTS buffer from the external connector to be run to the USART's CTS input.

Interrupt Selections

Each USART can generate a vectored interrupt when its receiver or transmitter requires service. The signal RxRDY

from the USART can be used to generate a receiver interrupt, while either TxRDY or TxEMPTY can be used to cause a transmitter interrupt. Also, the user can select that transmitter interrupts or reciever interrupts will be disabled from any particular USART.

This selection is made using the wire-wrap pins to the right of each USART. To select the proper transmitter interrupt mode, a shorting plug must be placed in one of the following positions:

Position	Interrupt Source
TXRDY	TxRDY generates an interrupt
TXMT	TxEMPTY generates an interrupt
TXOFF	Transmitter interrupts disabled

The receiver interrupt mode is selected by placing a shorting plug in one of the following two positions:

Position	Interrupt Mode
RXRDY	RxRDY generates an interrupt
RXOFF	Receiver interrupts disabled

Note that a shorting plug must be inserted for each USART interrupt source, or the corresponding interrupt line will always be active.

Once the interrupt modes for each USART are determined, a vectored interrupt level must be established. The user can pick any level (0-7) to receive the interrupt by placing a shorting plug on the appropriately marked pins on the board.

XACK and AACK Generation

In order for the board to acknowledge processor commands, two lines are provided to indicate when a data transfer is complete. The XACK (transfer acknoweledge) line is driven by the board when the transfer is completely finished, and the processor is allowed to complete the cycle. The AACK (advanced acknowledge) is provided to allow systems to operate at their full speed potential (by preventing wait states), since it can be returned before XACK. Only XACK is used to indicate when a cycle can end, with the function of AACK to give advance information concerning the timing of the board.

Both of the lines can be strap selectable to return to the processor from 0-1600ns after a command is received, in 200ns increments. The selection of timing for each line is done with shorting plugs placed over wire-wrap pins on the

board.

The board has two rows of wire-wrap pins which are used for XACK/AACK generation. The top row is used for AACK, while the bottom row is for XACK. Each row consists of 9 pairs of pins, with each pair being one timing combination. To setup the board, the user needs to place a shorting plug in each row, under the timing number which he desires.

The timing numbers are marked to be the maximum return time for the signal involved (multiplied by 200ns). The minimum time is 200ns below the maximum time. For example, the pins marked "4" will return their signals from 600-800ns after a command is received. The pins marked "0" always return the signal immediately.

Since the XACK timing is tied to the access time of the board, the setting of that plug is suggested to be "3". The setting of the AACK strap will have to be determined by the system designer, using the information presented here.

5. Specifications

Word Size

8 bits

Access Time

450ns, maximum

Baud Rates Programmable

75, 150, 300, 600, 1200, 2400, 4800, 9600, and 19,200

Interrupt Sources

Any "transmitter empty" or "receiver full" condition can trigger an interrupt on any of the eight vectored interrupt lines of the Multibus. Straps for each USART determine whether receiver or transmitter interrupts can come from that chip.

Addressing

This board requires 16 I/O ports, and the base address for these ports can be on any 16 port boundary. Full 16-bit dip-switch I/O addressing is a strap selectable option.

The first eight ports on the board are divided between the USARTs. Each USART gets two consecutive ports, and their functions are as follows:

Address	Input Function	Output Function
0	Receiver Data Reg.	Transmiter Data Reg.
1	Status Register	Command Register

The interval timer uses the next four ports in the following manner:

Address	Input Function	Output Function
0	Read Counter 0	Load Counter 0
1	Read Counter 1	Load Counter 1
2	Read Counter 2	Load Counter 2

3	No-Operation	Write Control Word

The last four ports have no function on the board, even though the board acknowldeges commands for their addresses.

RS-232 Specifications

The drivers and receivers used on the board are the 1488 and 1489 type. This provides a standard interface for the following lines: TxD, RxD, DSR, CTS, DTR, and RTS.

Interface

All P1 signals meet the IEEE Multibus proposed specification.

Electrical Characteristics

Vcc= +5V ±5%
Vdd= +12V ±5%
Vbb= -12V ±5%
Icc= .85A typ, 1.0A max
Idd= .05A typ, .1A max
Ibb= .05A typ, .1A max

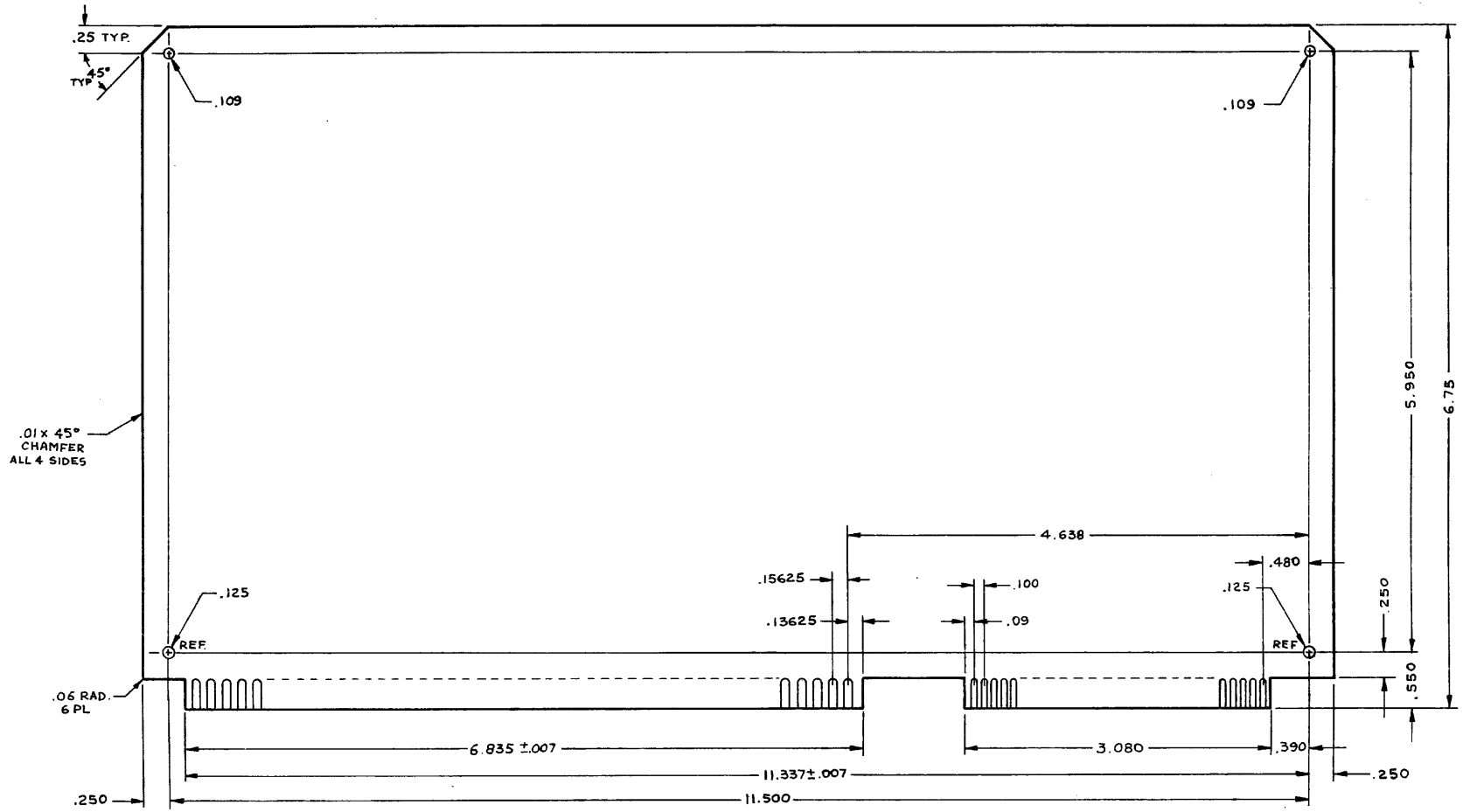
Physical Characteristics

Dimensions: See the basic Multibus dimensions on the following page. Each edge connector on the top of the board is 1.308" wide, with the right edge of each connector being 1.585", 3.885", 6.185", and 8.485" from the right-hand reference hole.

Weight: 9 oz (255gm)

Ordering Information

Part Number: B1019
Description: Multibus Quad Serial Interface Board



Central Data Corporation
P.O. Box 2530, Sta. A, Champaign, IL 61820 (217) 359-8010

NO.

SHEET OF

MULTIBUS BOARD DIMENSIONS

6. Software Driver Routines

The following pages show example driver routines written in Z8000 code. The routines are general purpose in nature, and are not meant to be used directly in any particular application.

```
LINE  ADDR   B1 B2 B3 B4 LABEL  OPCODE     OPERAND   COMMENTS

0001  000000             UN     EQU        8
0002  000000             Z      EQU        6
0003  000000             NZ     EQU        E
0004  000000             CY     EQU        7
0005  000000             NC     EQU        F
0006  002000             PL     EQU        D
0007  000000             MI     EQU        5
0008  002000             NE     EQU        E
0009  002000             EQ     EQU        6
0010  002000             OV     EQU        4
0011  200000             NOV    EQU        C
0012  000000             GE     EQU        9
0013  000000             LT     EQU        1
0014  002000             GT     EQU        A
0015  000000             LE     EQU        2
0016  000000             UGE    EQU        F
0017  000000             ULT    EQU        7
0018  002000             UGT    EQU        B
0019  000000             ULE    EQU        3
0020  000000             *
0021  000000             FORM   EQU        0C
0022  002000             LF     EQU        0A
0023  000000             CR     EQU        0D
0024  000000             ESC    EQU        1B
0025  002000             BS     EQU        08
0026  002000             BSC    EQU        28
0027  020020             BSW    EQU        0B
0028  000000             BSL    EQU        01
0029  002000             COPC   EQU        10
0030  002000             COPW   EQU        17
0031  002000             COPL   EQU        02
0032  000000             EDIT   EQU        03
0033  000000             CONX   EQU        18
0034  000000             *
0035  000000             FLGS   EQU        01
0036  000000             FCW    EQU        02
0037  002000             REF    EQU        03
0038  002000             PSEG   EQU        04
0039  002002             POFF   EQU        05
0040  000000             NSEG   EQU        06
0041  000000             NOFF   EQU        07
0042  002000             *
0043  000000             CRY    EQU        08
0044  000000             ZRO    EQU        04
0045  000000             SGN    EQU        02
0046  000000             PV     EQU        01
0047  000000             ALL    EQU        0F
0048  000000             *
0049  000000             *
0050  000000             *
0051  000000             INT2   EQU        40        ADDRESS OF FIRST PORT'S DATA REGISTER
0052  000000             *
0053  000000             *      THIS ROUTINE TRANSMITS A CHARACTER BY PUTTING IT
0054  000000             *      IN A RAM BUFFER 'WHICH IS READ BY THE INTERRUPT
0055  000000             *      SERVICE ROUTINE' IF THE USART IS BUSY TRANSMITTING
```

```
LINE  ADDR   B1 B2 B3 B4 LABEL  OPCODE     OPERAND    COMMENTS

0056  000000             *      OR BY WRITING IT DIRECTLY TO THE USART IF IT IS IDLE.
0057  000000             *
0058  000000             *      WHEN ENTERING THE ROUTINE, THE FOLLOWING REGISTERS
0059  000000             *      SHOULD BE SETUP:
0060  000000             *
0061  000000             *      RL0:  THE DATA BYTE TO BE TRANSMITED
0062  000000             *      R3:   THE ADDRESS OF THE PORT'S DATA REGISTER
0063  000000             *
0064  000000             *      THE ROUTINE RETURNS WITH ALL REGISTERS SAVED.
0065  000000             *
0066  000000 93 E2       TRDY   PUSH       @RR14,R2     SAVE THE REGISTERS
0067  000002 93 E3              PUSH       @RR14,R3
0068  000004 B3 34              RR         R3,#0        TWO ADDRESSES PER USART
0069  000006 07 03 00 03        AND        R3,#3        FOUR PORTS PER BOARD
0070  00000A 7C 01       TRD1   DI         0,1          DISABLE VECTORED INTERRUPTS
0071  00000C 30 0A 00 85        LDRB       RL2,OUTS     LOAD OUTPUT STATUS BYTE
0072  000010 26 03 0A 00        BITB       RL2,R3       TEST BIT FOR THIS PORT
0073  000014 E6 02              JR         Z,TRD3       OUTPUT BUFFER EMPTY
0074  000016 7C 05              EI         0,1          ALLOW INTERRUPTS BRIEFLY
0075  000018 E8 F8              JR         UN,TRD1      CHECK AGAIN
0076  00001A 30 02 00 78 TRD3   LDRB       RH2,OUTF     SEE IF TRANSMITTER IS IDLE
0077  00001E 26 03 02 00        BITB       RH2,R3
0078  000022 E6 13              JR         Z,TRD4       NO
0079  000024 22 03 02 00        RESB       RH2,R3       TURN IT BACK ON
0080  000028 32 02 00 6A        LDRB       OUTF,RH2     AND STORE THE NEW STATUS
0081  00002C 97 E3              POP        R3,@RR14     RESTORE R3
0082  00002E 21 02 02 27        LD         R2,#0227
0083  000032 3A 26 F0 02        OUTB       F002,RH2     MASK 8259 LEVEL 1 INTERRUPTS--PREVENT GLITCH
0084  000036 A9 30              INC        R3,#0        GET TO THE STATUS PORT
0085  000038 3E 3A              OUTB       @R3,RL2      START THE TRANSMITTER
0086  00003A AB 30              DEC        R3,#0        BACK TO THE DATA PORT
0087  00003C 3E 38              OUTB       @R3,RL0      SEND OUT THE BYTE
0088  00003E 8B 28              CLR        R2
0089  000040 3A A6 F0 02        OUTB       F002,RL2     ENABLE ALL INTERRUPTS
0090  000044 97 E2              POP        R2,@RR14
0091  000046 7C 05              EI         0,1
0092  000048 9E 09              RET        UN
0093  00004A 6E 38 82 00 TRD4   LDP        <^OUTH,OUTH(R3),RL0   PUT THE CHAR INTO THE BUFFER
0094  00004E 00 90
0095  000050 24 03 0A 00        SETB       RL2,R3       INDICATE A CHARACTER IS WAITING
0096  000054 32 0A 00 3D        LDRB       OUTS,RL2     STORE THAT STATUS
0097  000058 97 E3              POP        R3,@RR14     RESTORE THE REGISTERS
0098  00005A 97 E2              POP        R2,@RR14
0099  00005C 7C 05              EI         0,1          ENABLE INTERRUPTS
0100  00005E 9E 08              RET        UN           LEAVE
0101  000060             *
0102  000060             *      THIS ROUTINE WAITS FOR A CHARACTER TO BE RECEIVED
0103  000060             *      BY A USART.  IT POLLS A STATUS BYTE TO DETERMINE
0104  000060             *      IF THE INTERRUPT DRIVER HAS GOTTEN A CHARACTER, AND
0105  000060             *      LOADS THAT CHARACTER WHEN IT IS AVAILABLE.
0106  000060             *
0107  000060             *      R3 SHOULD HAVE THE ADDRESS OF THE USART'S DATA
0108  000060             *      REGISTER ON ENTRY, AND RL0 WILL CONTAIN THE DATA
0109  000060             *      BYTE RECEIVED WHEN THE ROUTINE RETURNS.  ALL OTHER
0110  000060             *      REGISTERS ARE SAVED.
```

```
LINE  ADDR   B1 B2 B3 B4 LABEL  OPCODE     OPERAND   COMMENTS

0111  000060             *
0112  000060 91 E2       RRDY   PUSHL      @RR14,RR2    SAVE THE REGISTERS
0113  000062 B3 34            RR         R3.#0        TWO ADDRESSES PER USART
0114  000064 07 03 00 03      AND        R3,#3        FOUR PORTS PER BOARD
0115  000068 7C 01       RRD1   DI         0,1          DISABLE VECTORED INTERRUPTS
0116  00006A 30 0A 00 26      LDRB       RL2,INS      LOAD INPUT STATUS BYTE
0117  00006E 26 03 0A 00      BITB       RL2,R3       SEE IF CHARACTER IS WAITING
0118  000072 EE 02            JR         NZ,RRD3      BIT SET MEANS YES
0119  000074 7C 05            EI         0,1          ENABLE INTERRUPTS BRIEFLY
0120  000076 E8 F8            JR         UN,RRD1      AND TRY AGAIN
0121  000078 22 03 0A 00 RRD3   RESB       RL2,R3       WE ARE TAKING THE BYTE
0122  00007C 32 0A 00 14      LDRB       INS.RL2      STORE NEW STATUS
0123  000080 60 38 80 00      LDB        RL0,<^INH>INH(R3)   LOAD THE BUFFERED DATA
0124  000084 00 8C
0125  000086 95 E2            POPL       RR2,@RR14    RESTORE THE REGISTERS
0126  000088 7C 05            EI         0,1          ALLOW INTERRUPTS
0127  00008A 9F 08            RET        UN           RETURN
0128  00008C             *
0129  00008C             *
0130  00008C 00 00 00 00 INH    SAVE       4         HOLDS INPUT DATA WAITING FOR RRDY
0131  000090 00 00 00 00 OUTH   SAVE       4         HOLDS OUTPUT DATA WAITING FOR INTERRUPT
0132  000094 00          INS    SAVE       1         BIT SET MEANS INPUT CHAR WAITING
0133  000095 00          OUTS   SAVE       1         BIT SET MEANS OUTPUT CHAR WAITING
0134  000096 00          OUTF   SAVE       1         BIT SET MEANS TRANSMITTER IS DISABLED
0135  000097             *
0136  000097             *       THIS ROUTINE SHOULD BE EXECUTED WHEN AN INTERRUPT
0137  000097             *       OCCURS FROM THE SERIAL INTERFACE BOARD.  IT POLLS
0138  000097             *       THE STATUS BYTE OF EACH PORT, AND TAKES ACTION
0139  000097             *       ACCORDINGLY.
0140  000097             *
0141  000098 91 F0       INTH   PUSHL      @RR14,RR0    SAVE ALL REGISTERS
0142  00009A 91 F2            PUSHL      @RR14,RR2
0143  00009C 8D 28            CLR        R2           USED TO SET BITS IN STATUS BYTE
0144  00009F 21 01 00 40      LD         R1,#INT0     START WITH FIRST PORT
0145  0000A2 A9 10       INTA   INC        R1,#0        GET THE STATUS ADDRESS
0146  0000A4 3C 18            INB        RL0,@R1      READ IT
0147  0000A6 AB 10            DEC        R1,#0        BACK TO DATA ADDRESS
0148  0000A8 A6 81            BITB       RL0,#1       SEE IF RECEIVER IS READY
0149  0000AA E6 0A            JR         Z,INTB       NO
0150  0000AC 3C 10            INB        RH0,@R1      READ THE DATA BYTE
0151  0000AE 6F 20 83 00      LDB        <^INH>INH(R2),RH0   SAVE THE BYTE
0152  0000B2 00 8C
0153  0000B4 30 0B FF DC      LDRB       RL3,INS      SET STATUS BIT
0154  0000B8 24 02 0B 00      SETB       RL3,R2
0155  0000BC 32 0B FF D4      LDRB       INS.RL3
0156  0000C0 A6 80       INTB   BITB       RL0,#0       SEE IF TRANSMITTER IS EMPTY
0157  0000C2 E6 1E            JR         Z,INTC       NO
0158  0000C4 30 0B FF CD      LDRB       RL3,OUTS     IS A CHAR WAITING
0159  0000C8 26 02 0B 00      BITB       RL3,R2
0160  0000CC E6 09            JR         Z,INTD       NO
0161  0000CE 22 02 0B 00      RESB       RL3,R2       RESET STATUS
0162  0000D2 32 0B FF BF      LDRB       OUTS.RL3
0163  0000D6 60 20 87 00      LDB        RH0,<^OUTH>OUTH(R2)   TAKE THE CHARACTER
0164  0000DA 00 90
0165  0000DC 3E 10            OUTB       @R1,RH0      SEND IT OUT
```

```
LINE  ADDR    B1 B2 B3 B4 LABEL  OPCODE     OPERAND    COMMENTS

0166  0000DE  E8 0D              JR         UN,INTC    NEXT PORT
0167  0000E0  A9 10        INTD  INC        R1,#0      NO CHARACTERS WAITING TO GO
0168  0000E2  3C 10        INTE  INB        RH0,@R1    READ STATUS
0169  0000E4  A6 02              BITB       RH0,#2     DONT CUT OFF END OF LAST CHAR
0170  0000E6  E6 FD              JR         Z,INTE
0171  0000E8  C0 26              LDB        RH0,#26
0172  0000EA  3E 10              OUTB       @R1,RR0    TURN TRANSMITTER OFF
0173  0000EC  AB 10              DEC        R1,#0      BACK TO DATA PORT
0174  0000EE  30 00 FF A4        LDRB       RH0,OUTF
0175  0000F2  24 02 00 00        SETB       RH0,R2
0176  0000F6  32 00 FF 9C        LDRB       OUTF,RH0   OUTPUT NOW OFF
0177  0000FA  A9 11        INTC  INC        R1,#1
0178  0000FC  A9 20              INC        R2,#0
0179  0000FE  0B 02 00 04        CP         R2,#4
0180  000102  EF CF              JR         NE,INTA
0181  000104  7C 01              DI         0,1        DONT ALLOW DEADLY LOOPS
0182  000106  C8 20              LDB        RL0,#20
0183  000108  3A 86 F0 00        OUTB       F000,RL0   CLEAR ISR OF 8259
0184  00010C  95 E2              POPL       RR2,@RR14  RESTORE REGISTERS
0185  00010F  95 E0              POPL       RR0,@RR14
0186  000110  7B 00              IRET                  RETURN FROM INTERRUPT
```

7. Schematics

The following pages contain the schematics for the Quad Serial Interface board. A full description of the circuitry is given in the Principles of Operation section of this manual.

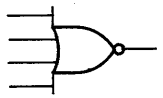
3-INPUT 'AND' GATE

2-INPUT 'OR' GATE

INVERTER

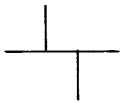
2-INPUT 'NAND' GATE

4-INPUT 'NOR' GATE

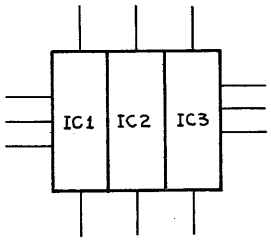


TWO LINES - NO CONNECTION

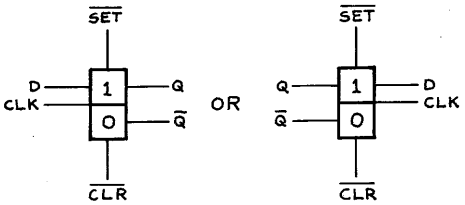
3 LINES - ALL CONNECTED



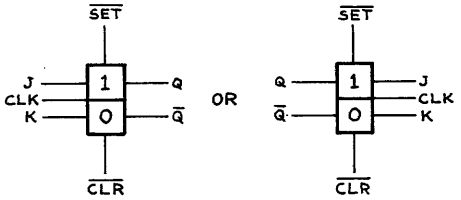
GROUP OF SIMILAR PARTS



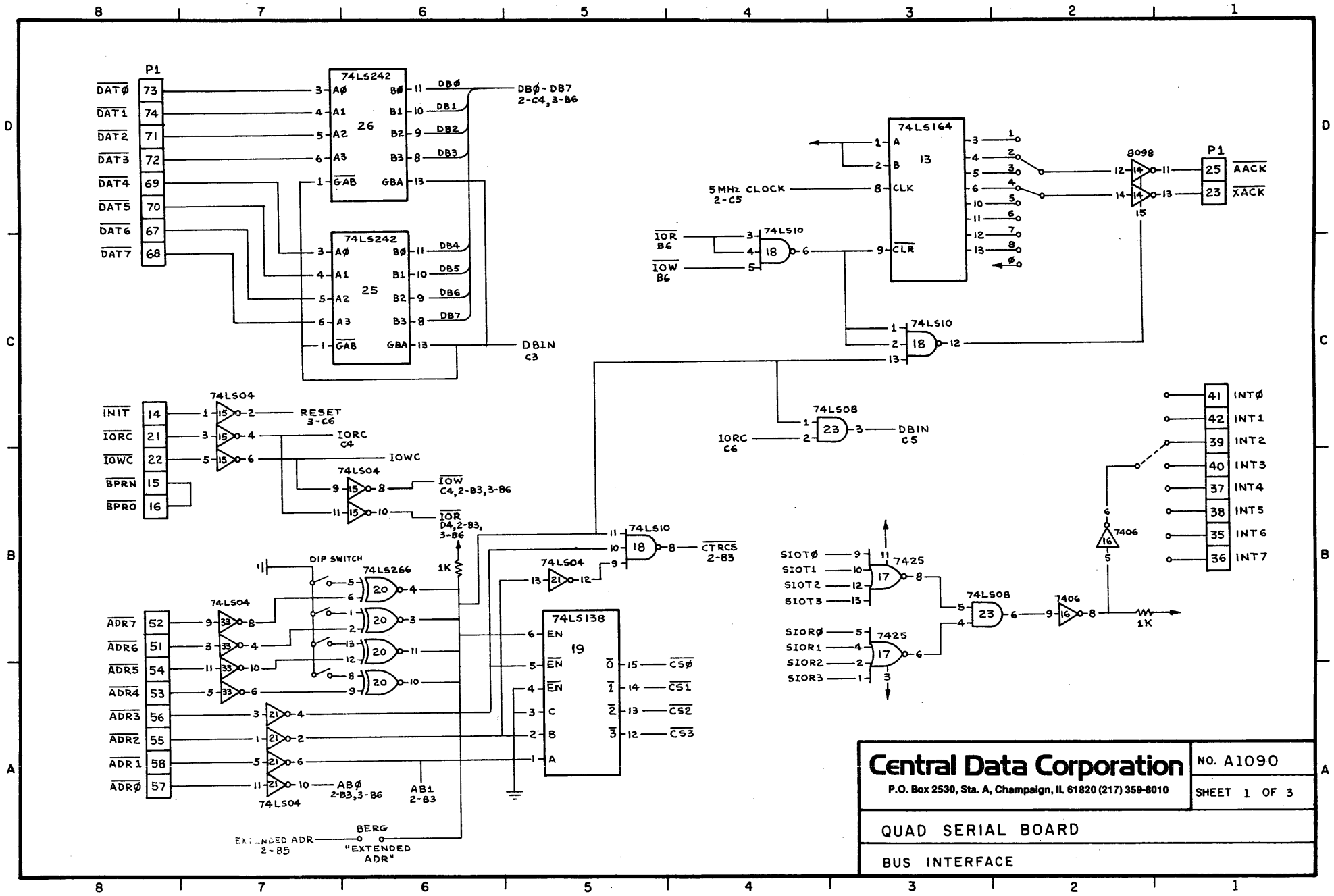
D-TYPE FLIP-FLOP



J/K FLIP-FLOP



Central Data Corporation P.O. Box 2530, Sta. A, Champaign, IL 61820 (217) 359-8010	NO. SHEET OF
DRAWING CONVENTIONS	



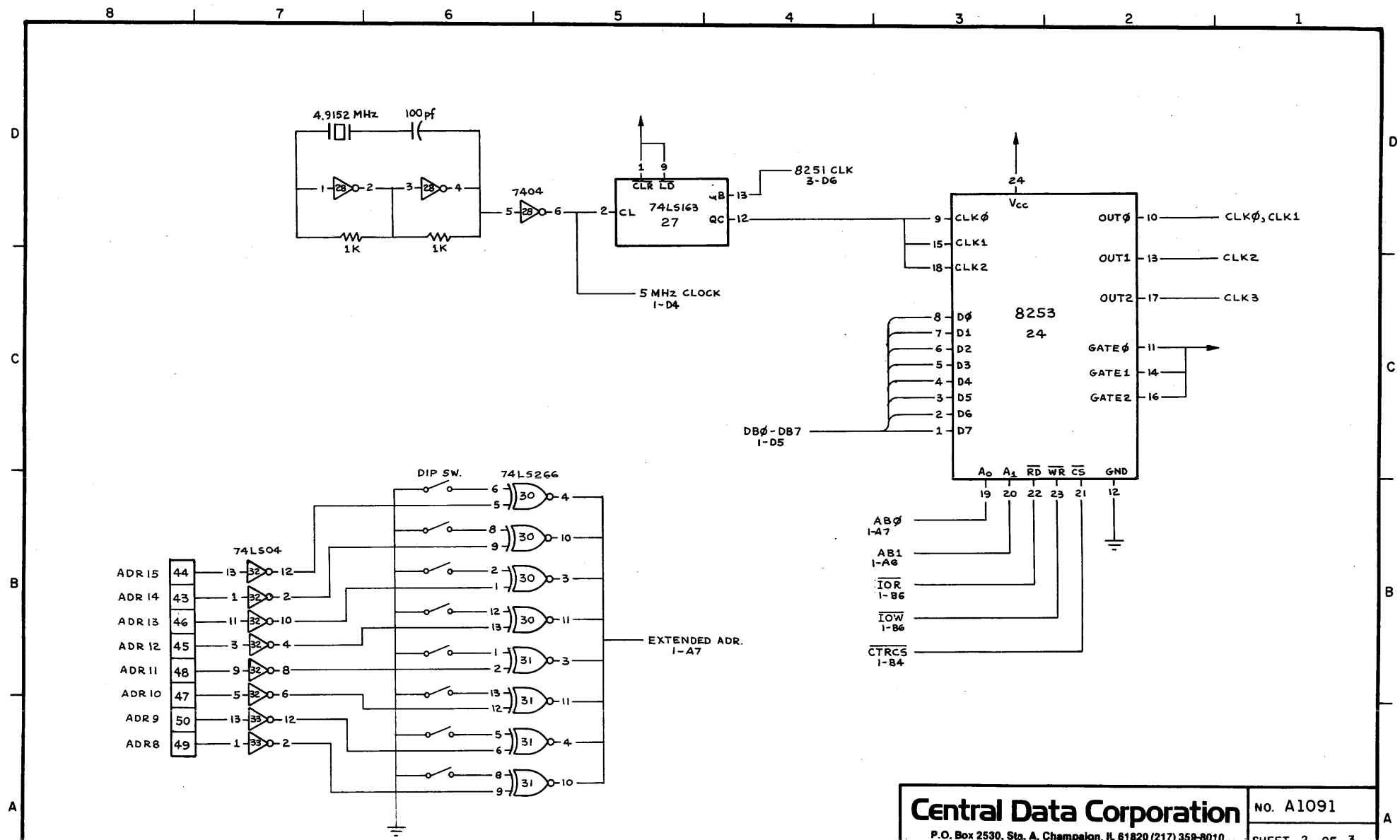
Central Data Corporation
P.O. Box 2530, Sta. A, Champaign, IL 61820 (217) 359-8010

NO. A1090

SHEET 1 OF 3

QUAD SERIAL BOARD

BUS INTERFACE



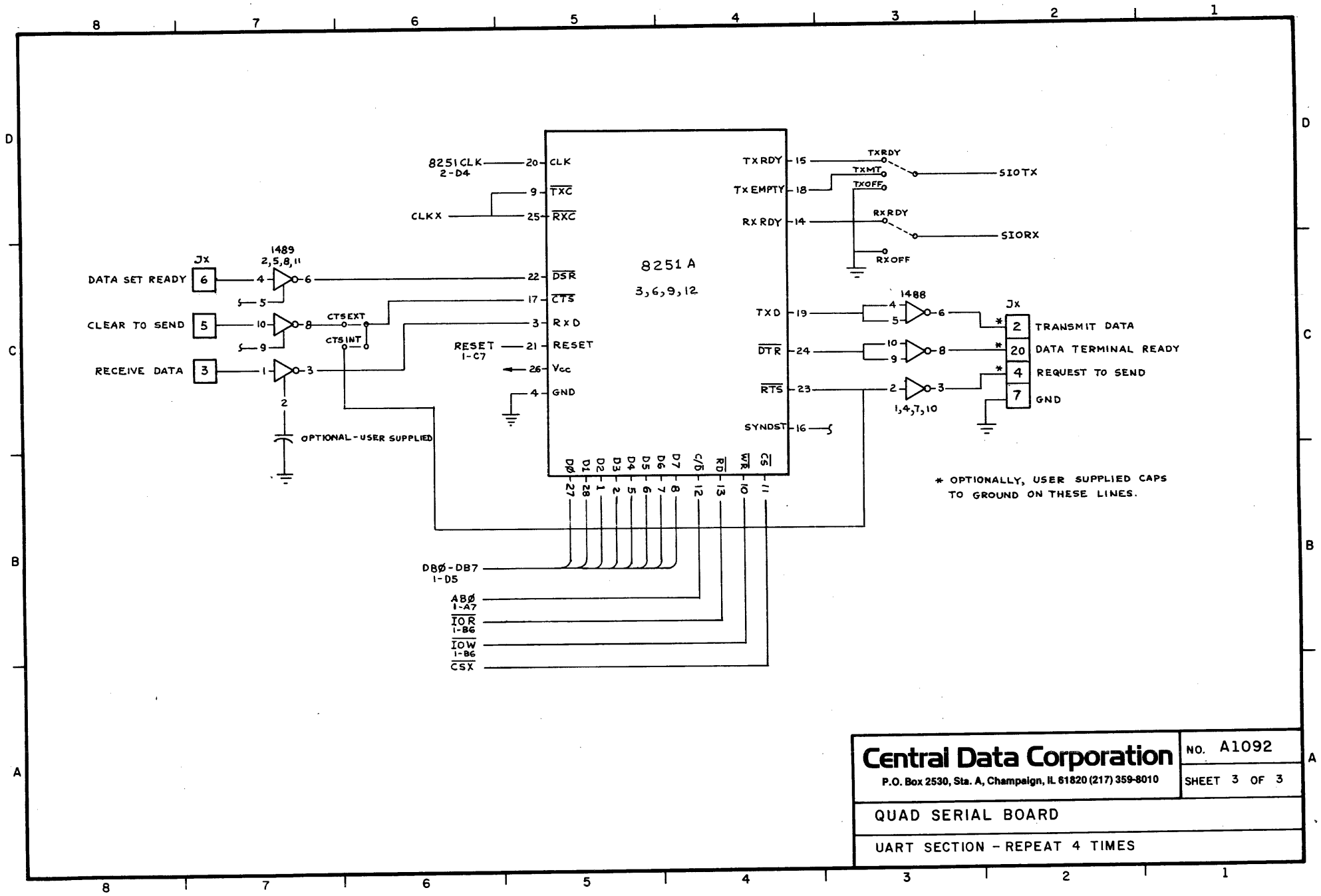
Central Data Corporation
P.O. Box 2530, Sta. A, Champaign, IL 61820 (217) 359-8010

NO. A1091

SHEET 2 OF 3

QUAD SERIAL BOARD

TIMER



Central Data Corporation P.O. Box 2530, Sta. A, Champaign, IL 61820 (217) 359-8010	NO. A1092 SHEET 3 OF 3
QUAD SERIAL BOARD	
UART SECTION - REPEAT 4 TIMES	