High Performance Intelligent Terminal Controller

CD91/3100-HMAN

Central Data Corporation
1602 Newton Drive
Champaign, IL 61821-1098
Phone: (217) 359-8010

TABLE OF CONTENTS

1. General Information

The High Performance Intelligent I/O Board is an improved version of the Intelligent Octal Terminal Controllers produced by Central Data (P/N B1Ø25 and B1Ø44). This board allows up to eight asynchronous EIA RS-232 interfaces to be attached to any Multibus[1] system. Each interface is controlled by a USART which has an on-chip baud rate generator. This enables the host processor to set each USART at a different speed. Available baud rates range from 5Ø to 19,2ØØ baud.

The USARTs are controlled by an on-board 8MHz 8Ø88 microprocessor. This processor handles all commands from the host and manages all data transfers between the USARTs and the 64K dual-port RAM. Up to 32K bytes of EPROM and 8K bytes of static RAM are available to the 8Ø88, allowing extremely flexible custom drivers to be written, where required. The standard drivers available with the board include extensive self diagnostic routines, which are executed automatically each time the board is reset. If an error is detected, an error code is displayed on four status LEDs, and the board will never indicate a "ready" status to the host. If no errors are detected, one status LED will blink, and the board will indicate to the host that it is ready to accept commands.

A 64K byte dual-port RAM is available for communication between the on-board 8Ø88 and the Multibus. Commands and output data are placed into the dual-port RAM by the host processor. The 8Ø88 microprocessor responds by performing the required operations, and returns status and/or input data. Status and input data are then accessed from the dual-port RAM by the host processor.

Parity error detection is supported on the dual-port RAM to improve data integrity. If enabled, a parity error will cause a non-maskable interrupt to the 8Ø88, which will execute a section of code designed to handle the occurance of a parity error. This routine might, for example, interrupt the host, log the error in dual-port RAM, or simply indicate the parity error with a 4-bit code on the status LEDs (depending on firmware design). Both byte and word transfers are supported on the host side of the

[1]Multibus is a trademark of Intel Corporation, and is used throughout this manual.

dual-port RAM, and the RAM may be locked from either port. Also, host accesses to the dual-port RAM may be disabled by the use of the Multibus INH1 signal.

There is one I/O port on the board for communication between the host and the board. The host issues commands to the board by presetting the dual-port RAM with the command information and then writing a command byte to the I/O port. Before issuing any command, the host should read the I/O port and verify that data bit Ø is high. When bØ is low, the board is busy, and the command port should not be written to. When bØ is high, the board is ready to accept a new command. All immediate commands will be completed before bØ is set high again, while time dependent commands will set bØ high again as soon as the operation has been fully initiated. Both 8-bit and 16-bit I/O addressing are supported. Therefore, the I/O port may be addressed on a one-port boundary within the 256-byte or 64K-byte I/O addressing space.

Since the board is controlled by a microprocessor chip, its function is totally programmable. A Standard High-Performance Driver is normally provided with the board, as described later in this manual. Also, an Emulation Mode Driver is available which configures the board to operate exactly like a B1Ø25/B1Ø44 (Central Data's previous generation Intelligent I/O Board) with Monitor/OS Driver firmware, with the exception that this board has 64K of dual-port RAM, and therefore must be addressed on a 64K boundary (the previous board had only 16K of RAM). This driver does not take advantage of any of the extra features provided by the High-Performance I/O board (except improved thruput). Its purpose is to allow present users of the B1Ø44 to evaluate the High-Performance I/O board without changes to their existing software. The evaluation driver is not described in this manual, since it is not suggested that any new applications should use it. Central Data has the capability to write custom drivers to meet any asynchronous application. Requests for such custom design should be referenced directly to Central Data.

2. Functional Description

The B1Ø31 High Performance Octal 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 processor section contains an 8Ø88-2 microprocessor, its buffers and support circuitry, EPROM, RAM, the internal I/O and memory address decoders, the error status LEDs, and the two crystal oscillator circuits used to generate the clock signals required on the board. Up to 32K bytes of EPROM and 8K bytes of static RAM are available to the 8Ø88 microprocessor. Neither of these resources can be accessed from the Multibus.

The Multibus interface consists of address and command line buffers, the Multibus memory and I/O address decoders, and the I/O port circuitry.

The dual-port RAM circuitry contains arbitration and control circuitry, 64K bytes of dynamic RAM, an 82Ø3 dynamic RAM controller, parity checking circuitry, data buffers, and the RAM address multiplexers which select between the internal and the Multibus address lines.

The Multibus interrupt control circuitry allows both bus-vectored and non-bus-vectored interrupts to be supported by the board, under 8Ø88 control.

The USART section of the board is the actual interface to the outside world. This section is repeated on the board eight times, which gives eight 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.

3. Principles Of Operation

This section details the operation of the entire 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 separate page, it will have the grid location preceeded by the sheet number (i.e. 2-B5).

An equation is used to define each PAL (Programmable Array Logic device) output. All PAL equations are done in logical form, without regard for the active state (high/low) of each referenced signal. Thus, PAL equations use a trailing slash to indicate an inactive signal, and the absence of the slash indicates the signal must be active. The determination of whether the actual signals are active-high or active-low does not need to be made until the device is programmed, and is not described in this manual. Also, the '*' symbol represents the AND condition, while the '+' symbol represents the OR condition.

This section of the manual describes the board in general terms--no specific driver program is assumed to be in EPROM. For details on the driver normally provided with the board, see section 4 of this manual.

Microprocessor and Support Circuitry

Sheet 1 of the schematics contains the 8088 microprocessor and its associated circuitry. The 8088-2 is a 16-bit microprocessor with an 8-bit external data bus which runs at 8MHz. Full programming details can be obtained from Intel, the manufacturer of the device.

The 8284A clock driver (IC60) and associated circuitry generates the CLK READY and RESET signals for the 8088. The 8284A divides the 24MHz output from a crystal oscillator (IC46) by three to provide an 8MHz clock signal at the appropriate duty cycle.

The buffered Multibus INIT signal drives the RES/ input to the 8284A, which in turn generates the RESET signal which is used to initialize the board.

The READY input to the 8088 must be activated after a cycle has been initiated to inform the processor that the transaction has been completed, and that it can terminate the bus cycle. The 8284A accepts the asynchronous READY signals from the selected devices, and generates a synchronous READY signal to the 8088. Accesses to the static RAM or EPROM, and 8088 interrupt acknowledge cycles require no wait states. The occurance of one of these types of the cycles will cause the AEN2/ input to the 8284A to be driven low, causing the READY signal to be asserted. All I/O cycles require one wait state, while accesses to the dual-port RAM are terminated by the IACK/ signal. For either type of cycle, the RDY1 input to the 8284A will be driven high when the cycle is complete, causing the READY signal to be generated.

The 8088 multiplexes address, data, and status information on a common set of 20 I/O pins. When ALE is active, a 20-bit address is valid on the AD0-A19 pins. Address lines A0-A7 are latched into IC50 while address lines A16-A19 are latched into IC49. Address lines A8-A15 are not multiplexed, and therefore do not need to be latched.

The MEM signal and the upper 4 address lines are used to select different sections of the board as follows:

Address Range	Memory Access (MEM=1)	I/O Access (MEM=0)
00000H-0FFFFH	Static RAM	8088 I/O ports (control)
10000H-2FFFFH	Illegal	Illegal
30000H-3FFFFH	Dual-Port RAM (unlocked)	Illegal
40000H-6FFFFH	Illegal	Illegal
70000H-7FFFFH	Dual-Port RAM (locked)	Illegal
80000H-EFFFFH	Illegal	Illegal
F0000H-FFFFFH	EPROM	Illegal

When writing custom drivers, it should be noted that the upper four address lines are not fully decoded, and therefore issuing commands to the address ranges listed above as Illegal will cause undesired results.

A 28-pin JEDEC compatible IC socket (IC32) is used to provide the 8088 with a scratchpad RAM area. This RAM is used to hold temporary data and the stack, and is not accessible through the Multibus. A 3-pin jumper array is provided to configure the socket to accept 16K (2Kx8), or 64K (8Kx8) static RAMs.

A full 64K byte page is allocated to the static RAM, even though the maximum RAM size is 8K. As a result, the available RAM will be repeated throughout the 64K-byte address range (i.e. 2K-byte RAMs: Ø-7FFH = 8ØØH-FFFH, etc.).

A 28-pin JEDEC compatible IC socket (IC31) provides the 8Ø88 with its program storage (firmware). A 3-pin jumper array is provided to configure the socket to accept 64K (8Kx8), 128K (16Kx8), or 256K (32Kx8) EPROMs.

The 8Ø88 supports two types of external interrupts: vectored interrupts (INTR), and non-maskable interrupts (NMI). INTR is a level triggered (active high) input which can be internally masked by resetting the interrupt enable bit of the processor. Once recognized, the 8Ø88 enters an interrupt acknowledge cycle and receives the vector number Ø5H from IC71. This causes the 8Ø88 to use the sixth entry in the interrupt vector lookup table (located in scratchpad RAM) as the address for the interrupt routine. NMI is an edge triggered (low to high) input which is not internally maskable. Unlike INTR, NMI does not generate an interrupt acknowledge cycle to acquire a vector. Instead, it always uses the third entry in the interrupt vector lookup table as its destination address.

There are three possible conditions that may cause an 8Ø88 interrupt:

1. A new character is received by any USART.
2. A new command is written by the host.
3. A parity error occurs in the dual-port RAM.

The use and definition of each type of interrupt will be determined by the driver installed on the board and the CMD INT jumper. All three conditions may cause interrupts, or a polled operation may be selected, where no interrupts are allowed.

The USARTs drive their RXRDY/ output pin low as each character is received. The RXRDY/ pins from all eight USARTs are tied together to form the USART INT/ signal, which is inverted to form the UP INT signal, which interrupts the processor. If 8Ø88 interrupts are enabled, a type 5 interrupt will be generated when UP INT goes high. Alternately, the 8Ø88 may disable interrupts and sample the UP INT line by reading the status port, if polled operation is desired.

The Multibus I/O port circuitry located on sheet 2 of the schematics is described later in this manual. For now, note that when the host writes a new command to the board's Multibus I/O

port, the NEW CMD line goes high. NEW CMD is inverted and connected to the CMD INT jumper pins. If the CMD INT jumper is installed, an 8088 interrupt (INTR) will be generated when the host writes to the board's Multibus I/O port. The 8088 may also determine the state of NEW CMD by reading the status port, if polled operation is desired.

The status port is also used to monitor two other signals, INT LOCK and INT RQ. The INT LOCK signal is generated in the dual-port RAM control circuitry shown on sheet 3 of the schematics, while the INT RQ signal is generated by the Multibus interrupt circuitry shown on sheet 4 of the schematics.

When the 8088 does a locked access to the dual-port RAM, INT LOCK will be high, locking the Multibus out of the dual-port RAM. During the self test, the 8088 can issue locked and unlocked dual-port RAM cycles, and verify the proper state of the INT LOCK line.

When the controller is in bus-vectored interrupt (BVI) mode, and a Multibus interrupt has been generated on the bus, the 8088 must be able to determine if the host has accepted the interrupt, or if it is still pending. When INT REQ is high, the interrupt has not been acknowledged by the host. Once the host has acknowledged the interrupt and received the interrupt vector from the board, INT REQ will go low.

The dual-port RAM and parity detect circuitry is located on sheet 3 of the schematics, and is discussed in detail later in this manual. If the parity detect circuit is enabled, a parity error will cause the PAR ERR signal to go high, thus causing an 8088 non-maskable interrupt.

The 8088 I/O circuitry (not to be confused with the Multibus I/O port), is used to select the eight USARTs, and to control and monitor events on the board. Two 74LS138, 3-8 line decoders (IC29 and IC38) are used to select the different I/O devices used by the 8088. The USARTs will be selected by IC29 during any 8088 I/O operation in which address line BA7 is high, with address lines BA2-BA4 determining which USART is selected. Alternately, IC38 will be enabled during any 8088 I/O operation when address line BA7 is low, with address lines BA4-BA6 determining which of the eight output pins of IC38 are selected (low). The outputs of IC38 are used to select (read, or write) the remaining I/O ports on the board. Complete memory and I/O maps are provided in the Programming Information section of this manual.

A 74LS259 8-bit addressable latch (IC3Ø) drives the four error status LEDs and selects several options available on the board. During an 8Ø88 I/O write in which WR STAT/ is low, IC3Ø will be selected. Address lines BAØ-BA2 will select which of the eight outputs is affected. The state of address line BA3 is latched into the selected output while WR STAT/ is active.

The BVI output selects either bus-vectored interrupt mode (BVI=1), or non-bus-vectored interrupt mode (BVI=Ø).

The NO LOCK/ output determines whether the Multibus will be allowed to lock the dual-port RAM. When NO LOCK/ is active (low), the Multibus is prevented from locking the on-board 8Ø88 out of the dual-port RAM, and any attempt to do so will be ignored. If NO LOCK/ is high, the Multibus is allowed to lock the dual-port RAM. This is accomplished by the host asserting the Multibus LOCK/ signal during an access to the dual-port RAM, as set forth in the Multibus specification.

The DIS PAR INT/ output disables the parity error detection interrupt output when active (low), and enables it when inactive (high).

When the board is reset, the four error status LEDs are all forced on, Non-Bus-Vectored interrupt mode is selected, and both the Multibus lock operation and the parity error detection interrupt output are disabled.

A 2Ø.2752MHz crystal oscillator (IC27), two sections of a 74S74 flip-flop (IC26), and an inverter provide the five remaining clocks required by the board. The complimentary 2ØMHz and 1ØMHz signals are used in two separate arbiters. In addition, the 2ØMHz clock is used by the 82Ø3 memory controller. The 5.Ø688MHz output from pin 5 of IC26 is used by the USARTs to generate the different baud rates.

Multibus Interface Circuitry

Sheet 2 of the schematics contains the Multibus address and command line buffers, the memory and I/O address comparators, and the I/O port circuitry.

The Multibus address lines and control signals are buffered by 74LS24Ø inverting buffers (IC73, IC74, IC78, and IC79). Address lines A16/-A23/ do not require buffers since they only connect to

the 74LS688 octal comparator (IC76), which meets Multibus loading specifications.

The memory address comparator circuit consists of a 74LS688 octal comparator and an 8-position dip-switch (IC75). This circuit allows the user to set the base address of the dual-port RAM on any 64K-byte boundary within the Multibus 16M-byte address space. The octal comparator does a comparison between the upper 8 Multibus address lines and the eight switch settings. If the address lines are equal to the dip-switch settings, and the Multibus INH1 line is inactive, MEM AEN/ will go low, indicating that the dual-port RAM has been selected. This signal is gated with the Multibus MRDC or MWRC signals by PAL3 (IC37, sheet 3) to produce the MBREQ/ signal, which initiates Multibus dual-port RAM cycles.

Address lines MA16/-MA23/ are active low inputs. It is therefore necessary to close the memory address comparitor switches to compare for an active address line, and to open the switches to indicate an inactive address line.

If the INH1/ line is activated after a memory cycle has been initiated, MEM AEN/ will switch high, preventing the board from driving the Multibus XACK/ or data lines.

The board contains one I/O port for communication between the on-board 8088 microprocessor and the host. After placing command information into the dual-port RAM (as required), the host initiates an operation by writing a command or control byte into the I/O port. Before the I/O port is ever written to, the host must read the I/O port and verify that data bit 0 is high (ready).

The board's Multibus I/O circuitry consists of four main sections:

1. The Multibus I/O address comparator
2. The Multibus XACK circuit
3. The Multibus I/O write port
4. The Multibus I/O read port

The Multibus I/O address comparator circuit allows the user to set the base address of the controller I/O port. Both 8-bit and 16-bit I/O addressing are supported, allowing the base address of the I/O port to be set on a 1 port boundary within a 256-port or 64K-port addressing range, respectively.

The I/O address comparator circuit consists of two 74LS688 octal comparators (IC54 and IC66), two 8-position dip-switches (IC65 and IC55), a 3-pin jumper array, and some other logic devices.

Each 74LS688 comparator determines the relationship between eight Multibus address lines and an 8-position dip-switch. If the two sets of inputs are equal, the P=Q/ output (pin 19) of the comparator device will be active. IC66 compares address lines MAØ-MA7 with the switches on IC55, while IC54 compares address lines MA8-MA15 with the switches on IC65. When open, the I/O address comparator switches compare for an active address line, and when closed, the switches compare for an inactive address line.

A 3-pin jumper array is provided to select between 8- and 16-bit I/O addressing. If 8-bit I/O addressing is selected, the comparator output for the upper eight address lines is ignored.

The I/O comparator circuit output (pin 18 of IC73) is gated with the IORC and IOWC signals to form the MRDIO/ and MWRIO/ signals. The MRDIO/ and MWRIO/ signals, when active, indicate that a Multibus I/O read or write cycle is in progress, respectively.

The controllers XACK circuit generates the Multibus XACK/ signal for interrupt acknowledge, memory, and I/O cycles. When either MACK/ (for memory cycles) or ACKOUT/ (for I/O and interrupt acknowledge cycles) goes low, pin 11 of IC39 follows, enabling IC42 and driving the Multibus XACK/ line low. The purpose of the two inverters between pin 11 of IC39 and pin 15 of IC42 is to allow the XACK/ line to be driven high before IC42 goes tri-state.

ACKOUT/ is generated by PAL1 (IC28, sheet 4). ACKOUT/ will be active when either MRDIO/ or MWRIO/ are active to generate the Multibus XACK/ signal for the I/O cycle currently in progress.

The Multibus I/O write port is a 74LS534 octal inverting latch (IC68). The inputs of IC68 are tied directly to the Multibus data lines DØ/-D7/, while its outputs drive the 8Ø88 data lines BDØ-BD7.

When the host writes to the board's I/O port, MWRIO/ goes low indicating I/O write selection. MWRIO/ drives the clock input of the latch and the NEW CMD flip-flop. It also causes ACKOUT/ to switch low, generating the XACK/ signal. When the host terminates the I/O write cycle, MWRIO/ switches high, latching the data, and setting the NEW CMD flip-flop.

The NEW CMD line drives the input to the Multibus I/O read buffer. When the host issues an I/O read to the controller MRDIO/ switches low, enabling the I/O read buffer, and driving the state of NEW CMD onto the Multibus data line DØ/. The Multibus read buffer is non-inverting, so the host should interpret the received bit as being busy if low and ready if high.

As previously mentioned, NEW CMD can be polled by the 8Ø88, or it can cause an 8Ø88 interrupt if the CMD INT jumper is installed. In either case, once recognized, the 8Ø88 reads the command by issuing an 8Ø88 I/O read to RDPORT/, which enables the output drivers of the latch. The CMD RDY flip-flop should be reset after the requested action has been completed. For time dependant command (i.e. output data), the flip-flop should be reset after the requested action has been fully initiated. This flip-flop is reset when the 8Ø88 writes to the WR CMD CPT/ I/O port.

Dual-Port RAM

Sheet 3 of the schematics contains the dual-port RAM and its associated control and interface circuitry. The main elements of the dual-port RAM circuitry are as follows:

1. The Multibus/Internal cycle arbiter and control circuit
2. The 82Ø3 dynamic RAM controller
3. The address multiplexers
4. The dynamic RAM array
5. The Multibus data path
6. The 8Ø88 data path
7. The parity error detection circuit

The dual-port RAM is the primary communication mechanism between the on-board 8Ø88 microprocessor and the host. It is therefore possible to receive simultaneous requests from both the 8Ø88 and the Multibus host, thus requiring an arbiter circuit to process these requests in an orderly fashion and prevent conflicts.

Furthermore, the dual-port RAM supports both 8-bit and 16-bit transfers with the Multibus, and 8-bit transfers with the 8Ø88, and it allows either port to lock the other port out of the RAM. The dual-port RAM arbiter and control circuit controls all of these operations.

When the host issues a memory request to the dual-port RAM, the MEM AEN/ signal is output from the memory address comparator circuit. Also, either the MRDC or MWRC signal will be true.

This causes the MBREQ/ output from PAL3 (IC37) to switch low, which sets the MULTIBUS CYCLE REQUEST flip-flop (pin 5 of IC62). The definition of MBREQ/ is as follows:

```
MBREQ = MRDC * MEM AEN
      + MWRC * MEM AEN
```

Alternately, when the 8088 issues a dual-port RAM request, its BA17 address line will be high and BA19 will be low. When the 8088 CMD line goes high, the 8088 CYCLE REQUEST flip-flop (pin 9 of IC40) will be set.

The arbitration between these two requests is handled by both halves of a 74F74 flip-flop (IC25) and the END MBCYC/ and CLR INT CYC/ outputs of PAL3. The data inputs to each half of the device connect to the two CYCLE REQUEST flip-flops, which will be high if a cycle is desired. The two clock inputs to the arbiter flip-flops are complimentary 10MHz signals derived from a common source, so that the two sections of IC25 are clocked on alternate edges of the clock signal. Therefore, if simultaneous Multibus and 8088 dual-port cycle requests are received, one half of the arbiter will initially see its request 50ns ahead of the other. Once either half of arbiter is set, the reset input to the other half is switched low, holding off the other request.

The arbiter outputs, MBCYC and INT CYC, are used to start Multibus and 8088 dual-port RAM cycles, respectively. The ST RD CYC/ and ST WR CYC/ inputs to the 8203 dynamic RAM controller are generated by PAL2 and initiate read and write cycles as follows:

```
ST RD CYC = MBCYC * MRDC
          + INT CYC * RD

ST WR CYC = MBCYC * MWRC
          + INT CYC * WR
```

When INT CYC switches high, a memory cycle will be initiated by the 8203 dynamic RAM controller, and the INT LOCK flip-flop will be set or reset depending on the state of BA18. At the end of the cycle, IACK/ will switch low to reset the 8088 CYCLE REQUEST flip-flop and to provide the READY signal to the 8088. The output of the INT LOCK flip-flop will lock the Multibus out of the dual-port RAM if it is high. This line is also conected to the RD STAT/ buffer so the function can be tested by the 8088.

Both 8- and 16-bit transfers are supported on the Multibus port, even though the dual-port RAM itself is only 8-bits wide. Therefore, when a 16-bit transfer is requested, the control circuitry must force two back-to-back 8-bit cycles. When MBCYC switches high, a memory cycle will be initiated by the 82Ø3 dynamic RAM controller, and the BUS HOLD flip-flop will be updated as follows:

BHEN	MAØ	BUS HOLD	Description
Ø	Ø	Ø	8-bit transfer to even byte
Ø	1	Ø	8-bit transfer to odd byte
1	Ø	1	16-bit transfer
1	1	Ø	8-bit transfer to odd byte

When the first memory cycle is started, MBAØ will be equal to the Multibus address line MAØ. MAØ determines whether an odd or even byte of the dual-port memory is to be accessed, and which Multibus data buffer (or latch) to use. At the end of the memory cycle, DTACK will switch high, indicating that the transfer is complete, and will clock the state of the EN MACK/ line into pin 8 of IC62. If BUS HOLD was low, indicating an 8-bit transfer, MACK/ will switch low, which causes the Multibus XACK/ signal to be generated. If BUS HOLD was high, MACK/ will be held off.

The rising edge of DTACK also clocks the state of BUS HOLD into pin 5 of IC4Ø. If BUS HOLD is high, MBAØ is forced high. This will cause the next transfer to be on an odd boundary, using the data buffer connected to the upper eight Multibus data lines.

When DTACK switches high, END MBCYC/ and CLR INT CYC/ both go low, removing the MBCYC or INT CYC signals (if active), and preventing either type of memory request. When the 82Ø3 dynamic RAM controller is ready to process another memory cycle, DTACK will switch low again, releasing the END MBCYC/ and CLR INT CYC/ inputs to the arbiter, allowing it to issue a new cycle request. There are certain cases when either END MBCYC/ or CLR INT CYC/ will not be released when DTACK goes inactive, thus preventing Multibus or 8Ø88 requests, respectively.

The END MBCYC/ and CLR INT CYC/ signals are generated by PAL3 as follows:

```
END MBCYC = INT CYC
          + INT LOCK
          + DTACK
          + MBREQ/ * MBCYC/
          + RESET
```

```
CLR INT CYC = DTACK
            + MBCYC
            + BLOCK
            + BUS HOLD
            + RESET
```

In the case of a 16-bit transfer, BUS HOLD will be true after the first cycle has been initiated. Therefore, when DTACK switches low after the first cycle has been completed, pin 9 of IC25 will be held low because of BLOCK/, allowing the second Multibus cycle to be started. When MBCYC again switches high, the second 8-bit memory cycle (to an odd boundary) will be initiated, and BUS HOLD will switch low. At the end of the second memory cycle, EN MACK/ will be high, and MACK/ will switch low at the rising edge of DTACK, generating the Multibus XACK/ signal. The PAL definitions for EN MACK/ and CLR REQ/ are shown below:

```
EN MACK = BUS HOLD
        + MBCYC/

CLR REQ = DTACK * BUS HOLD/ * MBCYC
        + MEM AEN/
```

The BLOCK/ signal, when active, locks the 8088 out of the dual-port RAM. BLOCK/ is generated by PAL3 as follows:

```
BLOCK = MBREQ * MBCYC * NO LOCK/
      + BLOCK * LOCK
      + BLOCK *MBREQ
```

In order for BLOCK/ to initially switch high, a Multibus cycle must have been requested and initiated, and the 8088 must have previously set the NO LOCK/ signal high to permit the Multibus to lock the dual-port RAM. Once BLOCK/ is active, it will remain active (thus holding off internal cycles) until LOCK goes false.

The RAM array consists of nine 64K dynamic RAMs completely bussed except for the data lines. Eight of the RAMs contain data information (IC85-IC92), while IC93 contains a parity bit. An 8203 dynamic RAM controller (IC70) is used to Multiplex the address lines as needed by the RAMs and generate the refresh cycles when needed. Normal read and write cycles are started with the ST RD CYC/ and ST WR CYC/ lines. Completion of a cycle is indicated when DP XACK/ goes low. The controller takes care of all timing associated with RAS/ and CAS/, and does all of the address switching necessary.

In order to switch the address and data busses between the 8088 and the Multibus, several buffers and multiplexers are used. Four 74LS157 2:1 Multiplexers (IC56, IC58, IC67, and IC69) are used to switch the address lines. When INT CYC is high, an 8088 dual-port RAM cycle is in progress and the internal address lines will be selected. Otherwise, INT CYC will be low, selecting the Multibus address lines.

The data buffers and latches are controlled by the following signals, generated by PAL2 and PAL3 as follows:

PAL2:

```
  EN INT BUF = INT CYC * WR * IACK/
             + RD * IACK

      EN MB0 = BHEN/ * MWRC * MBCYC
             + MBA0/ * MWRC * MBCYC * BHEN

      EN MB1 = MBA0 * MWRC * MBCYC * BHEN

     SEL ML0 = MBA0 * BHEN
             + CAS/
             + MBCYC/

     SEL ML1 = MBA0/
             + BHEN/
             + CAS/
             + MBCYC/

  RD INT BUF = RD
             + IACK
             + WR/
```

PAL3:

```
        EN ML = MRDC * MEM AEN
```

The 8088 data path consists of a 74LS648 bi-directional latched buffer (IC61). The Multibus data path uses two 74LS244 data buffers and two 74LS373 latches.

The EN INT BUF/ signal enables the 74LS648 bi-directional latched buffer during 8088 dual-port RAM cycles and the RD INT BUF/ signal determines the direction of data flow through the device.

During Multibus cycles the 74LS244 tri-state buffers are used during write operations, while the 74LS373 tri-state latches are used for read operations.

The parity error detection circuitry consists of a 74F280 parity generator/checker (IC84), a 64Kx1 dynamic RAM (IC93), a 2-input exclusive-NOR gate (IC52), a 2-input OR gate (IC44), and a flip-flop (IC51). During a write operation, the 74F280 generates an even parity bit (PAR IN), which is stored in the parity data RAM. During a read operation, the stored parity data bit is compared to the parity output bit of the 74F280, and the result is clocked into the PAR ERR flip-flop. If the two are not equal, the data input to the flip-flop will be low, which will set the PAR ERR signal. The DIS PAR INT/ signal will hold the flip-flop in the "no error" condition, if active. If the parity error detection circuitry is to be enabled, the 8088 must set the DIS PAR INT/ signal high.

The TST PAR jumper, when removed, opens the data path to the D0 RAM (IC85), thus forcing parity errors. This allows the operator to test the parity error detection circuitry. For normal operation, the TST PAR jumper must be installed.

Multibus Interrupt Circuitry

Sheet 4 of the schematics contains the Multibus interrupt circuitry. Both bus-vectored and non-bus-vectored interrupts are supported, with the state of the BVI line determining which mode is used.

When either type of interrupt is to be generated, the 8088 writes to the interrupt latch (IC53). The data input lines of IC53 are connected to address lines BA0-BA3, so it is the actual I/O address that gets stored in IC53 rather than the data written. Depending on whether bus-vectored interrupt (BVI) or non-bus-vectored interrupt (NBVI) mode is selected, the I/O address written to the WR INT/ port will have the following effects:

Address	Interrupt Level	Result BVI Mode	Result NVI Mode
0H	0	No Effect	Int 0 Off
1H	1	No Effect	Int 1 Off
2H	2	No Effect	Int 2 Off
3H	3	No Effect	Int 3 Off
4H	4	No Effect	Int 4 Off
5H	5	No Effect	Int 5 Off
6H	6	No Effect	Int 6 Off
7H	7	No Effect	Int 7 Off

8H	Ø	Int Ø On	Int Ø On
9H	1	Int 1 On	Int 1 On
AH	2	Int 2 On	Int 2 On
BH	3	Int 3 On	Int 3 On
CH	4	Int 4 On	Int 4 On
DH	5	Int 5 On	Int 5 On
EH	6	Int 6 On	Int 6 On
FH	7	Int 7 On	Int 7 On

When the WR INT/ line switches low, the NEW INT/ line (PAL1) switches low to indicate that a new or another interrupt request level has been written to the port.

In the NBVI mode, EN INT/ will switch low when NEW INT/ is low and WR INT/ is removed. EN INT/ is clocked up by two flip-flops (IC35) to produce the CH INT/ signal. CH INT/ latches the I8, I9, I1Ø, and INT RQ lines into IC64, and clears pin 9 of IC35. IC64 is a 74LS259 bit settable latch. When enabled, the input lines I8-I1Ø select one of the outputs, while the state of the INT RQ line determines if the output is turned on or off. The outputs of IC64 drive 74Ø6 open-collector inverters which drive the Multibus interrupt lines INTØ/-INT7/.

In the BVI mode, EN INT/ will be held off if a bus vectored interrupt acknowledge cycle is in progress (When either INTA1/ or INTA3/ is active). Otherwise, it will be processed in the exact manner as when in NBVI mode. When the host receives the interrupt request, and is ready to process the interrupt, it will enter an interrupt achnowledge cycle.

The interrupt acknowledge cycle consists of three distinct phases defined by the INTA1/-INTA3/ outputs of PAL1, as follows:

```
INTA1 = BINTA
      + INTA1 * XACKIN/ * RESET/

INTA2 = INTA1 * BINTA/
      + INTA2 * XACKIN/ * RESET/

INTA3 = INTA2 * BINTA
      + INTA3 * BINTA * RESET/
```

In the first phase, the Multibus INTA/ line is driven low by the host and inverted to form BINTA. When BINTA switches high, INTA1/ will switch low to prevent further interrupts from being issued to the Multibus by the B1Ø31.

In the second phase, BINTA will switch low causing INTA2/ to switch low.

In the third phase, BINTA will switch high causing INTA3/ to switch low. At this point, the host provides a 3-bit code on address lines A8-A1Ø, and waits for an 8-bit vector on the Multibus data lines DØ-D7. When INTA3/ switches low, the Multibus buffered address lines MA8-MA1Ø are compared with the 3-bit interrupt level code (I8-I1Ø) to determine if this board is being acknowledged. If the two 3-bit codes match, EQUAL will be high, EN VECTOR/ will be driven low, and the interrupt vector latch (IC57) will be enabled, driving the interrupt vector onto the lower 8 Multibus data lines.

The ACKOUT/ signal will also switch low at this time, which will cause the XACK/ signal to be generated on the Multibus. When XACK/ switches low, XACKIN will switch high, removing the INTA1/ and INTA2/ signals and causing CLR INT/ to switch low, which resets IC53 and IC64. The PAL Definitions for the interrupt related signals are shown below:

```
EN VECTOR = INTA3 * INTRQ * BVI * EQUAL * NEW INT/
          + EN VECTOR * CLR INT/

  CLR INT = EN VECTOR * BINTA/
          + RESET

   EN INT = NEW INT * RESET/ * BVI/ * WR INT/
          + NEW INT * RESET/ * INTA1/ * INTA3/ * WR INT/

  NEW INT = WR INT
          + NEW INT * CH INT/ * RESET/

  ACK OUT = MRDIO
          + MWRIO
          + EN VECTOR * BINTA
```

It should be noted that prior to the 8Ø88 initiating a BVI interrupt sequence, a vector must be written into the interrupt vector latch. Once the host has accepted the interrupt vector, it completes the process by removing the second INTA/ pulse. This causes INTA3/ and EN VECTOR/ to switch high, terminating the BVI sequence.

USART Section

Sheet 5 of the schematics shows the actual interface to the external devices. Note that this sheet is repeated eight times on

the board, with the IC numbers listed for ports Ø-7, in that order. Also, the signal CSX is referenced with the number Ø-7 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 2661 are provided in the Signetics 2661 data sheet. This data sheet is reprinted later in this manual.

Note that there is a CTS strap selection available for each USART. This strap 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 CTS INT 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 CTS EXT strap in place, the external device must drive CTS in order for the board to operate properly.

Since the USART will not receive characters unless the DCD/ signal is low, a pull-up resistor is placed on the receiver input for that signal to guarantee DCD is low if the attached device does not drive it.

The USARTs will generate an interrupt to the 8Ø88 on the occurrence of any receiver full condition. The transmitter empty condition is not used to generate an interrupt because the 8Ø88 would be swamped with interrupts if that were the case. This way only important interrupts (where lost data could occur) get to the 8Ø88.

All of the RS-232 signals from the external connector are buffered by 1488s and 1489s. Note tha capacitors can be added to slow the rise and fall times of the RxD input and TxD output. Normally, however, these additional capacitors are not needed.

Connection to the DB-25 connectors is done using two 6Ø-pin ribbon cables. Each cable has the signals related to four ports, with each port using 15 contiguous conductors of the cable. These 15 pins then connect to pins 1-8 and 14-2Ø of the DB-25 connector. These pins are all that are required for the seven RS-232 signals that the board supports.

Six straps are provided on the board for each port to determine the direction of the data flow to the connector. A ribbon cable is normally used to connect a DB-25 connector to the board, and the direction (to or from the board) of each signal must be se-

lected based on how the device being attached drives the cable. The sets of jumper options are marked TERM and CPU, and strapping the board in either mode will connect the following signals to the indicated pins of the DB-25:

Input Signal	Term Pin	CPU Pin
RxD	3	2
DSR	5	20
CTS	5	4

Output Signal	Term Pin	CPU Pin
TxD	2	3
DTR	20	6
TRS	4	5

The DCD signal always appears at pin 8 of the DB-25.

4. Standard High-Performance Driver

The following gives details on the Standard High-Performance Driver firmware which is normally shipped with the board.

This driver partitions the 64K dual-port RAM into eight 8K blocks. One block is used for each port and the organization of each block is as follows:

Offset	Length	Description
Ø	3Ø56	Output Buffer Ø
BFØH	3Ø56	Output Buffer 1
17EØH	1Ø24	Input Buffer
1BEØH	1Ø24	Error Buffer
1FE1H	1	Command Code
1FE2H	2	Command Data
1FE5H	1	Command Status
1FE7H	1	Status From USART
1FE8H	2	Input Filling Pointer
1FEAH	2	Input Emptying Pointer
1FECH	2	Receive Buffer Overflow Count
1FEFH	1	Interrupt Source Flag
1FF1H	1	Output Busy
1FF3H	1	Output Stopped
1FF5H	1	Parity Error
1FFCH	2	Firmware Revision Number
1FFFH	1	Port Interrupt Status Flags

The Output Buffers are loaded by the host with characters to be sent out of the board. Up to 3Ø56 characters can be loaded into each Output Buffer for transmission by a single Send Block command. The use of two Output Buffers allows the host to be filling one as the other one is being transmitted.

There are two buffers associated with input characters, the Input Buffer and the Error Buffer. Each buffer is 1Ø24 bytes long, with the Input Buffer holding the actual input data characters, and the Error Buffer holding the USART status associated with each character. Bits within this status byte will then indicate if the character was received with a parity error, overrun error, or framing error. Note that the receipt of a break character is signaled by a ØØH character with a framing error.

The Input Filling Pointer and Input Emptying Pointer are used to allow the on-board 8Ø88 processor and the host to know the status

of the input buffers. Both of these pointers are cleared for a port by the board when a Set USART Mode Registers command is received by the board for the port. After that, the board increments the Filling Pointer after each character and its associated error status are stored into the buffers. Likewise, the host updates the Emptying Pointer to equal the next offset to be read by the host after it takes characters from the buffer. If an input character is ever received when the Filling Pointer is equal to the Emptying Pointer-1, the character is discarded and the Received Buffer Overflow Count is incremented. Note that both pointers are modulo-1Ø24, and their values should never exceed 3FFH. Note that the host should access all 16-bit pointers in word mode. If byte accesses are to be made to 16-bit pointers, the host must lock to bus to avoid errors.

The input interrupt scheme to the host is very flexible. This allows the system to be tuned to minimize interrupts and yet guarantees fast user response time and no loss of data. When a port is first initialized, the Input Delay Time is set to zero. If this delay time is changed the board will wait for the specified delay time before interrupting upon receipt of a character. The amount of time it waits is somewhat dependent on the loading of the board. The delay time is best chosen empirically. If any subsequent characters come in after the delay counter has started, the counter is not reset, and an interrupt will still occur based on the delay from the first character received.

Four things can happen to cause an immediate input interrupt after receiving a character, assuming input interrupts have been enabled. First, if the delay counter is disabled, an immediate input interrupt will occur. Second, if storing the character causes the Input Buffer to exceed the Input High Water Mark value, an input interrupt is generated. The third condition is when a character is received and any of the error status bits (bits 3-5) match either the Interrupt on Status Bits Ø/1 Masks. Finally, if the character (after being masked with the Immediate Input Interrupt Mask) is found to be less than the Immediate Input Interrupt Value, an immediate input interrupt occurs. This allows the board to interrupt immediately for any control characters.

Three words of the control block are reserved for command related information. The Command Code holds the opcode of any command to be performed. These command codes are listed below. The Command Data bytes hold any parameters needed by the individual commands. Finally, the Command Status byte is set to FFH by the host before a command is executed, and the board clears it if the

command is successfully initiated, or sets it to a Ø1H if the command was invalid. Note that the command data is stored MSB first (in the even byte), and LSB second (in the odd byte).

The Status From USART byte is updated periodically by reading the USART Status. This can give the host information regarding various modem control inputs from the port.

The Interrupt Source Flag indicates for an individual port the interrupt service request conditions for the port. bØ is set for an output done interrupt, bl is set for an input interrupt, b2 is set for a status change interrupt, and b3 is set for a parity error interrupt. All other bits are always zero. If no bits are set, then the port is not requesting interrupt servicing, and the bit corresponding to the port in the Port Interrupt Status Flags (described below) will be zero.

The Output Busy flag can be used if the host is doing output in polled (non-interrupt) mode. This flag is set to FFH by the board when the current output operation is initiated, and cleared when the operation is completed. The host can poll the Output Busy flag until it changes to Ø before issuing another Send Block command.

The Output Stopped flag indicates that output is halted due to the receipt of an XOFF character (when XON/XOFF mode is enabled for output on the channel). This allows the host to see if a line is hung due to this condition.

The Parity Error flag is set to FFH on all eight channels and the status LEDs are set to EH if a parity error is detected on the dual-port RAM. An interrupt can also be generated to alert the host.

The last two flags are available for port Ø only. These flags are board-level functions and are not dependent on individual ports. The Firmware Revision Number will equal the revision level printed on the label of the PROM. For example, Vl.ØØl firmware will be labeled as such on the PROM, and the value at location lFFCH of port Ø will be 3E9H (1ØØlD). The Port Interrupt Status Flags indicate whether the board is currently generating an interrupt to the bus, and if so which ports have interrupt service requests. This is a one-byte flag with one bit for each port. bØ will be 1 if port Ø is interrupting, bl for port 1, etc. If all the bits are low, the board is not currently generating an interrupt to the bus.

Most of the available commands affect an individual port. The

few interrupt related commands that are universal may be issued to any port with the same effect. All commands are issued by writing the Command Code and Command Data (if required) into the appropriate dual-port location as shown above. Dissimilar commands may be issued to each port's dual-port RAM, and all eight ports may be reprogrammed at once, if desired. Once the dual-port RAM has been properly setup, a byte must be written to the I/O port to indicate which of the eight ports have new commands pending. Each bit of the byte written to the I/O port represents a specific port, with bØ representing port Ø, and b7 representing port 7. Ports that have commands are represented with a set bit, while ports that do not that have commands are represented with a cleared bit.

The 19 commands which the host may issue to the controller are shown below:

Command Code	Description	Command Data High (1FE2)	Command Data Low (1FE3)
ØØH	Set USART Mode Registers	Mode Reg 1	Mode Reg 2
Ø1H	Set USART Command Register		Command Reg
Ø2H	Set XON/XOFF Mask/Enables	Mask	Enables
Ø3H	Set XON/XOFF Characters	XON	XOFF
Ø4H	Send Block Ø	Length-MSB	Length-LSB
Ø5H	Send Block 1	Length-MSB	Length-LSB
Ø6H	Send Immediate Character		Character
Ø7H	Send Break		
Ø8H	Set Input Delay Time	Delay-MSB	Delay-LSB
Ø9H	Set Input High Water Mark	Value-MSB	Value-LSB
ØAH	Set Immediate Input Interrupt	Mask	Value
ØBH	Set Int on Status Bits Ø Mask		Mask
ØCH	Set Int on Status Bits 1 Mask		Mask
ØDH	Set Interrupt--Inputs	Level	Vector
ØEH	Set Interrupt--Outputs	Level	Vector
ØFH	Set Interrupt--Status	Level	Vector
1ØH	Set Interrupt--Parity Error	Level	Vector
11H	Clear Interrupt		
12H	Allow Multibus Lock Operation		

The first two commands allow complete control of the USART. The first command also resets certain parameters for each port. The parameters are setup as follows:

XON/XOFF Mask	Ø
XON/XOFF Enables	Ø
XON/XOFF Characters (both)	Ø

Input Delay Time	Ø
Input High Water Mark	Ø
Immediate Input Interrupt Mask	Ø
Immediate Input Interrupt Value	Ø
Interrupt on Status Bits Ø Mask	Ø
Interrupt on Status Bits 1 Mask	Ø
Input Filling Pointer	Ø
Input Emptying Pointer	Ø
Receive Buffer Overflow Count	Ø
Output Busy	Ø
Output Stopped	Ø

Complete information on the bits associated with Mode Register 1, Mode Register 2, and the Command Register can be found in the specification sheet for the 2661 USART. The board requires the bits associated with internal/external baud rate generator to be set to internal and the asynchronous/synchronous mode to be set to asynchronous mode. If these states are improperly selected in the Command Data field, an invalid command indication will be generated, and the USART will not be programmed.

XON/XOFF mode allows flow control for the board. Any device that allows this mode can stop an input stream by transmitting an XOFF character and re-enable an input stream by transmitting an XON character. The board has a flexible method for selection of XON/XOFF characters and supports flow control in either or both directions.

The selection of the XON/OFF character is made by a mask and character values. For input characters received by the board, the mask value is ANDed with the input character before comparing it to the XON/XOFF characters for the port. This allows the maximum in flexibility. Further, two bits enable the XON/XOFF mode for the board. bØ controls XON/XOFF mode for characters being sent from the board. If this bit is set, the board will stop transmitting an output block after the receipt of an XOFF character (properly masked), and will resume transmission (at the point where it was stopped) when an XON character is received. If bl is set, the board will transmit an XOFF character if the Input Buffer reaches the input high water mark. The board will transmit an XON character when it determines the Input Buffer is equal to 1/2 the value of the input high water mark. This will resume the flow of input characters from the external device.

b2 of the Enables byte allows flow control in a different way--by using the RTS signal out of the board. Using the same parameters as XON/XOFF flow control uses, the RTS signal is made inactive to stop input characters, and re-activated when the input buffer

empties. Only one mode of flow control should be enabled for each channel (either XON/XOFF or RTS, but not both).

The Send Immediate Character and Send Break commands allow the host to interrupt the normal output stream to send an immediate value. Neither of these events will affect the Output Busy flag in dual-port RAM or cause an output done interrupt. If a request to send an immediate character or a break character is received with a Send Immediate Character or Send Break Command pending, the command status byte is set to 01H (Illegal Command), and the new request is ignored.

The Input Delay Time and Input High Water Mark values were described above, and can be set to optimize system performance. If the input high water mark is set to greater than 1023, the input XON/XOFF feature will be disabled and normal input interrupts will be determined entirely by the delay counter and compare value. The input high water mark is meant to be used as a "safety valve" for the delay counter and it is therefore not recommended to set it above 1023. The Immediate Input Interrupt Mask and Value allow each character to be ANDed with the mask and compared to the value. If the input character (as masked) is less than the value, an immediate interrupt will happen. Often, for terminal control applications where a delay time is set, the mask is set to 7FH, and the value is set to 20H. This causes all control characters to immediately interrupt.

The Interrupt on Status Bits 0/1 Masks allow the host to be interrupted when certain USART status bits change states. For each bit that is set in the mask, the corresponding bit of the status word is periodically checked for a 0/1 (as appropriate). If any selected bit is found to be in the proper state, a status interrupt is generated by the controller. Note that the three error bits associated with input characters are not checked using these masks, except when input characters are received. If a match occurs on those three bits during input status bit checking, an input interrupt is generated. By clearing the Interrupt on Status Bits 1 Mask bits associated with any of the input error conditions, the program can defeat the immediate interrupt for input for that particular error. The error status will, however, be stored into the Error Buffer regardless of the setting of the mask bits.

The last six commands define functions that are not associated with a particular port. Four of the commands allow interrupt levels and vectors to be setup for the four general types of interrupts that the board has. If b7 of the level is set, bus-vectored interrupts are enabled. If that bit is cleared, non-

bus-vectored mode is selected. The interrupt level itself is given in b2-bØ of the field, with b3 being set if an interrupt is to be generated. If b3 is not set, interrupts are disabled. The vector is only used when bus-vectored mode has been enabled. It will determine the vector given to the host when the interrupt acknowledge cycle occurs on the Multibus. The Clear Interrupt command tells the board that the current interrupt has been serviced by the host, and causes the board to remove the interrupt and clear all bytes in the dual-port RAM relating to interrupts. If another interrupt is pending when the Clear Interrupt command is received, the interrupt status bytes in the dual-port RAM will be changed to reflect the pending interrupt and a new interrupt will be issued immediately. For this reason, after receiving an interrupt from the board, the host must service the interrupt completely before issuing the Clear Interrupt command.

After a parity error has been detected, the parity error detection circuitry is permanently disabled, preventing any further interrupts due to parity errors. When the host issues the Clear Interrupt command after a parity error interrupt, the interrupt is removed and the port interrupt status and interrupt source flags are cleared. However, the error status LEDs and the parity error bytes in the dual-port RAM are not reset, indicating that a parity error has occurred and the board should be serviced. It is not recommended that the host continue to operate after a parity error has occurred, but the driver will not prohibit further operation.

Finally, since allowing Multibus lock operations (as needed by 8-bit masters accessing the board's 16-bit pointers) degrades the operation of the board (even for non-locked accesses), the user must specifically enable it if necessary. Note that any 16-bit master accessing the board should not need to use the Multibus lock facility (if it accesses 16-bit pointers in word mode).

5. Diagnostics

Each time power is applied to the board or the Multibus INIT/ line is activated, the board enters into its power-on reset sequence, as follows:

1. bØ of the host Status Port is cleared (indicating busy status), and the Error Status LEDs are all turned on.

2. The diagnostic tests are executed. If a fatal error is discovered, the proper Error Code is displayed on the Error Status LEDs and operation is halted.

3. If no errors were found, bØ of the host Status Port is set (indicating ready status), indicating to the host that the self tests passed, and the board may be initialized by the host.

4. The 4-bit code on the Error Status LEDs is changed between ØH and 1H and the test is re-executed (repeat steps 2 thru 4).

After the first pass of the diagnostics, the Status Port will indicate that the board is ready to accept commands. The board will continue diagnostic testing until the host issues an initialization I/O write to the board's I/O port. Each pass of the diagnostic tests takes approximately 4 seconds. Therefore, the Error Status LEDs will alternate between ØH and 1H every 4 seconds until the host issues an I/O write to the board. This first I/O command written to the board is not executed, and can be any value. Instead, it sets the host Status Port bØ low (busy), terminates the self test mode, and initializes the board for normal operations. When the board is ready, it will permanently set the Error Status LEDs to ØH, and set the host Status Port bØ high, indicating that it is initialized and may be programmed by the host.

Note that during the diagnostic testing, the dual-port RAM is tested. Therefore, the host must issue the Initialization I/O Write and wait for the Status Port to go ready, before writing to the dual-port RAM.

PROM Sumcheck

This test sums the value in all bytes of the EPROM and compares the sum to zero. A two's complement byte is loaded into the last

location of the EPROM to force the sum to zero. This insures the PROM is readable and the program is correct. Error Code--2H.

Scratchpad RAM

A memory test is performed on the scratchpad RAM. The RAM is tested as four blocks of 2K bytes. This causes a 2Kx8 RAM chip to be tested four times or an 8Kx8 RAM chip to be tested once. For each block, an incrementing pattern of data (skipping zero) is written to and verified at every byte. Each new block starts with a different initial data byte. Error Code--3H.

Dual-Port RAM

A memory test is performed on the 64K bytes of dual-port RAM. An incrementing pattern of data (skipping zero) is written to and verified at every byte. This is repeated four times with a new initial byte each time. Error Code--4H.

Bus Lock

This tests the bus lock internally only. The bus lock is disabled, enabled, and disabled again. Each time the state is changed, it is verified by reading an internal status bit. Error Code--5H.

USART Tests

The next two steps use the same error codes. If any errors occur, a non-fatal error code is issued, and the board is still deemed partially functional (Error Code--8H). If all eight USARTs exhibit failures from either of these tests, a fatal error will be issued (Error Code--6H).

USART Address Decode

This test writes non-identical bytes to the command registers of all eight USARTs. It then reads the command registers back to verify that the data read is the same as what was written.

USART Data Turnaround

All 8 USARTs are programmed for local loopback mode. Two bytes are transmitted and received for each USART. The data is verified as being the same as that sent. The status registers are checked for each character to verify no USART errors.

Receiver Interrupts

A single USART is run in local loopback mode with the receiver interrupt enabled. The time required to interrupt is verified to be within limits, and the character received is verified. Error Code--7H.

Non-Fatal Errors

A non-fatal error occurs when from one to seven USARTs fail during the diagnostics. Since the host can still gain control of the board, information on the error is left in the dual-port memory.

A 16 byte error map is provided at the beginning of the dual-port RAM for the host to inspect. The first eight bytes (the only ones of significance to the host) detail problems with any of the eight USARTs which do not cause a fatal error. This memory area should be read after the status bit for the board goes active for the second time, indicating that the board completed its self tests with no fatal errors.

For each USART, one byte is used as its error bit map. The address of the byte corresponds to the USART number (i.e. byte Ø for USART Ø, etc.). The byte will be zero if there were no errors for that USART. A non-zero value indicates some type of USART error. Below is a table of the type of error indicated when each bit is set.

Bit	Error Type
Ø	USART Addressing Error
1	USART Timeout Error
2	USART Data Error
3	USART Found Parity Error
4	USART Found Overrun Error
5	USART Found Framing Error
6	Not Used
7	Not Used

From address 1ØH to the end of dual-port RAM, a test pattern is available for the host to check. It should consist of 14H at location 1ØH, and the data value should increment for each successive data location, skipping the data value of ØØH. Note that for each port, the last 32 bytes (offsets 1FEØH-1FFFH) are preset by the controller during initialization, thus disrupting the pattern at those memory locations.

6. Installation/User Selectable Options

The High Performance Intelligent 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 two 8-position dip-switches to select the I/O port address it will respond to. Each position of the switch corresponds to one address line, from AØ to A15. As marked on the board, A15 is selected by the left-most switch at location IC65, while A8 is selected by the right-most. AØ is selected by the left-most switch at position IC55, while A7 is selected by the right-most. Each address line is compared for Ø 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 EXT. For systems where only 8-bit I/O addressing is used, this shorting plug should be placed over the pins labeled 8-BIT. Also, for 8-bit systems, the upper eight address switches are not used.

The address range for the dual-port RAM is selected with an 8-position dip-switch at location IC75. This allows the user to set a comparison for address lines A23 through A16, dividing the system into 64K sections. The switch corresponding to A23 is on the top, and the switch corresponding to A16 is on the bottom. Any switch position turned on will be compared for a 1, while any position left open will compare for a Ø.

PROM Size Selection

A 3-pin jumper array is provided to select between 2764, 27128 and 27256 EPROMs. The array is located beneath IC31 and is marked: 64/128 and 256. If 2764 or 27128 EPROMs are used, the pins labeled 64/128 should be jumpered. If 27256 EPROMs are used, the pins labeled 256 should be jumpered.

RAM Size Selection

A 3-pin jumper array is provided to select between 2Kx8 and 8Kx8 static RAMs. The array is located beneath IC32 and is marked 16 and 64. If 2Kx8 (16K) static RAMs are used, the pins labeled 16

should be shorted. If 8Kx8 (64K) static RAMs are used, the pins labeled 64 should be shorted.

Command Interrupt Jumper

The CMD INT jumper pins determine whether response to new commands from the host are interrupt driven or whether a polled operation must be performed. With the CMD INT pins shorted, the 8088 will receive a maskable interrupt each time the host writes to the board's I/O port. With the CMD INT pins open, the 8088 must read its internal status port to determine if the host has written to the I/O port. This strap is completely dependent on the firmware implementation. Both firmware options offered by Central Data use polled host commands, so the pins should not be shorted.

Parity Error Jumper

The PAR ERR jumper is provided to allow the operator to test the parity error detection circuitry. When the PAR ERR jumper is removed, the data path to the D0 RAM is broken to force errors. For normal operation the PAR ERR jumper must be installed.

CTS Selection

Since the USARTs 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 CTS INT 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 CTS EXT 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.

DB-25 Direction Straps

Connection to the DB-25 connectors is done using two 60-pin ribbon cables. Each cable has the signals related to four ports, with each port using 15 contiguous conductors of the cable. These 15 pins then connect to pins 1-8 and 14-20 of the DB-25 connector. These pins are all that are required for the seven RS-232 signals that the board supports.

Six straps are provided on the board for each port to determine the direction of data flow to the connector. A ribbon cable is normally used to connect a DB-25 connector to the board. The direction (to or from the board) of each signal must be selected based on how the device being attached drives the cable. The sets of jumper options are marked TERM and CPU. Strapping the board in either mode will connect the following signals to the indicated pins of the DB-25.

Input Signal	Term Pin	CPU Pin
RxD	3	2
DSR	6	20
CTS	5	4

Output Signal	Term Pin	CPU Pin
TxD	2	3
DTR	20	6
RTS	4	5

The DCD signal always appears at pin 8 of the DB-25.

7. Specifications

Word Size

Host accesses to the dual-port RAM may be done in 8- or 16-bit mode. If 8-bit mode is utilized, all 16-bit dual-port pointers or values must be accessed in the locked mode.

Addressing

This board has one Multibus I/O port. It can be addressed using full 16-bit I/O addressing. By changing a strap, only 8-bit I/O addressing is used. The base address of the I/O port can be selected on a 1 port boundary within a 256-port or 64K-port addressing range.

The board contains 64K of dual-port RAM, and the base address for the memory can be set to start on any 64K boundary in the system's 16-megabyte address space.

Access Time

Typical byte mode access time is 645ns to 1120ns (865ns to 1415ns, max), while word mode access time is 1050ns to 1525ns (1270ns to 1820ns, max). The variance in access time depends on whether the dual-port RAM is being refreshed or accessed by the 8088 at the beginning of the cycle.

Baud Rates Available

50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, and 19,200.

Interrupt Sources

For the Emulation Mode Driver, all interrupts go to the same non-bus-vectored interrupt level. That level is programmed whenever any port is reset.

For the Standard High-Performance Driver, both bus-vectored and non-bus-vectored interrupts are supported. Interrupts may be generated by new input, output complete, USART status change, and dual-port RAM parity error conditions. A separate interrupt level and interrupt vector (if applicable) may be programmed for each of the four types of interrupts. For each of the four

types of interrupts, however, all eight ports will cause the same interrupt level/vector to be used.

RS-232 Specifications

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

Connection to the DB-25 connectors is done using two 60-pin ribbon cables. Each cable has the signals related to four ports, with each port using 15 contiguous conductors of the cable. These 15 pins then connect to pins 1-8 and 14-20 of the DB-25 connector. These pins are all that are required for the seven RS-232 signals that the board supports.

Six straps are provided on the board for each port to determine the direction of data flow to the connector. A ribbon cable is normally used to connect a DB-25 connector to the board, and the direction (to or from the board) of each signal must be selected based on how the device being attached drives the cable. The sets of jumper options are marked TERM and CPU, and strapping the board in either mode will connect the following signals to the indicated pins of the DB-25:

Input Signal	Term Pin	CPU Pin
RxD	3	2
DSR	6	20
CTS	5	4

Output Signal	Term Pin	CPU Pin
TxD	2	3
DTR	20	6
RTS	4	5

The DCD signal always appears at pin 8 of the DB-25.

Bus Interface

All signals meet the IEEE Mulbibus proposed specification. 796 Bus Compliance: Slave D16 M24 I16 V2L.

Physical Characteristics

Dimensions: See the Multibus dimensions drawing at the end of this section.

Electrical Characteristics

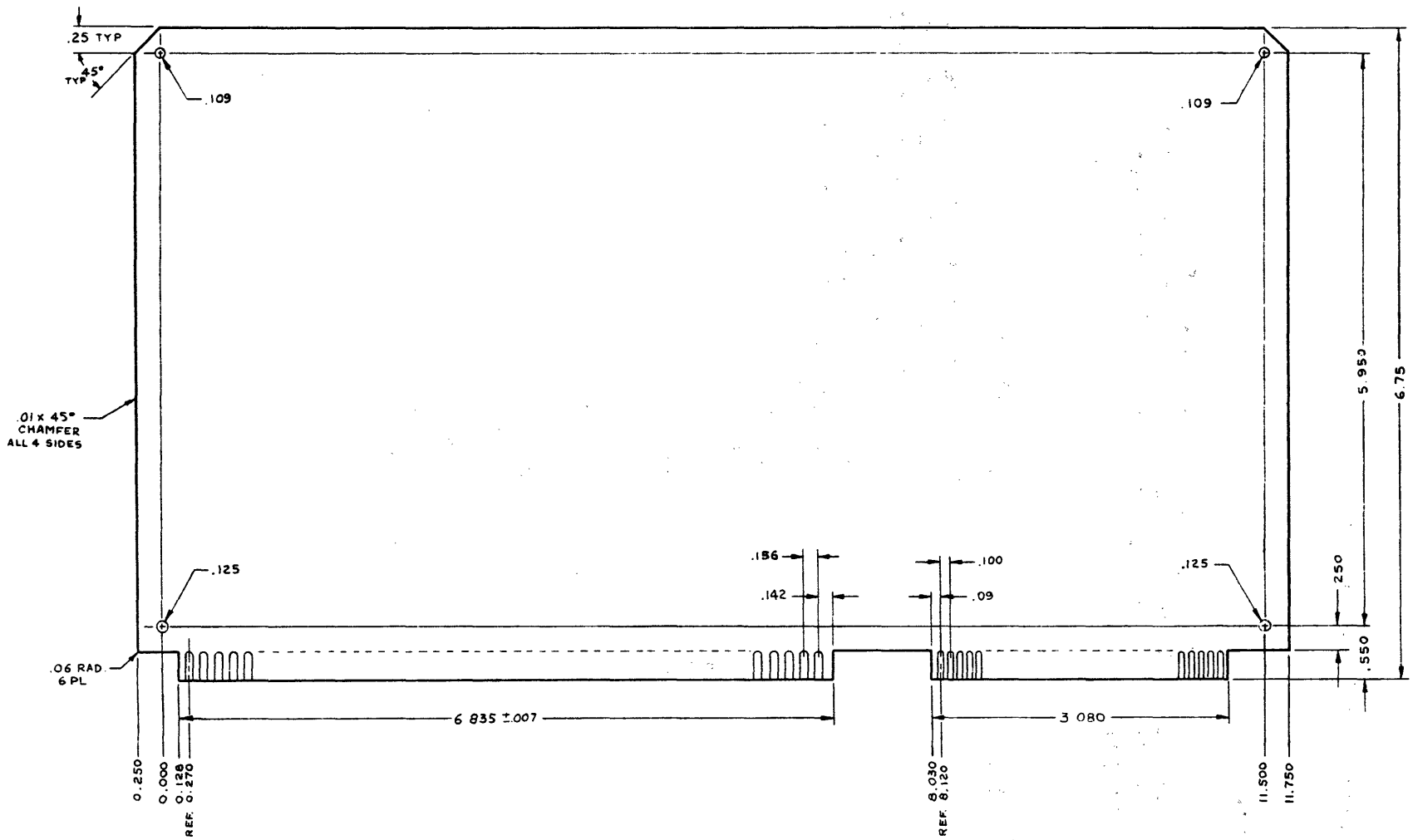
Vcc=+5V ±5%
Vdd=+12V ±5%
Vbb=-12V ±5%
Icc=2.7A typ, 3.9A max
Idd=Ø.2A typ, Ø.3A max
Ibb=Ø.1A typ, Ø.2A max

Environmental Characteristics

Operating Temperature: Ø°C to +55°C
Relative Humidity: Ø to 9Ø% (non-condensing)

Ordering Information

Part Number: B1Ø31
Description: High Performance Intelligent I/O Board
Options: /EM Emulation Mode Driver firmware



Central Data Corporation
1602 Newton Drive, Champaign, IL 61820 (217)359-8010

NO.

SHEET OF

MULTIBUS BOARD DIMENSIONS

8. Programming Information

Memory Map

00000H-01FFFH: Scratchpad RAM and 8088 I/O
30000H-3FFFFH: Unlocked access to the dual-port RAM
70000H-7FFFFH: Locked access to the dual-port RAM
F0000H-FFFFFH: EPROM

I/O Map

Address Range	RD	WR	Description
00-0F		X	Write interrupt level (see chart)
10-1F		X	Write control port (see chart)
20-2F		X	Set command complete flip-flop
30-3F		X	Write interrupt vector
40-4F	X		Read host command
50-5F	X		Status port (see chart)
80-83	X		USART 0
84-87	X		USART 1
88-8B	X		USART 2
8C-8F	X		USART 3
90-93	X		USART 4
94-97	X		USART 5
98-9B	X		USART 6
9C-9F	X		USART 7
A0-A3		X	USART 0
A4-A7		X	USART 1
A8-AB		X	USART 2
AC-AF		X	USART 3
B0-B3		X	USART 4
B4-B7		X	USART 5
B8-BB		X	USART 6
BC-BF		X	USART 7

Interrupt Level Port

I/O Address	Interrupt Level	Result BVI Mode	Result NBVI Mode
ØØH	Ø	No Effect	Int Ø Off
Ø1H	1	No Effect	Int 1 Off
Ø2H	2	No Effect	Int 2 Off
Ø3H	3	No Effect	Int 3 Off
Ø4H	4	No Effect	Int 4 Off
Ø5H	5	No Effect	Int 5 Off
Ø6H	6	No Effect	Int 6 Off
Ø7H	7	No Effect	Int 7 Off
Ø8H	Ø	Int Ø On	Int Ø On
Ø9H	1	Int 1 On	Int 1 On
ØAH	2	Int 2 On	Int 2 On
ØBH	3	Int 3 On	Int 3 On
ØCH	4	Int 4 On	Int 4 On
ØDH	5	Int 5 On	Int 5 On
ØEH	6	Int 6 On	Int 6 On
ØFH	7	Int 7 On	Int 7 On

Control Port

I/O Address	Function
1ØH	Status LED 1 On
11H	Status LED 2 On
12H	Status LED 4 On
13H	Status LED 8 On
14H	Spare
15H	BVI=Ø (NBVI Mode Selected)
16H	NO LOCK/=Ø (Multibus Lock Disabled)
17H	DIS PAR INT/=Ø (Parity Checking Disabled)
18H	Status LED 1 Off
19H	Status LED 2 Off
1AH	Status LED 4 Off
1BH	Status LED 8 Off
1CH	Spare
1DH	BVI=1 (BVI Mode Selected)
1EH	NO LOCK/=1 (Multibus Lock Allowed)
1FH	DIS PAR INT/=1 (Parity Checking Enabled)

Status Port

Data Bit	Definition
bØ-b3	Not Used
b4	1=Internal Lock Active
b5	1=Multibus Interrupt Pending
b6	1=New Command Ready
b7	1=UP Interrupt Pending

8Ø88 Interrupt Vector Table (In Scratchpad RAM)

Address Range	Interrupt #	Type of Interrupt
ØH-3H	Ø	Divide Error
4H-7H	1	Single Step
8H-BH	2	NMI (Parity Error)
CH-FH	3	Interrupt Instruction
1ØH-13H	4	Overflow
14H-17H	5	External Maskable Interrupt (USART or Host Command Interrupt)

RAM locations 18H to 7FFH (2Kx8) or 1FFFH (8Kx8) are available for stack operations and temporary storage.

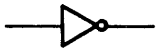
9. Schematics

The following pages contain the schematics for the High Performance Intelligent I/O Board. A full description of the circuitry is given in the Principles of Operation section of this manual.

3 - INPUT 'AND' GATE

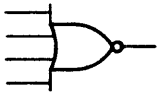
2 - INPUT 'OR' GATE

INVERTER



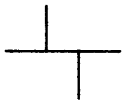
2 - INPUT 'NAND' GATE

4 - INPUT 'NOR' GATE

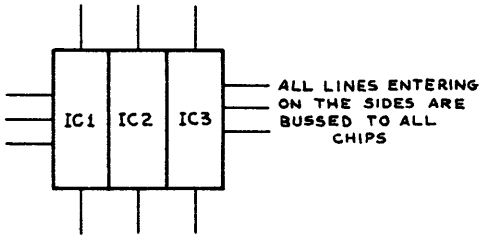


TWO LINES - NO CONNECTION

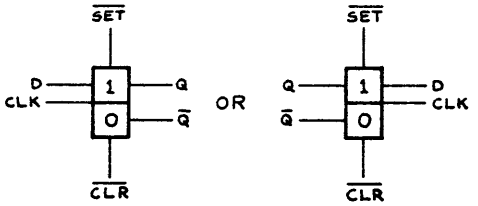
3 LINES - ALL CONNECTED



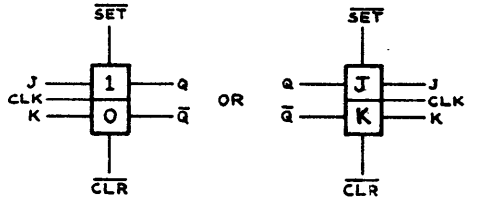
GROUP OF SIMILAR PARTS



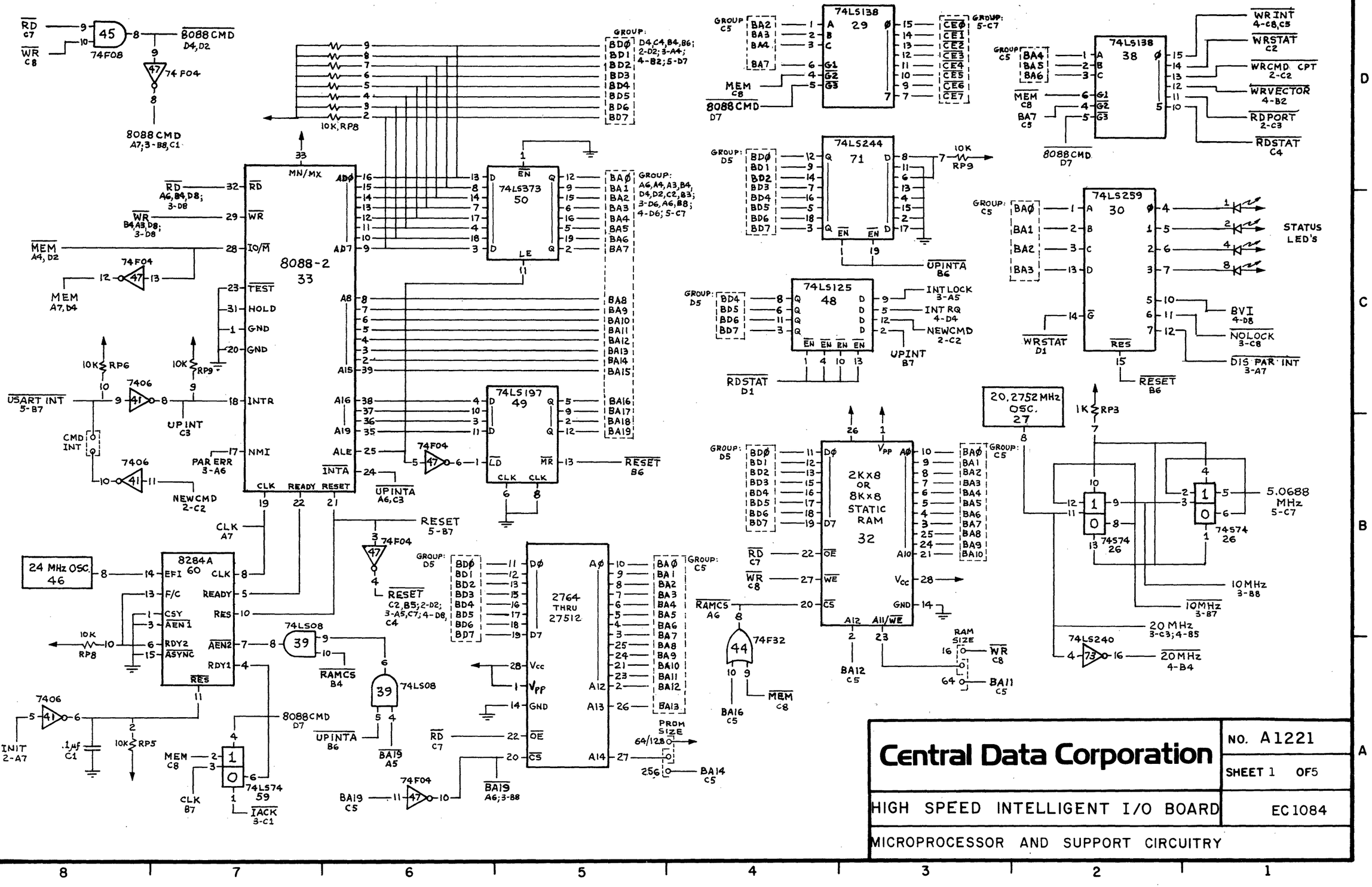
D-TYPE FLIP-FLOP



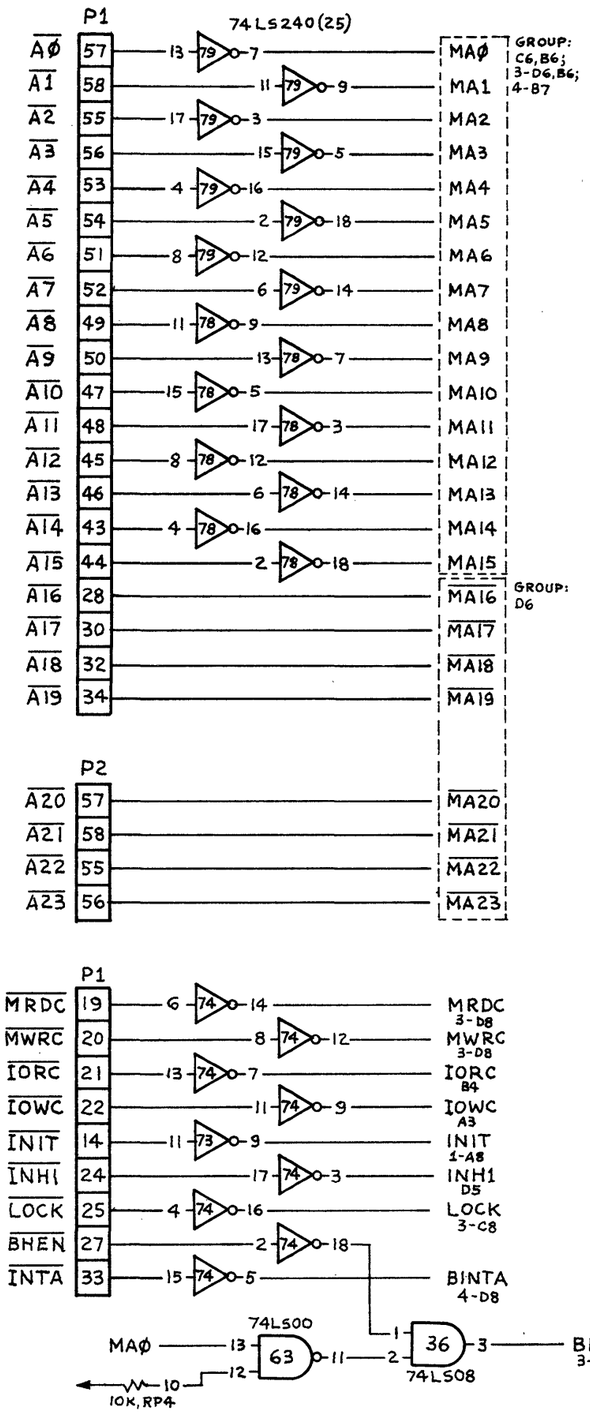
J/K FLIP-FLOP

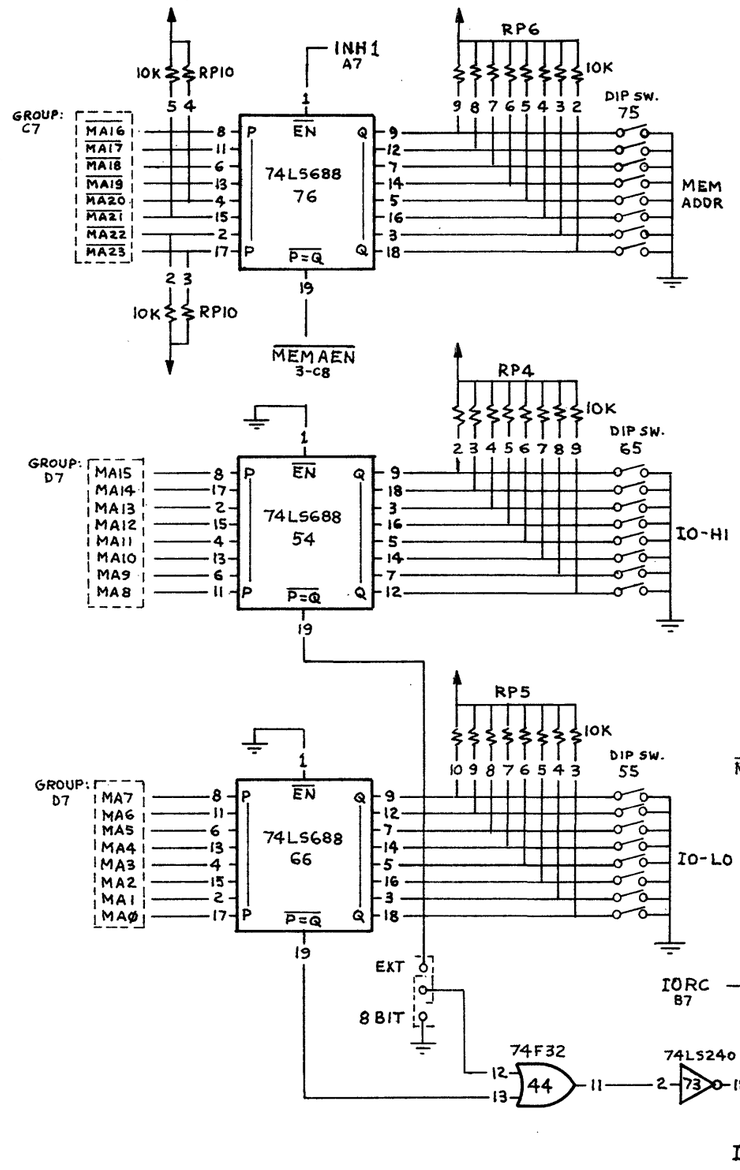


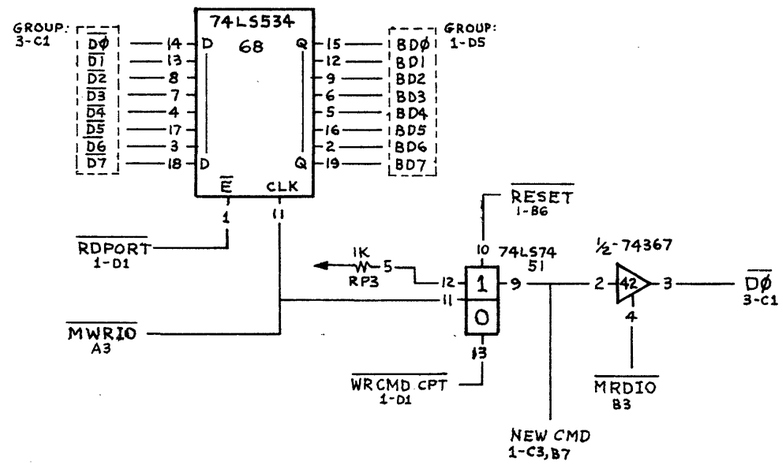
Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010	NO. SHEET OF
DRAWING CONVENTIONS	

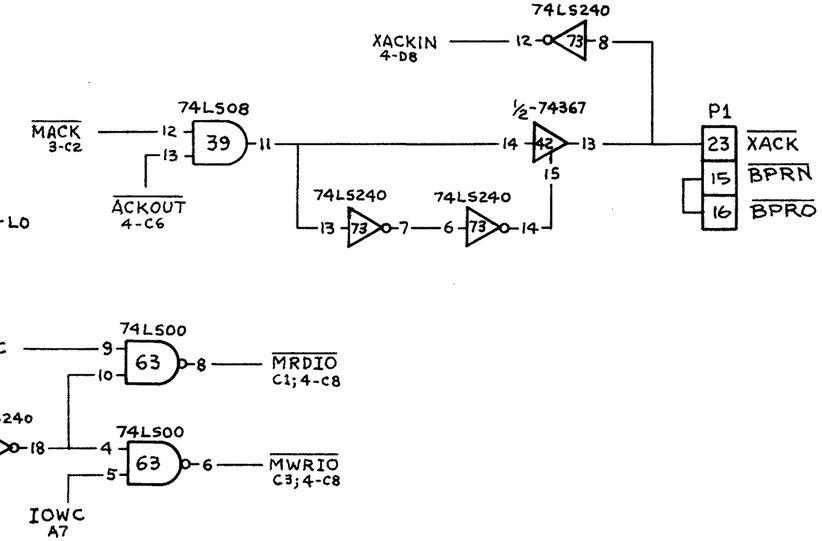


Central Data Corporation	NO. A1221
	SHEET 1 OF5
HIGH SPEED INTELLIGENT I/O BOARD	EC1084
MICROPROCESSOR AND SUPPORT CIRCUITRY	









Central Data Corporation

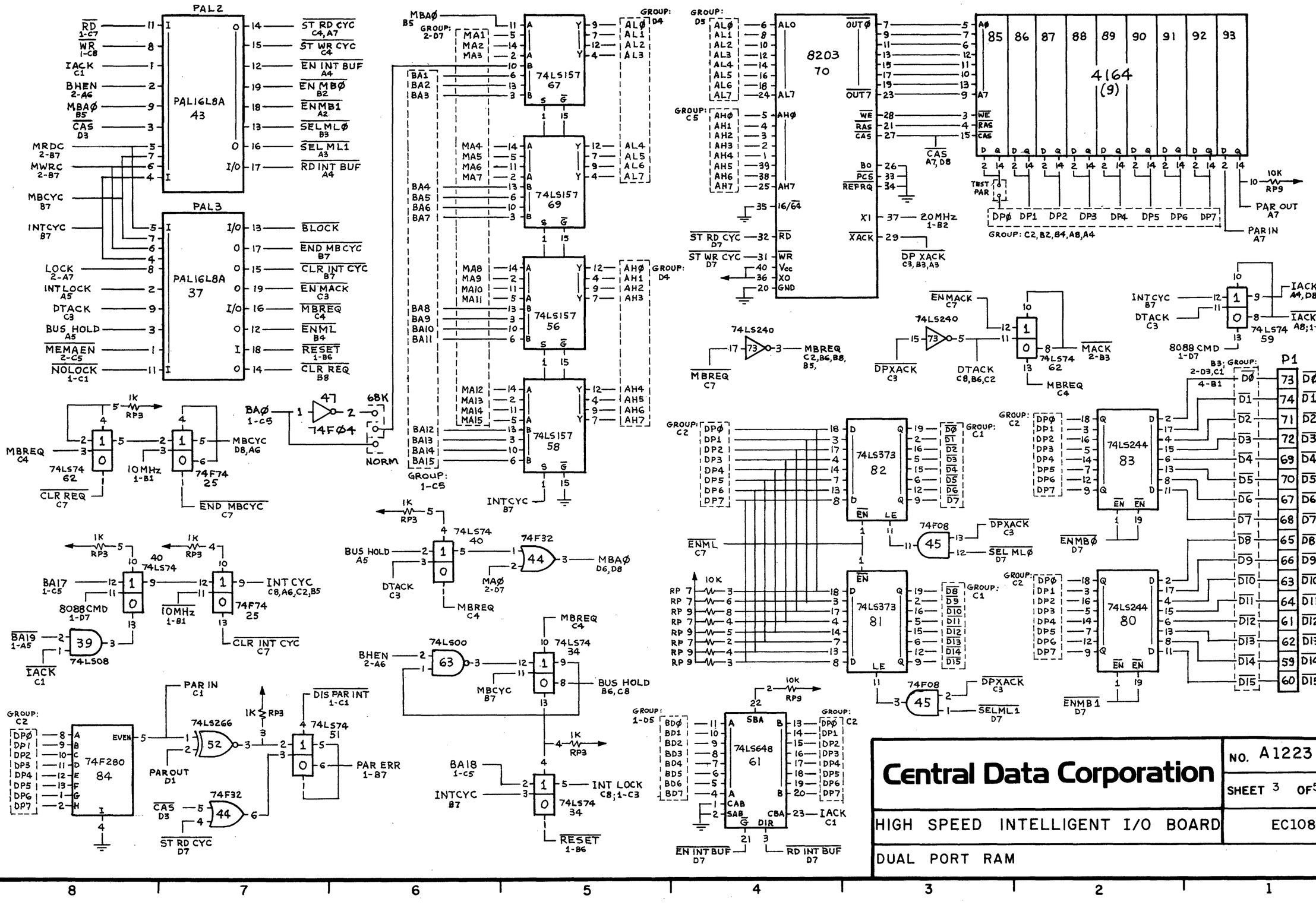
NO. A1222

SHEET 2 OF5

HIGH SPEED INTELLIGENT I/O BOARD

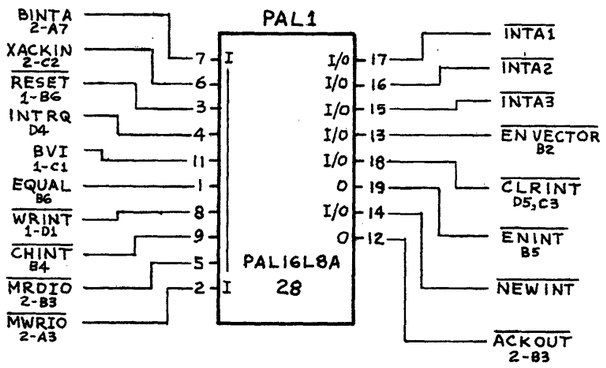
EC1084

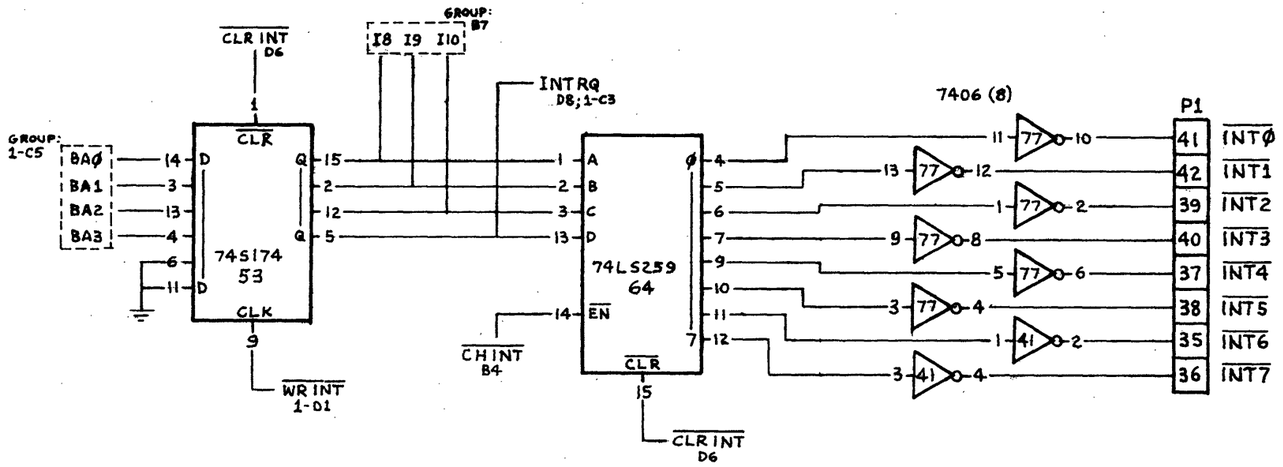
MULTIBUS INTERFACE CIRCUITRY

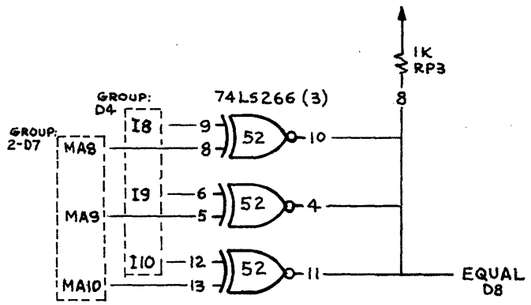


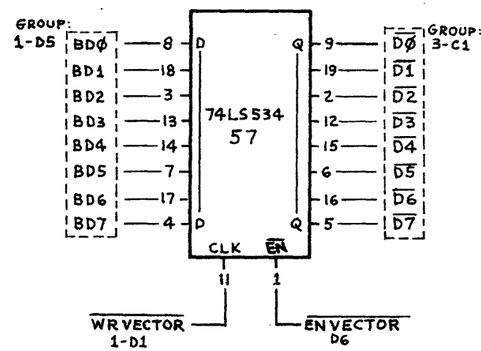
Central Data Corporation

NO.	A1223
SHEET	3 OF 5
HIGH SPEED INTELLIGENT I/O BOARD	EC1084
DUAL PORT RAM	

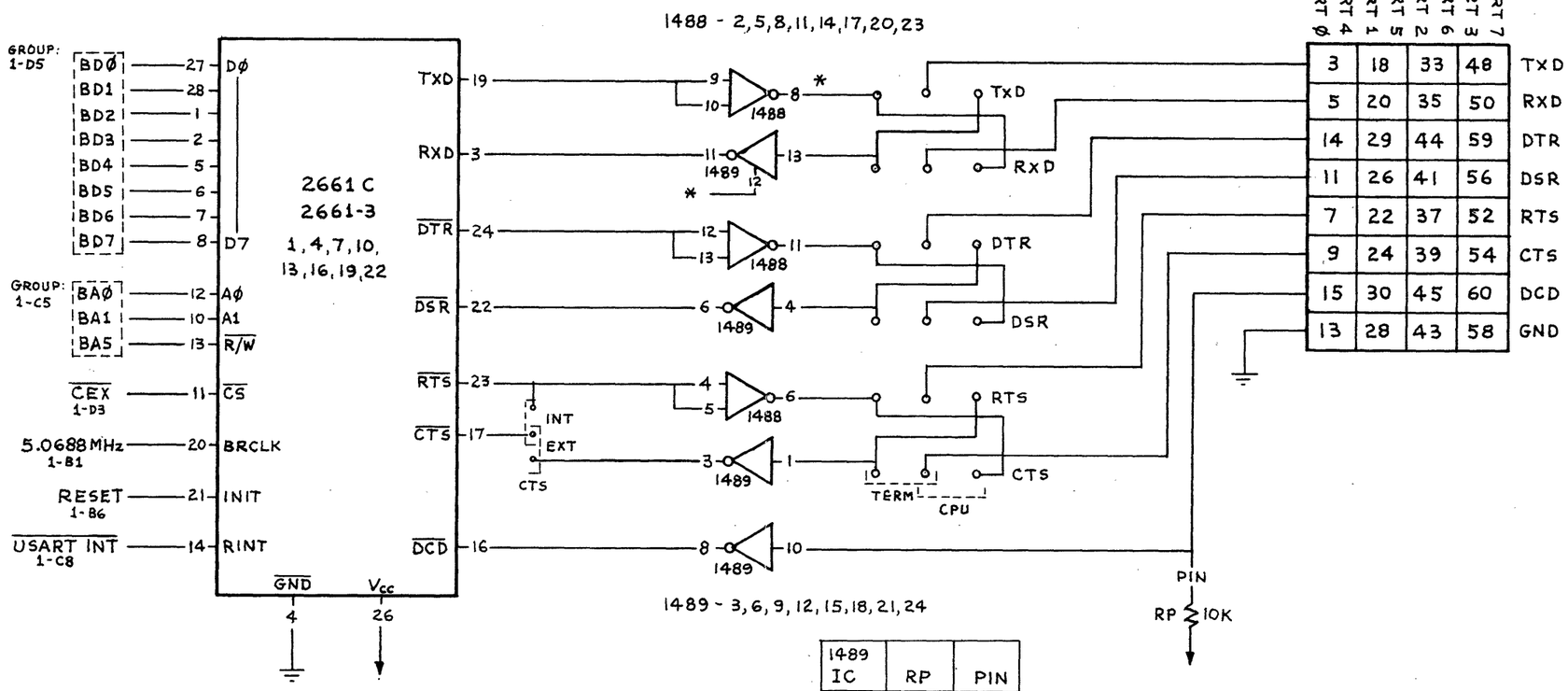








Central Data Corporation	NO. A1224
	SHEET 4 OF 5
HIGH SPEED INTELLIGENT I/O BOARD	EC1084
MULTIBUS INTERRUPT CIRCUITRY	



* USER SUPPLIED CAP

1489 IC	RP	PIN
3	1	2
6	1	3
9	1	5
12	1	4
15	2	2
18	2	3
21	2	4
24	2	5

Central Data Corporation

NO. A 1201

SHEET 5 OF 5

HIGH SPEED INTELLIGENT I/O BOARD

EC1084

USART SECTION – REPEAT 8 TIMES

10. 2661 Data Sheets

The following pages contain the data sheets for the Signetics 2661 USART. This data sheet is reproduced with the permission and courtesy of Signetics Corporation.

ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI) SC2661

DESCRIPTION

The Signetics 2661 EPCI is a universal synchronous/asynchronous data communications controller chip that is an enhanced pin compatible version of the 2651. It interfaces directly to most 8-bit microprocessors and may be used in a polled or interrupt driven system environment. The 2661 accepts programmed instructions from the microprocessor while supporting many serial data communications disciplines—synchronous and asynchronous—in the full or half-duplex mode. Special support for BISYNC is provided.

The EPCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The 2661 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode. Each version of the EPCI (A, B, C) has a different set of baud rates.

The EPCI is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 28-pin DIP.

FEATURES

- **Synchronous operation**
  - **5 to 8-bit characters plus parity**
  - **Single or double SYN operation**
  - **Internal or external character synchronization**
  - **Transparent or non-transparent mode**
  - **Transparent mode DLE stuffing (Tx) and detection (Rx)**
  - **Automatic SYN or DLE-SYN insertion**
  - **SYN, DLE and DLE-SYN stripping**
  - **Odd, even, or no parity**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
- **Asynchronous operation**
  - **5 to 8-bit characters plus parity**
  - **1, 1½ or 2 stop bits transmitted**
  - **Odd, even, or no parity**
  - **Parity, overrun and framing error detection**
  - **Line break detection and generation**
  - **False start bit detection**
  - **Automatic serial echo mode (echoplex)**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
    dc to 62.5K bps (16X clock)
    dc to 15.625K bps (64X clock)

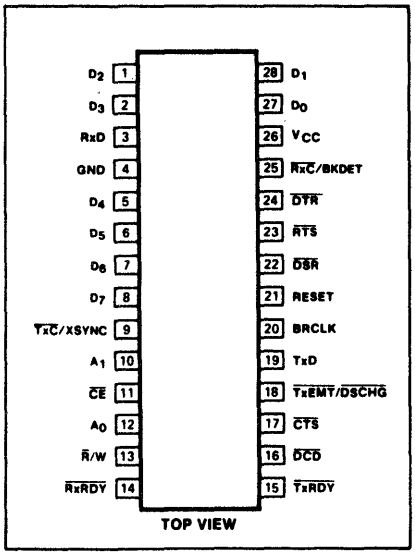
OTHER FEATURES

- **Internal or external baud rate clock**
- **3 baud rate sets**
- **16 internal rates for each set**
- **Double buffered transmitter and receiver**
- **Dynamic character length switching**
- **Full or half duplex operation**
- **Fully compatible with 2650 CPU**
- **TTL compatible inputs and outputs**
- **RxC and TxC pins are short circuit protected**
- **3 open drain MOS outputs can be wire-ORed**
- **Single 5V power supply**
- **No system clock required**
- **28-pin dual in-line package**

APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front end processors**
- **Remote data concentrators**
- **Computer to computer links**
- **Serial peripherals**
- **BISYNC adaptors**

PIN CONFIGURATION



ORDERING CODE

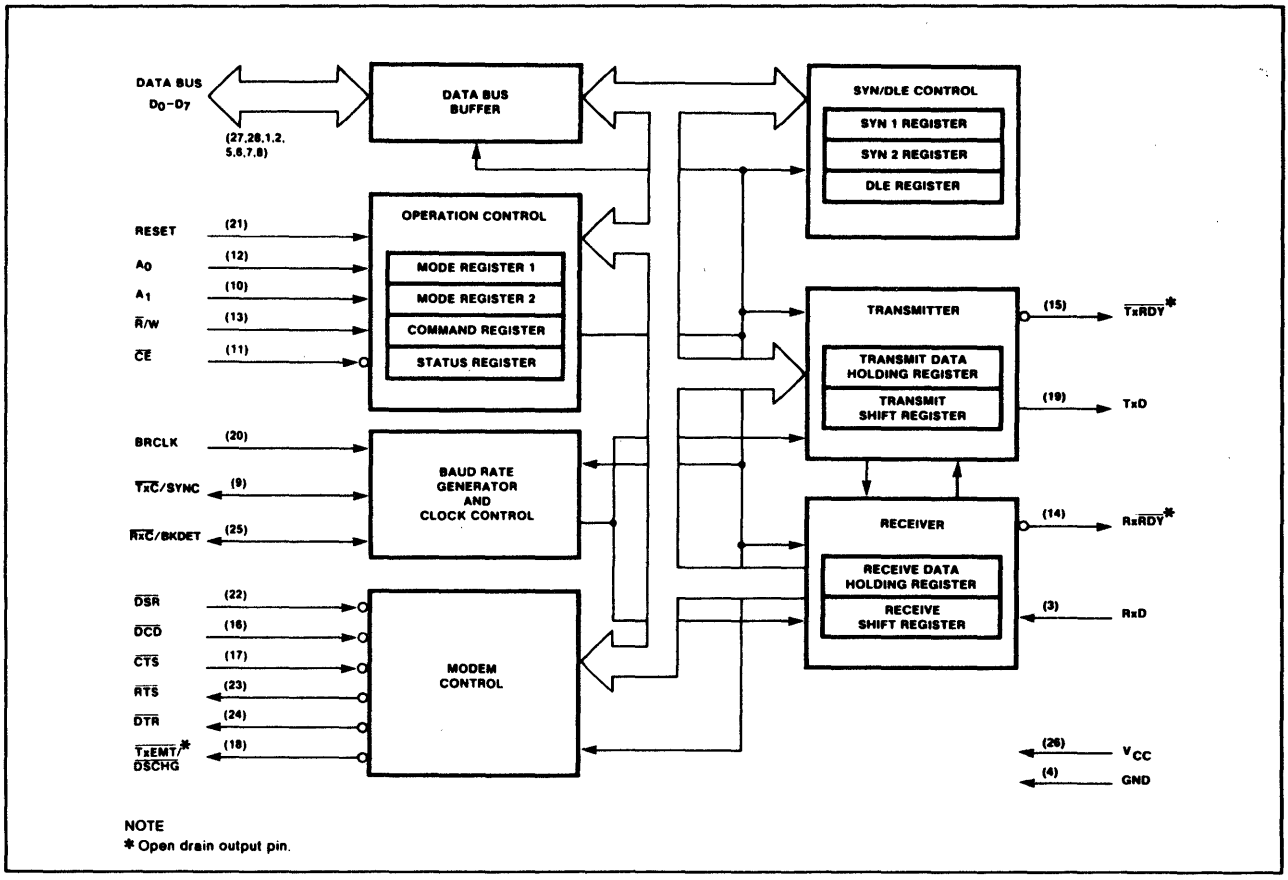
PACKAGES	COMMERCIAL RANGES V_{CC} = 5V ±5%, T_A = 0°C to 70°C	
Ceramic DIP	SC2661ACSI28 SC2661BCSI28 SC2661CCSI28	See table 1 for baud rates
Plastic DIP	SC2661ACSN28 SC2661BCSN28 SC2661CCSN28	See table 1 for baud rates

PIN DESIGNATION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
27,28,1, 2,5-8	D_0-D_7	8-bit data bus	I/O
21	RESET	Reset	I
12,10	A_0-A_1	Internal register select lines	I
13	R/W	Read or write command	I
11	CE	Chip enable input	I
22	DSR	Data set ready	I
24	DTR	Data terminal ready	O
23	RTS	Request to send	O
17	CTS	Clear to send	I
16	DCD	Data carrier detected	I
18	TxEMT/DSCHG	Transmitter empty or data set change	O
9	TxC/XSYNC	Transmitter clock/external SYNC	I/O
25	RxC/BKDET	Receiver clock/break detect	I/O
19	TxD	Transmitter data	O
3	RxD	Receiver data	I
15	TxRDY	Transmitter ready	O
14	RxRDY	Receiver ready	O
20	BRCLK	Baud rate generator clock	I
26	V_{CC}	+5V supply	I
4	GND	Ground	I

ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI) SC2661

BLOCK DIAGRAM



NOTE
* Open drain output pin.

BLOCK DIAGRAM

The EPCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing and protocol are presented in the EPCI programming section of this data sheet.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS SC2661A (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6144
0001	75	1.2	-	4096
0010	110	1.7598	−0.01	2793
0011	134.5	2.152	-	2284
0100	150	2.4	-	2048
0101	200	3.2	-	1536
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1050	16.8329	0.196	292
1001	1200	19.2	-	256
1010	1800	28.7438	−0.19	171
1011	2000	31.9168	−0.26	154
1100	2400	38.4	-	128
1101	4800	76.8	-	64
1110	9600	153.6	-	32
1111	19200	307.2	-	16

Timing

The EPCI contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See table 1.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

Table 1 BAUD RATE GENERATOR CHRACTERISTICS (Cont'd)

SC2661B (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	45.5	0.7279kHz	0.005	6752
0001	50	0.8	-	6144
0010	75	1.2	-	4096
0011	110	1.7598	−0.01	2793
0100	134.5	2.152	-	2284
0101	150	2.4	-	2048
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1200	19.2	-	256
1001	1800	28.7438	−0.19	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	-	128
1100	4800	76.8	-	64
1101	9600	153.6	-	32
1110	19200	307.2	-	16
1111	38400	614.4	-	8

SC2661C (BRCLK = 5.0688MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6336
0001	75	1.2	-	4224
0010	110	1.76	-	2880
0011	134.5	2.1523	0.016	2355
0100	150	2.4	-	2112
0101	300	4.8	-	1056
0110	600	9.6	-	528
0111	1200	19.2	-	264
1000	1800	28.8	-	176
1001	2000	32.081	0.253	158
1010	2400	38.4	-	132
1011	3600	57.6	-	88
1100	4800	76.8	-	66
1101	7200	115.2	-	44
1110	9600	153.6	-	33
1111	19200	316.8	3.125	16

NOTE
16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X and BRG can be used only for TxC.

Table 2 CPU-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
V_{CC}	26	I	+5V supply input
GND	4	I	Ground
RESET	21	I	A high on this input performs a master reset on the 2661. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A_1-A_0	10,12	I	Address lines used to select internal EPCI registers.
$\overline{R}/W$	13	I	Read command when low, write command when high.
$\overline{CE}$	11	I	Chip enable command. When low, indicates that control and data lines to the EPCI are valid and that the operation specified by the $\overline{R}/W$, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the three-state condition.
D_7-D_0	8,7,6,5, 2,1,28,17	I/O	8-bit, three-state data bus used to transfer commands, data and status between EPCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{TxRDY}$	15	O	This output is the complement of status register bit SR0. When low, it indicates that the transmit data holding register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{RxRDY}$	14	O	This output is the complement of status register bit SR1. When low, it indicates that the receive data holding register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{TxEMT}$/ $\overline{DSCHG}$	18	O	This output is the complement of status register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs has occurred. This output goes high when the status register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.

OPERATION

The functional operation of the 2661 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the EPCI programming section of the data sheet.

After programming, the EPCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The 2661 is conditioned to receive data when the $\overline{DCD}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low (mark to space) transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and one stop bit have been assembled. The data are then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{RxRDY}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive going edge of $\overline{RxC}$ corresponding to the received character boundary. If the stop bit is present, the receiver will immediately begin its search for the next start bit. If the stop bit is absent (framing error), the receiver will interpret a space as a start bit if it persists into the next bit time interval. If a break condition is detected (RxD is low for the entire character as well as the stop bit), only one character consisting of all zeros (with the FE status bit SR5 set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

Pin 25 can be programmed to be a break detect output by appropriate setting of MR27-MR24. If so, a detected break will cause that pin to go high. When RxD returns to mark for one RxC time, pin 25 will go low. Refer to the break detection timing diagram.

Table 3 DEVICE-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
BRCLK	20	I	Clock input to the internal baud rate generator (see table 1). Not required if external receiver and transmitter clocks are used.
*$\overline{RxC}$/BKDET	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. Data are sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin can be a 1X/16X clock or a break detect output pin.
*$\overline{TxC}$/XSYNC	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin can be a 1X/16X clock output or an external jam synchronization input.
RxD	3	I	Serial data input to the receiver. "Mark" is high, "space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{DSR}$	22	I	General purpose input which can be used for data set ready or ring indicator condition. Its complement appears as status register bit SR7. Causes a low output on $\overline{TxEMT}$/$\overline{DSCHG}$ when its state changes if CR2 or CR0 = 1.
$\overline{DCD}$	16	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{TxEMT}$/$\overline{DSCHG}$ when its state changes if CR2 or CR0 = 1. If $\overline{DCD}$ goes high while receiving, the RxC is internally inhibited.
$\overline{CTS}$	17	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{DTR}$	24	O	General purpose output which is the complement of command register bit CR1. Normally used to indicate data terminal ready.
$\overline{RTS}$	23	O	General purpose output which is the complement of command register bit CR5. Normally used to indicate request to send. If the transmit shift register is not empty when CR5 is reset (1 to 0), then $\overline{RTS}$ will go high one TxC time after the last serial bit is transmitted.

NOTE
*$\overline{RxC}$ and $\overline{TxC}$ outputs have short circuit protection max. C_L = 100pF. Outputs become open circuited upon detection of a zero pulled high or a one pulled low.

When the EPCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN(CR2). In this mode, as data are shifted into the receiver shift register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the EPCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization.) When synchronization has been achieved, the EPCI continues to assemble characters and transfer them to the holding register, setting the RxRDY status bit and asserting the $\overline{RxRDY}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the holding register. Note that the SYN characters used to establish initial synchronization are not transferred to the holding register in any case.

External jam synchronization can be achieved via pin 9 by appropriate setting of MR27-MR24. When pin 9 is an XSYNC input, the internal SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled. Each positive going signal on XSYNC will cause the receiver to establish synchronization on the rising edge of the next RxC pulse. Character assembly will start with the RxD input at this edge. XSYNC may be lowered on the next rising edge of RxC. This external synchronization will cause the SYN DETECT status bit to be set until the status register is read. Refer to XSYNC timing diagram.

Transmitter

The EPCI is conditioned to transmit data when the $\overline{CTS}$ input is low and the TxEN command register bit is set. The 2661 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{TxRDY}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data are transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the $\overline{TxEMT}/\overline{DSCHG}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit (CR3) high.

In the synchronous mode, when the 2661 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the EPCI unless the CPU fails to send a new character to the EPCI by the time the transmitter has completed sending the previous character. Since synchronous communication does not allow gaps between characters, the EPCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the transmit data holding register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in the THR.

EPCI PROGRAMMING

Prior to initiating data communications, the 2661 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The EPCI can be reconfigured at any time during program execution. A flowchart of the intialization process appears in figure 1.

The internal registers of the EPCI are accessed by applying specific signals to the $\overline{CE}$, $\overline{R}/W$, A_1 and A_0 inputs. The conditions necessary to address each register are shown in table 4.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and

Table 4 2661 REGISTER ADDRESSING

$\overline{CE}$	A_1	A_0	$\overline{R}/W$	FUNCTION
1	X	X	X	Three-state data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers ½
0	1	0	1	Write mode registers ½
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE
See AC characteristics section for timing requirements.

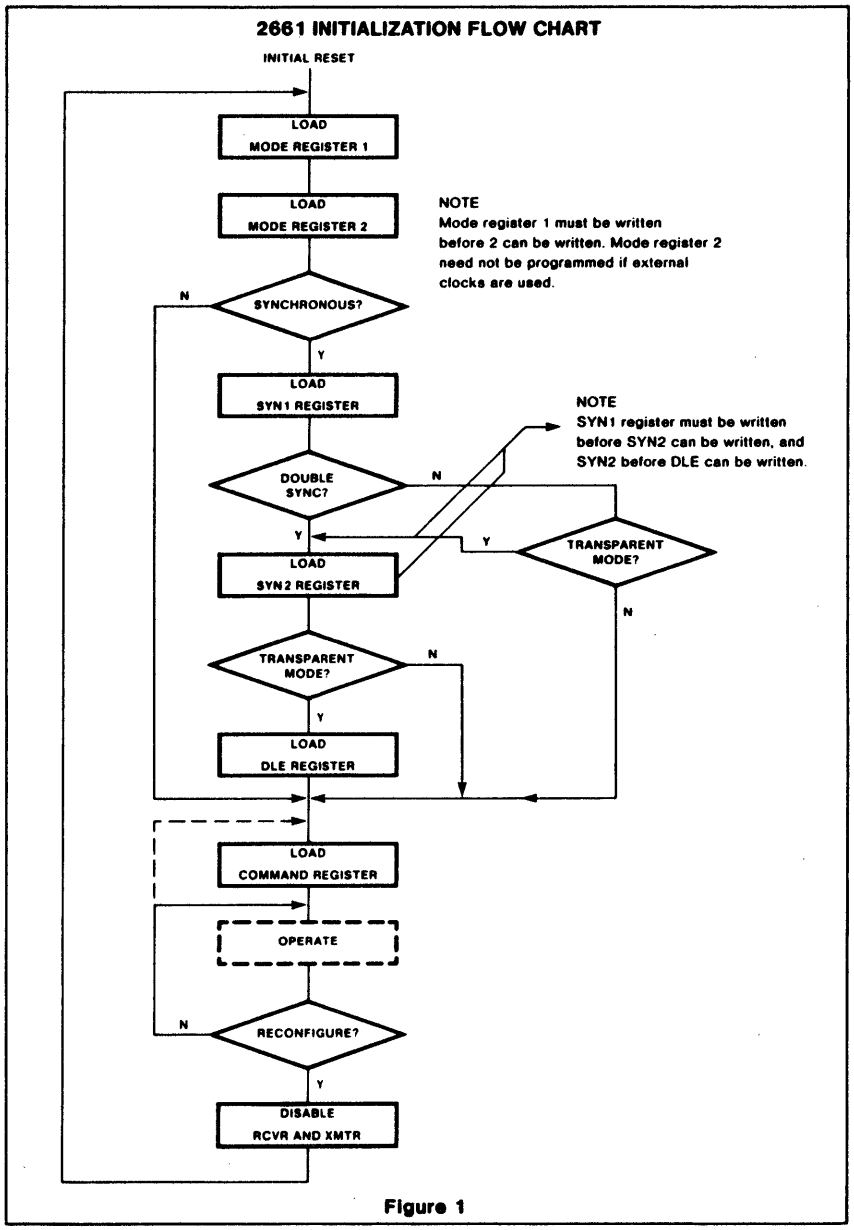


Figure 1

$\overline{R}/W$ = 1. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses mode register 1, and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 register and mode register 1 by a RESET input or by performing a read command register operation, but are unaffected by any other read or write operation.

The 2661 register formats are summarized in tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the EPCI, while the command register controls the operation within this basic framework. The EPCI indicates its status in the status register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7 or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asynchronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit.) In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill and SYN detect, but the normal synchronization sequence is used to establish character sync. When transmitting, a DLE character in the transmit holding register will cause a second DLE character to be transmitted. This DLE stuffing eliminates the software DLE compare and stuff on each transparent mode data character. If the send DLE command (CR3) is active when a DLE is loaded into THR, only one additional DLE will be transmitted. Also, DLE stripping and DLE detect (with MR14 = 0) are enabled.

The bits in the mode register affecting character assembly and disassembly (MR12-MR16) can be changed dynamically (during active receive/transmit operation). The character mode register affects both the transmitter and receiver; therefore in synchronous mode, changes should be made only in half duplex mode (RxEN = 1 or TxEN = 1, but not both simultaneously = 1). In asynchronous mode, character changes should be made when RxEN and TxEN=0 or when TxEN = 1 and the transmitter is marking in half duplex mode (RxEN = 0).

To effect assembly/disassembly of the next received/transmitted character, MR12-15 must be changed within n bit times of the active going state of $\overline{RxRDY}$/$\overline{TxRDY}$. Transparent and non-transparent mode changes (MR16) must occur within n-1 bit times of the character to be affected when the receiver or transmitter is active. (n = smaller of the new and old character lengths.)

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21 and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable for each EPCI version (−1, −2, −3). Version 1 and 2 specify a 4.9152 MHz TTL input at BRCLK (pin 20); version 3 specifies a 5.0688 MHz input which is identical to the Signetics 2651. MR23-20 are don't cares if external clocks are selected (MR25-MR24 = 0). The individual rates are given in table 1.

MR24-MR27 select the receive and transmit clock source (either the BRG or an external input) and the function at pins 9 and 25. Refer to table 6.

Command Register (CR)

Table 7 illustrates the command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{RxC}$ rising edge. Disabling the receiver causes $\overline{RxRDY}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state

Table 5 MODE REGISTER 1 (MR 1)

<table>
<tr><th colspan="2">MR17 MR16</th><th>MR15</th><th>MR14</th><th>MR13 MR12</th><th>MR11 MR10</th></tr>
<tr><td colspan="2">Sync/Async</td><td>Parity Type</td><td>Parity Control</td><td>Character Length</td><td>Mode and Baud Rate Factor</td></tr>
<tr><td colspan="2">Async: Stop Bit Length
00 = Invalid
01 = 1 stop bit
10 = 1½ stop bits
11 = 2 stop bits</td><td rowspan="2">0 = Odd
1 = Even</td><td rowspan="2">0 = Disabled
1 = Enabled</td><td rowspan="2">00 = 5 bits
01 = 6 bits
10 = 7 bits
11 = 8 bits</td><td rowspan="2">00 = Synchronous 1X rate
01 = Asynchronous 1X rate
10 = Asynchronous 16X rate
11 = Asynchronous 64X rate</td></tr>
<tr><td>Sync:
Number of SYN char
0 = Double SYN
1 = Single SYN</td><td>Sync:
Transparency Control
0 = Normal
1 = Transparent</td></tr>
</table>

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 6 MODE REGISTER 2 (MR2)

MR27–MR24											MR23–MR20
	TxC	RxC	Pin 9	Pin 25		TxC	RxC	Pin 9	Pin 25	Mode	Baud Rate Selection
0000	E	E	TxC	RxC	1000	E	E	XSYNC[1]	RxC/TxC	sync	See baud rates in table 1
0001	E	I	TxC	1X	1001	E	I	TxC	BKDET	async	
0010	I	E	1X	RxC	1010	I	E	XSYNC[1]	RxC	sync	
0011	I	I	1X	1X	1011	I	I	1X	BKDET	async	
0100	E	E	TxC	RxC	1100	E	E	XSYNC[1]	RxC/TxC	sync	
0101	E	I	TxC	16X	1101	E	I	TxC	BKDET	async	
0110	I	E	16X	RxC	1110	I	E	XSYNC[1]	RxC	sync	
0111	I	I	16X	16X	1111	I	I	16X	BKDET	async	

NOTES
1. When pin 9 is programmed as XSYNC input, SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled.

E = External clock
I = Internal clock (BRG)
1X and 16X are clock outputs

Table 7 COMMAND REGISTER (CR)

CR7 CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode	Request To Send	Reset Error	Sync/Async	Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = Normal operation 01 = Async: Automatic echo mode Sync: SYN and/or DLE stripping mode 10 = Local loop back 11 = Remote loop back	0 = Force $\overline{RTS}$ output high one clock time after TxSR serialization 1 = Force $\overline{RTS}$ output low	0 = Normal 1 = Reset error flags in status register (FE, OE, PE/DLE detect)	**Async:** **Force break** 0 = Normal 1 = Force break	0 = Disable 1 = Enable	0 = Force $\overline{DTR}$ output high 1 = Force $\overline{DTR}$ output low	0 = Disable 1 = Enable
			Sync: **Send DLE** 0 = Normal 1 = Send DLE			

Table 8 STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{DSR}$ input is high 1 = $\overline{DSR}$ input is low	0 = $\overline{DCD}$ input is high 1 = $\overline{DCD}$ input is low	**Async:** 0 = Normal 1 = Framing Error	0 = Normal 1 = Overrun Error	**Async:** 0 = Normal 1 = Parity error	0 = Normal 1 = Change in $\overline{DSR}$, or $\overline{DCD}$,or transmit shift register is empty	0 = Receive holding register empty 1 = Receive holding register has data	0 = Transmit holding register busy 1 = Transmit holding register empty
		Sync: 0 = Normal 1 = SYN detected		**Sync:** 0 = Normal 1 = Parity error or DLE received			

(high) while $\overline{TxRDY}$ and $\overline{TxEMT}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected. A 0 to 1 transition of CR2 will initiate start bit search (async) or hunt mode (sync).

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{DTR}$ and $\overline{RTS}$ outputs. Data at the outputs are the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for at least one bit time before beginning transmission of the next character in the transmit data holding register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the transmit data holding register. Since this is a one time command, CR3 does not have to be reset by software. CR3 should be set when entering and exiting transparent mode and for all DLE—non-DLE character sequences.

Setting CR4 causes the error flags in the status register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

Table 9 SC2661 EPCI vs SC2651 PCI

FEATURE	EPCI	PCI
1. MR2 Bit 6, 7	Control pin 9, 25	Not used
2. DLE detect-SR3	SR3 = 0 for DLE-DLE, DLE-SYNC1	SR3 = 1 for DLE-DLE, DLE-SYNC1
3. Reset of SR3, DLE detect	Second character after DLE, or receiver disable, or CR4 = 1	Receiver disable, or CR4 = 1
4. Send DLE-CR3	One time command	Reset via CR3 on next $\overline{\text{TxRDY}}$
5. DLE stuffing in transparent mode	Automatic DLE stuffing when DLE is loaded except if CR3 = 1	None
6. SYNC1 stripping in double sync non-transparent mode	All SYNC1	First SYNC1 of pair
7. Baud rate versions	Three	One
8. Terminate ASYNC transmission (drop RTS)	Reset CR5 in response to $\overline{\text{TxRDY}}$ changing from 1 to 0	Reset CR0 when $\overline{\text{TxEMT}}$ goes from 1 to 0. Then reset CR5 when $\overline{\text{TxEMT}}$ goes from 0 to 1
9. Break detect	Pin 25[1]	FE and null character
10. Stop bit searched	One	Two
11. External jam sync	Pin 9[2]	No
12. Data bus timing	Improved over 2651	—
13. Data bus drivers	Sink 2.2mA Source 400μA	Sink 1.6mA Source 100μA

NOTES
1. Internal BRG used for RxC.
2. Internal BRG used for TxC.

When CR5 (RTS) is set, the $\overline{\text{RTS}}$ pin is forced low and the transmit serial logic is enabled. A 1 to 0 transition of CR5 will cause $\overline{\text{RTS}}$ to go high (inactive) one TxC time after the last serial bit has been transmitted (if the transmit shift register was not empty).

The EPCI can operate in one of four sub-modes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

In asynchronous mode, CR7-CR6 = 01 places the EPCI in the automatic echo mode. Clocked, regenerated received data are automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{\text{TxRDY}}$ output = 1.
4. The $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7-CR6 = 01 places the EPCI in the automatic SYN/DLE stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the receive data holding register (RHR).
2. In the non-transparent, double SYN mode (MR17-MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR.
3. In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped.

Note that automatic stripping mode does not affect the setting of the DLE detect and SYN detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In local loop back mode (CR7-CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{\text{DTR}}$ is connected to $\overline{\text{DCD}}$ and $\overline{\text{RTS}}$ is connected to $\overline{\text{CTS}}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{\text{DTR}}$, $\overline{\text{RTS}}$ and $\overline{\text{TxD}}$ outputs are held high.
5. The $\overline{\text{CTS}}$, $\overline{\text{DCD}}$, $\overline{\text{DSR}}$ and RxD inputs are ignored.

Additional requirements to operate in the local loop back mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the EPCI.

The second diagnostic mode is the remote loop back mode (CR7-CR6 = 11). In this mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data are sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{\text{RxRDY}}$, $\overline{\text{TxRDY}}$, and $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ outputs are held high.
5. CR1 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the status register (as shown in table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit data holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the holding register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{\text{TxRDY}}$ output pin is low. In

the automatic echo and remote loop back modes, the output is held high.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to zero, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{\text{RxRDY}}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$ inputs (when CR2 or CR0 = 1) or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when TxEN = 1 or RxEN = 1. It is cleared when the status register is read by the CPU. If the status register is read twice and SR2 = 1 while SR6 and SR7 remain unchanged, then a TxEMT condition exists. When SR2 is set, the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching DLE register was received and the present character is neither SYN1 nor DLE. This bit is cleared when the next character following the above sequence is loaded into RHR, when the receiver is disabled, or by a reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled or by the reset error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by a stop bit, i.e., only the first stop bit is checked. If RHR = 0 when SR5 = 1, a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character in single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the reset error command is given in asynchronous mode, or when the status register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{\text{DCD}}$ and $\overline{\text{DSR}}$ inputs respectively. A low input sets its corresponding status bit, and a high input clears it.

Central Data Corporation
1602 Newton Drive
Champaign, IL 61821-1098
(217) 359-8010
TWX 910-245-0787
(800) 482-0315 Outside Illinois