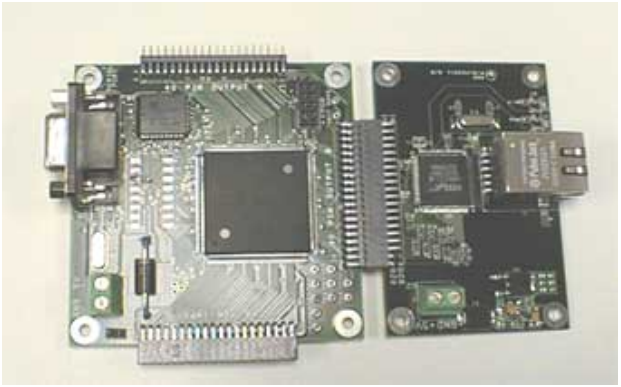
COM-5002 LAN / IP NETWORK INTERFACE

Key Features

- TCP-IP server connects ComBlock assemblies to network clients for data transfer, monitoring and control.
- Standard 100Base-Tx/10Base-T, RJ-45 connector. Autonegotiation or manual settings: 10/100 Mbit/s, full/half duplex.
- Maximum sustained throughput:
  25 Mbits/s (100Base-Tx).
  5.3 Mbits/s (10Base-T).
  Actual speed depends on host computer.
- Elastic buffering and flow-control on each transmit and receive link.
- Monitoring and control of ComBlock assemblies over LAN or serial link from a graphical user interface.
- Single 5V supply. Standard 40 pin 2mm dual row connectors (right, left)

For the latest data sheet, please refer to the **ComBlock** web site: www.comblock.com/download/com5002.pdf.
These specifications are subject to change without notice.

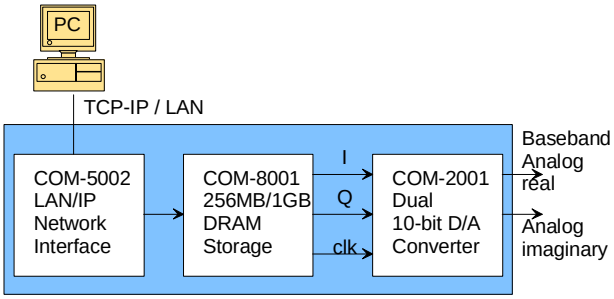
For an up-to-date list of **ComBlock** modules, please refer to www.comblock.com/product_list.htm .



Typical Applications

Arbitrary Waveform Signal Generation

Files representing binary or analog sampled signals can be uploaded through the COM-5002 to the COM-8001 SDRAM memory over the network, then played back at the selected speed. Various ComBlocks can be used to generate analog signals at baseband, 70 MHz intermediate frequency or radio-frequency.



Arbitrary waveform generator, analog baseband example

MSS • 18221 Flower Hill Way #A • Gaithersburg, Maryland 20879 • U.S.A.
Telephone: (240) 631-1111 Facsimile: (240) 631-1676 www.ComBlock.com
 Issued 3/16/2005

Interfaces

Input Interface	Definition
DATA_IN[7:0]	Input signal. The input width is user programmable as a function of the data source.
SAMPLE_CLK_IN	Input signal sampling clock. One CLK_IN-wide pulse. Read the input signal at the rising edge of CLK_IN when SAMPLE_CLK_IN = '1'. Samples can be consecutive. For example, SAMPLE_CLK_IN can be fixed at '1' to indicate that new input samples are provided once per CLK_IN clock period. Signal is pulled-up.
CLK_IN	Input reference clock for synchronous I/O. DATA_IN and SAMPLE_CLK_IN are read at the rising edge of CLK_IN. Maximum 40 MHz.
Output Interface	**Definition**
DATA_OUT[7:0]	Output signal. The output width is user programmable as a function of the data sink.
SAMPLE_CLK_OUT	Output signal sampling clock. One CLK_OUT-wide pulse. Read the output signal at the rising edge of CLK_OUT when SAMPLE_CLK_OUT = '1'.
CLK_OUT	40 MHz output reference clock. (from internal oscillator).
Other Interfaces	**Definition**
LAN	4 wire. 10Base-T/100Base-TX. RJ45 connector. NIC wiring. Use standard category 5 cable for connection to a Hub. Use crossover cable for connection to a host computer.
Serial Monitoring & Control	DB9 connector. 115 Kbaud/s. 8-bit, no parity, one stop bit. No flow control.
Power Interface	4.75 – 5.25VDC. Terminal block. Power consumption is typically 350mA.

Initial Configuration (via Serial Link)

The IP address must first be configured over serial link. This network setting is saved in non-volatile memory. Once the correct network setting is configured, the Comblock Control Center and this ComBlock assembly can communicate over the intranet or internet as well as over a serial link.

Configuration (via Serial Link / LAN)

Complete assemblies can be monitored and controlled centrally over a single serial or LAN connection.

The module configuration parameters are stored in non-volatile memory. All control registers are read/write.

Undefined control registers or register bits are for backward software compatibility and/or future use. They are ignored in the current firmware version.

Parameters	Configuration
IP address	4-byte IP address. Example : 0x AC 10 01 80 designates address 172.16.1.128 The new address becomes effective immediately (no need to reset the ComBlock). REG0: MSB REG1 REG2 REG3: LSB
Reserved	REG4-19 Reserved for other network configurations. No need to write any data.
Input format	00000 = J2 input is disabled 00001 = 1-bit wide from J2 01000 = 8-bit wide from J2 11110 = test mode. Internally generated 8-bit wide periodic counting sequence (0-255) as input. J2 input is disabled. The throughput is determined by the TCP-IP client. REG20 bits 4-0
Output format	00001 = 1-bit wide 01000 = 8-bit wide REG21 bits 4-0
COM-8001 external trigger	Special use: Writing to REG22 with a '1' in bit 1 will generate a 1 CLK wide pulse on pin J3/B6. The main application is to trigger the COM-8001 file playback/download. There is no need to reset this bit to '0' prior to writing a '1'. REG22 bit 1.
10Base-T / 100Base-TX LAN selection	00 = 10Base-T 01 = 100Base-TX 10 = Auto negotiation Changes will take effect at the next power up. REG22 bits 3-2

Half / Full duplex LAN link	0 = half duplex 1 = full duplex Changes will take effect at the next power up. REG22 bit 4
Promiscuous (listen) mode	Test mode. Incoming packets are not checked for matching destination address. 0 = disabled. 1 = enabled. REG22 bit 5

Baseline configurations can be found at www.comblock.com/tsbasic_settings.htm and imported into the ComBlock assembly using the ComBlock Control Center File | Import menu.

Monitoring (via Serial Link / LAN)

Monitoring registers are read-only.

Parameters	**Monitoring**
TCP-IP connection on port 1024 (data stream)	1 = connected, 0 otherwise. REG23 bit 0
TCP-IP connection on port 1028 (Monitoring & Control)	1 = connected, 0 otherwise. REG23 bit 2
Transmit data elastic buffer empty	1 = empty, 0 otherwise REG23 bit 3
Transmit data elastic buffer full	1 = full, 0 otherwise REG23 bit 4
Receive data elastic buffer empty	1 = empty, 0 otherwise REG23 bit 5
Receive data elastic buffer more than half full	1 = more than half full, 0 otherwise REG23 bit 6
Number of bytes transmitted from LAN to digital device	Total number of bytes transmitted over data and signaling channels. 32-bit byte count. Counter rolls over when reaching 0xFFFFFFFF. REG24: bits 7-0 (LSB) REG25: bits 15-8 REG26: bits 23-16 REG27: bits 31-24 (MSB)
Number of bytes received from digital device and forwarded to LAN	Total number of bytes received over data and signaling channels. 32-bit byte count. Counter rolls over when reaching 0xFFFFFFFF. REG28: bits 7-0 (LSB) REG29: bits 15-8
	REG30: bits 23-16 REG31: bits 31-24 (MSB)
MAC address	Unique 48-bit hardware address (802.3). In the form REG32:REG33:REG34:…:REG37
Option o / Version v	Returns ‘5002ov’ when prompted for option o and version v numbers.

As the monitoring data is constantly changing, it is important to be able to prevent changes while reading a multi-byte parameter. The monitoring data is latched upon reading register 23. Therefore, register 23 should always be read first.

IP Protocols

This module supports the following IP protocols:

- Ping
- ARP
- TCP-IP

Ping

The module responds to ping requests with size up to 470 bytes. Ping can be used to check the module response over the network. Ping can be used at any time, concurrently with other transmit and receive transactions. For example, on a Windows operating system, open the Command prompt window and type “ping –t –l 470 172.16.1.128” to send pings forever of length 470 bytes to address 172.16.1.128.

TCP-IP

As a Server, the module opens the following sockets in listening mode:

Port 1024: transmit and receive data streams

Port 1028: monitoring and control port

Operation

Concept

The COM-5002 converts a TCP-IP socket stream into a simple data stream and vice versa. On the transmit side, the COM-5002 decodes the TCP-IP protocol and extracts the data from the network client. TCP, IP and Network information, and in particular routing information, are not transmitted from one end to the other.

At the receiving end, the network client must first connect to the COM-5002 to receive data.

The COM-5002 maintains the flow-control information between the TCP-IP socket and the input/output interfaces. For example, if the COM-5002 is connected to a COM-1001 QPSK modulator configured for 1 Mbit/s data throughput, the network client (i.e. data source) will be asked for 1 Mbit/s throughput over the TCP-IP link.

Throughput Benchmarks

The COM-5002 is capable of a sustained (average) throughput of 25 Mbits/s over 100base-Tx and 5.3 Mbit/s over 10base-T. In most cases, the sustained throughput is limited by the TCP-IP client computer and the application running on the client computer as illustrated in the one-way data transfer benchmarks below:

Throughput tests conditions	**Throughput**
Client: Intel Pentium 4 2.6 GHz running winsock-based console application. Direct cross-over LAN cable. No network connection. No other application running. COM-5002 configured as 'Auto Negotiation". 100Base-Tx connection.	25 Mbits/s 100Mbytes transferred in 32 seconds.
Client: Intel Celeron 766 MHz running winsock-based console application. Direct cross-over LAN cable. No network connection. No other application running. COM-5002 configured as 'Auto Negotiation". 100Base-Tx connection.	14 Mbits/s 100Mbytes transferred in 57 seconds.
Client: Intel Pentium 4 2.6 GHz running winsock-based console application. Direct cross-over LAN cable. No network connection. No other application running. COM-5002 or client computer configured as '10Base-T". 10Base-T connection.	5.36 Mbits/s 100Mbytes transferred in 149 seconds.
Client: Intel Celeron 766 MHz running winsock-based console application. Direct cross-over LAN cable. No network connection. No other application running. COM-5002 or client computer configured as '10Base-T". 10Base-T connection.	4.16 Mbits/s 100Mbytes transferred in 192 seconds.
Client: Intel Pentium 4 2.6 GHz running Java JRE-based application (ComBlock Control Center). Direct cross-over LAN cable. No network connection. No other application running. COM-5002 or client computer configured as '10Base-T". 10Base-T connection.	4.82 Mbits/s 208Mbytes transferred in 345 seconds.

Format Conversion

Parallel to serial conversion occurs at the output when a 8-bit byte received over the TCP-IP link is converted to n-bit serial, where the sample width n is selected by the user. The key rule for parallel to serial conversion is that the most significant bit (MSb) is transmitted first.

Likewise, in the serial-to-parallel conversion which occurs at the input, the first received bit is placed at the MSb position in the byte.

Client Programming

This section is intended to help designers who want to design their own client application. It can be skipped by users of ready-to-use applications such as Hyperterminal, ComBlock Control Center, etc.

In network terminology, the COM-5002 is a server. It awaits connection establishment and connection termination under the initiation of clients. It never initiate any connection establishment or termination.

An example of C-language Winsock programming for Windows OS clients is shown below. More information about Winsock programming can be found at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winsock/winsock/finished_server_and_client_code.asp

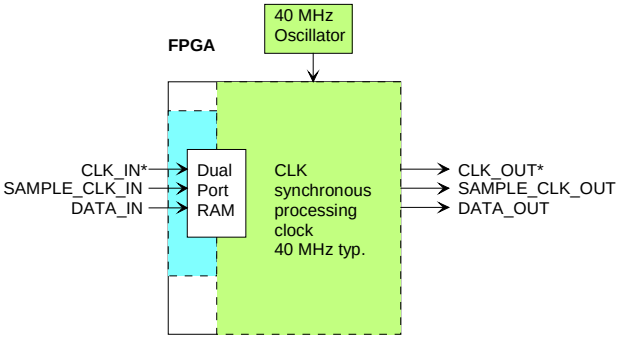
Be sure to include a reference to the Winsock2 library (WS2_32.lib) in the project release and/or debug settings.

```
#include <stdio.h>
#include "winsock2.h"

void main() {

    // Initialize Winsock.
    WSADATA wsaData;
    int iResult = WSAStartup( MAKEWORD(2,2), &wsaData );
    if ( iResult != NO_ERROR )
        printf("Error at WSAStartup()\n");

    // Create a socket.
    SOCKET m_socket;
    m_socket = socket( AF_INET, SOCK_STREAM, IPPROTO_TCP );

    if ( m_socket == INVALID_SOCKET ) {
        printf( "Error at socket(): %ld\n", WSAGetLastError() );
        WSACleanup();
        return;
    }

    // Connect to a server.
    sockaddr_in clientService;

    clientService.sin_family = AF_INET;
// insert destination address below
    clientService.sin_addr.s_addr = inet_addr( "172.16.1.128" );
// insert destination port below
    clientService.sin_port = htons(1024);

    if ( connect( m_socket, (SOCKADDR*) &clientService, sizeof(clientService) ) ==
SOCKET_ERROR) {
        printf( "Failed to connect.\n" );
        WSACleanup();
        return;
    }

    // Send and receive data.
    int bytesSent;
    int bytesRecv = SOCKET_ERROR;
    char sendbuf[32] = "Client: Sending data.";
    char recvbuf[32] = "";

    bytesSent = send( m_socket, sendbuf, strlen(sendbuf), 0 );
    printf( "Bytes Sent: %ld\n", bytesSent );

    while( bytesRecv == SOCKET_ERROR ) {
        bytesRecv = recv( m_socket, recvbuf, 32, 0 );
        if ( bytesRecv == 0 || bytesRecv == WSAECONNRESET ) {
            printf( "Connection Closed.\n");
            break;
        }
        if (bytesRecv < 0)
            return;
        printf( "Bytes Recv: %ld\n", bytesRecv );
    }

    return;
}
```

Timing

Clocks

The clock distribution scheme embodied in the COM-5002 is illustrated below.



Baseline clock architecture
Green = 40 MHz processing zone
Light blue = user defined input clock
**** indicates edge-trigger signal***

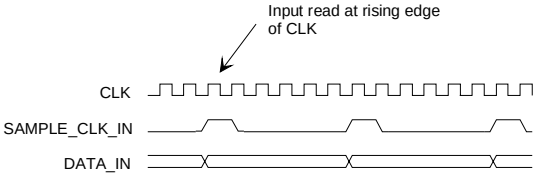
The core signal processing performed within the FPGA is synchronous with the processing clock $\mathbf{f_{clk}}$. The processing clock is derived from a 40 MHz oscillator. $\mathbf{f_{clk}}$ is not related to the external CLK_IN clock.

A 512-sample Dual-port RAM elastic buffer is used at the boundary between inputs and internal processing area. Thus, the input clocks frequencies can be independent from the internal processing clock frequency.

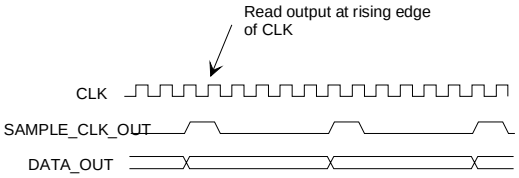
The input signals at the J2 connector are synchronous with the CLK_IN clock at J2/A1. This clock can be up to 40 MHz.

The output signals are synchronous with the rising edge of the 40 MHz reference clock CLK_OUT (i.e. all signals are stable at the rising edge of the reference clock CLK_OUT).

Input



Output



LEDs

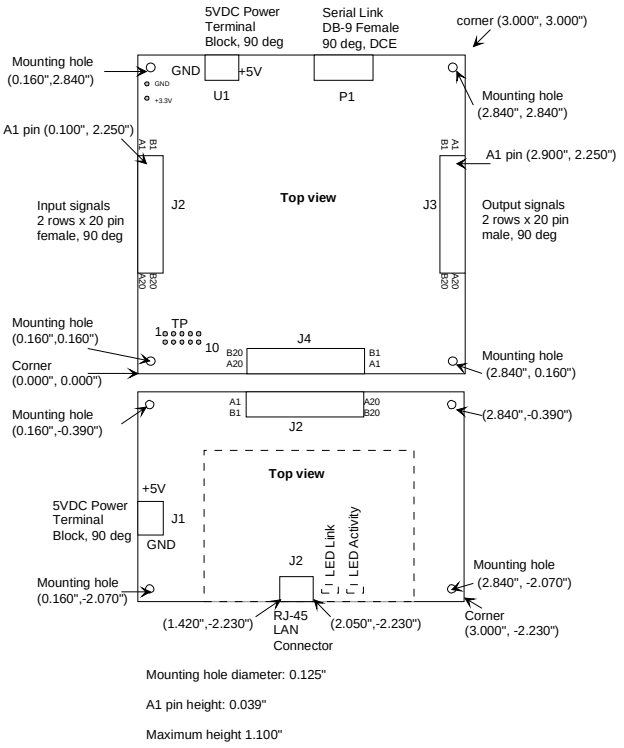
2 LEDs located close to the LAN RJ-45 jack provide summary information as to the LAN: Link and activity.

Test Points

Test points are provided for easy access by an oscilloscope probe. The main focus of these test points is to help monitor proper flow control operation.

Test Point	Definition
TP 1	TCP-IP connection on port 1024 (data stream) 1 = connected, 0 otherwise
TP 2	Future use
TP 3	TCP-IP connection on port 1028 (Monitoring & Control) 1 = connected, 0 otherwise
TP 4	Transmit data elastic buffer empty 1 = empty, 0 otherwise
TP 5	Transmit data elastic buffer full 1 = full, 0 otherwise
TP 6	Receive data elastic buffer empty 1 = empty, 0 otherwise
TP 7	Receive data elastic buffer more than half full 1 = more than half full, 0 otherwise
TP 8	Sample requests received through the J3 connector
TP9	Future use
TP10	Future use

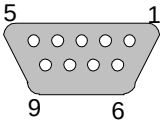
Mechanical Interface



Pinout

Serial Link P1

The DB-9 connector is wired as data circuit terminating equipment (DCE). Connection to a PC is over a straight-through cable. No null modem or gender changer is required.

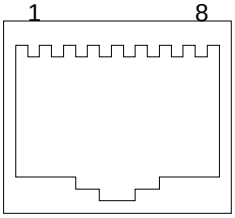


2 Transmit
3 Receive
5 Ground

DB-9 Female

LAN Connector J2

The RJ-45 Jack is wired as a standard PC network interface card. Connection to a LAN Hub is over a straight-through cable.



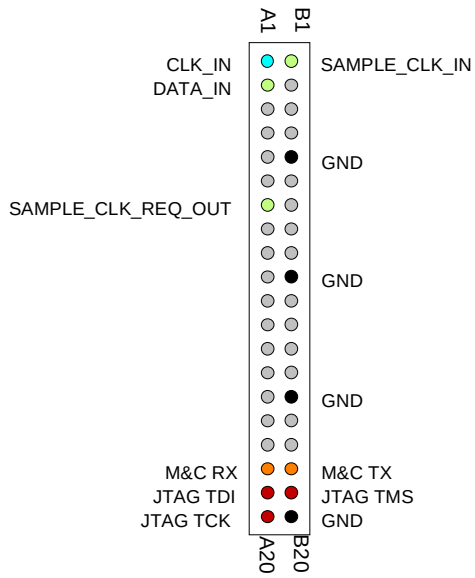
1 Tx+
2 Tx-
3 Rx+
6 Rx-

RJ-45 Jack

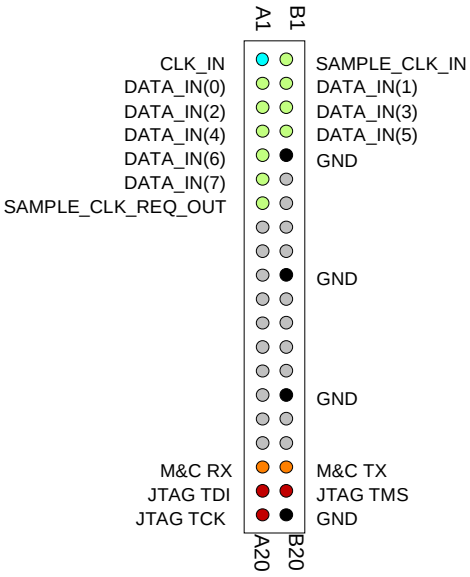
Input Connector J2

There are several possible connector configurations, depending on the application:

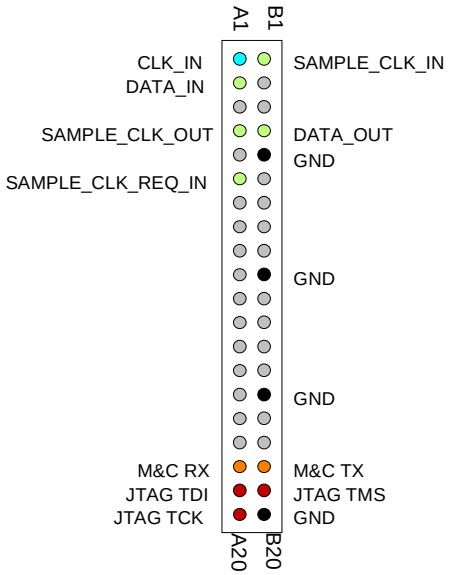
(a) 1-bit wide connection to another ComBlock [COM-1001, COM-1011,etc]

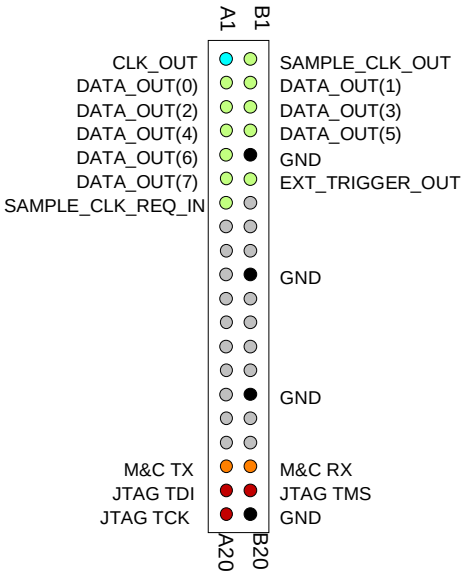


(b) 8-bit wide connection to another ComBlock [COM-8002,etc]



(c) Special case: input connector is used for bi-directional connection to COM-7001 module.

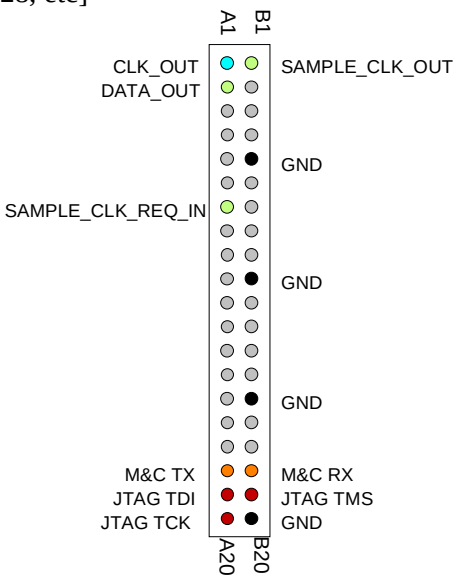




Output Connector J3

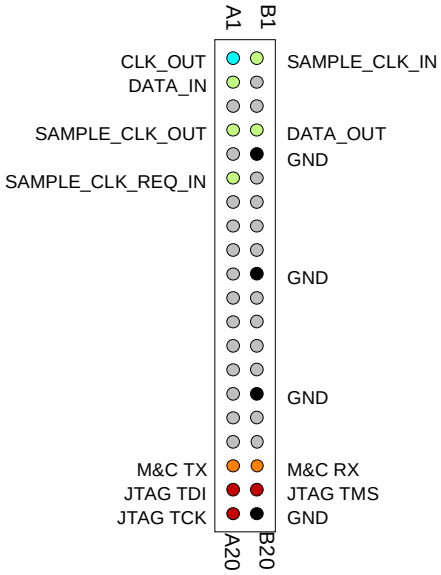
There are several possible connector configurations, depending on the application:

(a) 1-bit wide connection to another ComBlock [COM-1002, COM-1012, COM-1019, COM-1028, etc]



(b) 8-bit wide connection to another ComBlock [COM-8001, etc]

(c) Special case: output connector is used for bi-directional connection to COM-7001 module.



I/O Compatibility List

(not an exhaustive list)

Input	Output
COM-8002 Data acquisition module	COM-8001 Arbitrary Waveform Generator

Configuration Management

This specification is to be used in conjunction with VHDL software revision 6.

ComBlock Ordering Information

COM-5002 LAN/IP NETWORK INTERFACE

MSS • 18221 Flower Hill Way #A •
Gaithersburg, Maryland 20879 • U.S.A.
Telephone: (240) 631-1111
Facsimile: (240) 631-1676
E-mail: sales@comblock.com