

DYNAMIC ENGINEERING

435 Park Dr., Ben Lomond, Calif. 95005

831-336-8891 Fax 831-336-3840

<http://www.dyneng.com>

sales@dyneng.com

Est. 1988

User Manual

IP-Parallel-CDU

Digital Serial/Parallel Interface

IP Module

Revision A

Corresponding Hardware: Revision B

Corresponding Firmware: Revision A

10-2001-0102

IP-Parallel-CDU Digital Serial/Parallel Interface IP Module

Dynamic Engineering
435 Park Drive
Ben Lomond, CA 95005
831-336-8891
831-336-3840 FAX
www.dyneng.com

©2000-2004 by Dynamic Engineering.
IndustryPack is a registered trademark of GreenSpring
Computers Inc.
Other trademarks and registered trademarks are owned by their
respective manufactures.
Manual Revision A. Revised February 25, 2004

This document contains information of proprietary interest to Dynamic Engineering. It has been supplied in confidence and the recipient, by accepting this material, agrees that the subject matter will not be copied or reproduced, in whole or in part, nor its contents revealed in any manner or to any person except to meet the purpose for which it was delivered.

Dynamic Engineering has made every effort to ensure that this manual is accurate and complete. Still, the company reserves the right to make improvements or changes in the product described in this document at any time and without notice. Furthermore, Dynamic Engineering assumes no liability arising out of the application or use of the device described herein.

The electronic equipment described herein generates, uses, and can radiate radio frequency energy. Operation of this equipment in a residential area is likely to cause radio interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Dynamic Engineering's products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of Dynamic Engineering.

This product has been designed to operate with IP Module carriers and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.



Table of Contents

Product Description and Operation	6
Address Map	10
Programming	11
Register Definitions	12
CNTLO-485	12
CNTL1-485	12
CNTL2-485	12
Bit Mapping Table	12
Base_CNTL	13
INTerrupt Enable	15
INTerruptEdge_Lvl	16
INTerrupt Polarity	17
Data Input Filtered	18
Data Input Direct	19
Pre-Load Registers	20
Mask Registers	21
Read-Back Registers	22
Status Register	24
BIS_VECTOR	24
CDU specific registers	25
TX_DATA0	25
TX_DATA1	25
TIM_CNTL	26
RX_DATA1	27
RX_TX_CNT	28
Interrupts	29
ID PROM	30
IP-Parallel-CDU Logic Interface Pin Assignment	31
IP-Parallel-CDU Interface Pin Assignment.....	32
Applications Guide	33
Interfacing	33
Loop-back Connections	34
Construction and Reliability	34
Thermal Considerations	35
Warranty and Repair	35
Service Policy	36
Out of Warranty Repairs	36
For Service Contact:	36
Specifications	37





List of Figures

FIGURE 1	IP-PARALLEL-485 DATA IO CIRCUIT	6
FIGURE 2	IP-PARALLEL-485 DATA FILTER CIRCUIT	7
FIGURE 3	IP-PARALLEL-CDU INTERNAL ADDRESS MAP	10
FIGURE 4	IP-PARALLEL-485 CONTROL REGISTER 0 BIT MAP	12
FIGURE 5	IP-PARALLEL-485 CONTROL REGISTER 1 BIT MAP	12
FIGURE 6	IP-PARALLEL-485 CONTROL REGISTER 2 BIT MAP	12
FIGURE 7	IP-PARALLEL-CDU BASE CONTROL REGISTER BIT MAP	13
FIGURE 8	IP-PARALLEL-485 INTERRUPT ENABLE 0 BIT MAP	15
FIGURE 9	IP-PARALLEL-485 INTERRUPT ENABLE 1 BIT MAP	15
FIGURE 10	IP-PARALLEL-485 INTERRUPT EDG_LVL 0 BIT MAP	16
FIGURE 11	IP-PARALLEL-485 INTERRUPT EDG_LVL 1 BIT MAP	16
FIGURE 12	IP-PARALLEL-485 INTERRUPT POL 0 BIT MAP	17
FIGURE 13	IP-PARALLEL-485 INTERRUPT POL 1 BIT MAP	17
FIGURE 14	IP-PARALLEL-485 INTERRUPT DATAIN_FILO BIT MAP	18
FIGURE 15	IP-PARALLEL-485 INTERRUPT DATAIN_FIL1 BIT MAP	18
FIGURE 16	IP-PARALLEL-485 INTERRUPT DATAIN_DIRO BIT MAP	19
FIGURE 17	IP-PARALLEL-485 INTERRUPT DATAIN_DIR1 BIT MAP	19
FIGURE 18	IP-PARALLEL-485 PRE-LOAD LOWER BIT MAP	20
FIGURE 19	IP-PARALLEL-485 PRE-LOAD UPPER BIT MAP	20
FIGURE 20	IP-PARALLEL-485 COUNTER/TIMER A	21
FIGURE 21	IP-PARALLEL-485 MASK LOWER BIT MAP	21
FIGURE 22	IP-PARALLEL-485 MASK UPPER BIT MAP	21
FIGURE 23	IP-PARALLEL-485 READ-BACK LOWER BIT MAP	22
FIGURE 24	IP-PARALLEL-485 READ-BACK UPPER BIT MAP	22
FIGURE 25	IP-PARALLEL-485 COUNTER/TIMER B	23
FIGURE 26	IP-PARALLEL-CDU STATUS BIT MAP	24
FIGURE 27	IP-PARALLEL-CDU TX DATA REGISTER 0 BIT MAP	25
FIGURE 28	IP-PARALLEL-CDU TX DATA REGISTER 1 BIT MAP	25
FIGURE 29	IP-PARALLEL-CDU TIMING/CONTROL REGISTER BIT MAP	26
FIGURE 30	IP-PARALLEL-CDU RX DATA REGISTER 0 BIT MAP	27
FIGURE 31	IP-PARALLEL-CDU RX DATA REGISTER 1 BIT MAP	27
FIGURE 32	IP-PARALLEL-CDU TX/RX BIT COUNT REGISTER BIT MAP	28
FIGURE 33	IP-PARALLEL-CDU ID PROM	30
FIGURE 34	IP-PARALLEL-CDU LOGIC INTERFACE	31
FIGURE 35	IP-PARALLEL-CDU INTERFACE	32



Product Description and Operation

The IP-Parallel-CDU is part of the IP Module family of modular I/O components. It is based on the IP-Parallel-485, which has 24 RS-485 differential pairs for data output or input. See figure 1 below.

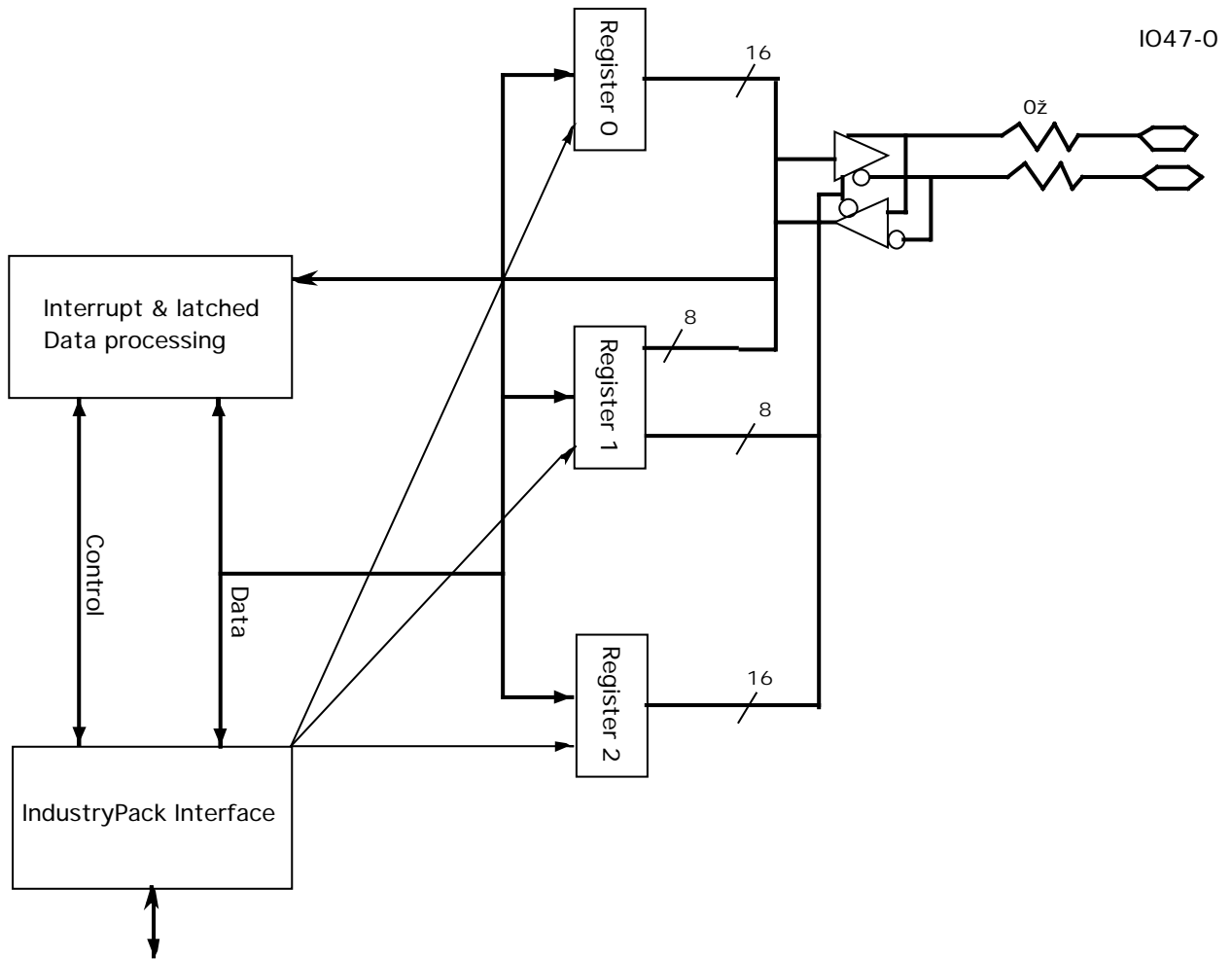


FIGURE 1

IP-PARALLEL-485 DATA IO CIRCUIT

The direction and output data value of each IO line is controlled by the FPGA [Xilinx SpartanXL40]. If the IO line is configured as an input, the data values are available to be read from the IP address space as a direct read or after filtering.



To filter the data, each channel has an enable, sense, and edge/level bit associated with it. The enable will block or enable a particular channel from being received into the filtering logic. The sense controls whether the inverted or non-inverted received data is used. The edge or level control will make the hold circuit wait for an edge from $0 > 1$ (or $1 > 0$ if inverted) or react to a level. The hold circuit captures data and holds it until read. See figure 2 below.

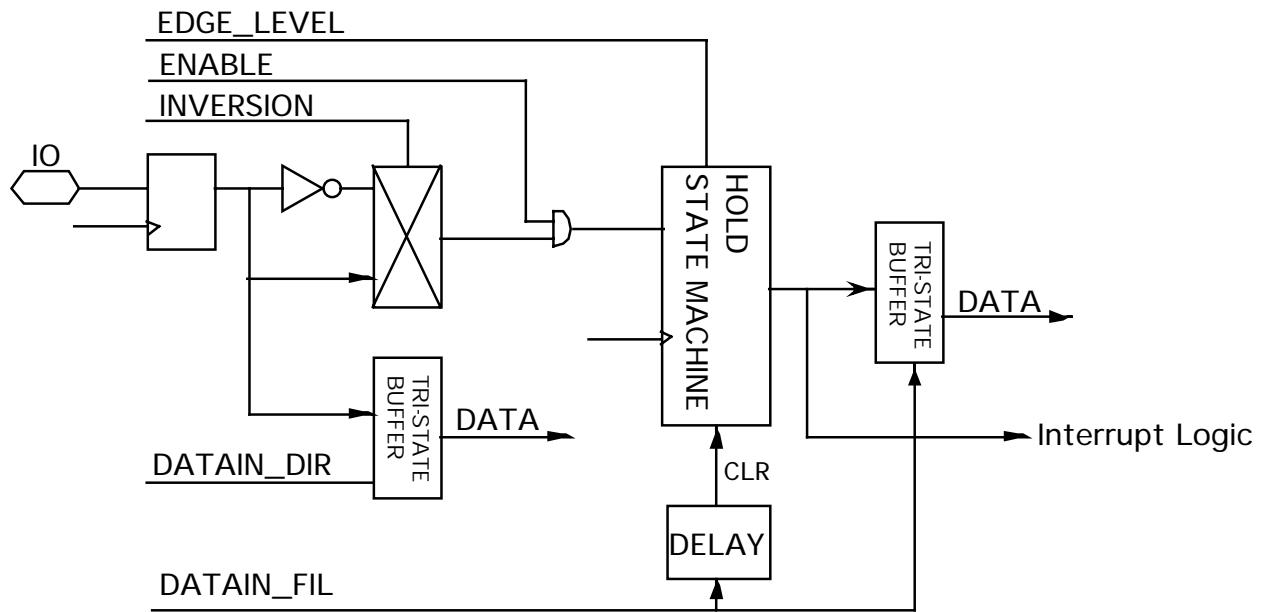


FIGURE 2

IP-PARALLEL-485 DATA FILTER CIRCUIT

The data is registered at the chip edge and then again after the enable and inversion circuitry. Each channel has a separate hold circuit. If a signal is detected to meet the filtering criteria, then the signal is held as a high until the data is read. Each channel can be programmed to detect a high, low, rising edge, or falling edge when the channel has something of interest. The registers are referenced to the IP clock and operate at 8 or 32 MHz depending on the slot configuration. Each group of 16 channels has a separate read clear signal. The channels can be read in any order and not loose data. The circuit will capture pulses down to 2 reference clocks wide. 62.5 nS (32 MHz IP clock) or 250 nS (8 MHz) with the standard IP reference clock.

The active high signals are combined to create an interrupt request based on the captured and held data. If the master interrupt is enabled, then the interrupt is passed onto the system. Reading the filtered data or disabling the master



interrupt clears the interrupt. The user can program each channel to use the edge or level condition. The edge is particularly useful for long duration signal where repeated interrupts are not desired. The alternate approach is to flip the sense bit and create an interrupt when the signal has switched to the opposite polarity. Instruction order is important. Once the interrupt is detected the sense needs to be switched before the interrupt is re-enabled or a second interrupt is likely to be generated. It is recommended to read from the filtered data path after the processing parameters are changed to clear any interrupts that are created by the changing filter parameters. With each IO line having all three controls plus independent direction many control possibilities exist.

For ease of use the control registers are mapped to group the direction bits and the data bits together at the upper and lower ends of the control words [cntIO-2] respectively. The space is divided in half the lower 24 bits are data and the upper 24 control.

When the IP-Parallel-CDU is powered-on it defaults to a dedicated serial interface that uses four IO lines to implement a full-duplex clock and data protocol. The transmit interface defaults to 32-bit LSB first at 500 kHz bit rate. Both the clock rate and the number of bits are programmable, but these nominal values are in effect on power-up.

The receive interface uses an external clock at a nominal rate of 614 kHz, but will function over a wide range of frequencies. The receive words are also 32-bits LSB first, although the word size is programmable.

The IP-Parallel-CDU supports both 8 and 32 MHz. IP Bus operation. All configuration registers support read and write operations for maximum software convenience. The ID, IO, and INT spaces are utilized by the IP-Parallel-CDU design.

The IP-Parallel-CDU conforms to the VITA standard. This guarantees compatibility with multiple IP Carrier boards. Because the IP may be mounted on different form factors, while maintaining plug and software compatibility, system prototyping may be done on one IP Carrier board, with final system implementation on a different one. The PCI3IP card makes a convenient development platform in many cases.
http://www.dyneng.com/pci_3_ip.html

Frequently it is necessary to correlate the time and the event. The IP-Parallel design supports an enhanced MC68230 capability with two - 32 bit counter-



timers. The counter-timers are easy to use with a minimum of registers to access and complete independence. The IP clock is used as a reference.

Counter/Timer A features a 32 bit down-counter with a pre-load register. The counter output is tested against a "zero" value. When zero the counter is re-loaded with the pre-load value to create a cycle. At each zero detection an interrupt can be generated. At each zero detection a waveform can be transitioned. The waveform can be enabled onto the upper data bit.

Counter / Timer B has a 32 bit up counter which can be cleared by the software. The counter output is masked with a user programmable value to select a particular counter bit or bits to use for interrupt creation. The counter output is also available to read via software and can serve as a real-time clock.

Interrupts are supported by the IP-Parallel-CDU. The counter/timer interrupts occur when the programmed conditions match the counter/timer counts, the data interrupt occurs when a programmed transition occurs, and the Rx interrupt is asserted when a serial Rx word is received. The interrupt conditions are individually maskable. The interrupt occurs on IntReq0. The vector is user programmable by a read/write register and can be read in the IO space or automatically when the INT space is accessed.



Address Map

Function	Offset	Width	Type
Registers for CDU interface – Active when BASE_CONTROL bit 5 = '0'			
TX_DATA0	0x00	2 bytes	read/write
TX_DATA1	0x02	2 bytes	read/write
TIM_CNTL	0x04	2 bytes	read/write
RX_DATA0	0x08	2 bytes	read only
RX_DATA1	0x0A	2 bytes	read only
Registers for CDU interface – Always Active			
RX_TX_CNT	0x00C	2 bytes	read/write
Registers for IpParallel-485 – Active when BASE_CONTROL bit 5 = '1'			
CNTLO	0x00	word	read/write
CNTL1	0x02	word	read/write
CNTL2	0x04	word	read/write
INTENO	0x08	word	read/write
INTEN1	0x0A	word	read/write
Registers for IpParallel-485 – Always Active			
BASE_CONTROL	0x06	word	read/write
VECTOR	0x0E	byte	read/write
INTEGLVO	0x10	word	read/write
INTEGLV1	0x12	word	read/write
INTPOLO	0x18	word	read/write
INTPOL1	0x1A	word	read/write
STATUS	0x1E	word	read/write clear
DATFILO	0x20	word	read only
DATFIL1	0x22	word	read only
DATDIRO	0x28	word	read only
DATDIR1	0x2A	word	read only
PRLD_L	0x2E	word	read/write
PRLD_U	0x30	word	read/write
MASK_L	0x32	word	read/write
MASK_U	0x34	word	read/write
RDBK_L	0x36	word	read only
RDBK_U	0x38	word	read only

FIGURE 3

IP-PARALLEL-CDU INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within the IP-Parallel-CDU. The addresses are all offsets from a base address. The carrier board that the IP is installed into provides the base address.



Programming

Programming the IP-Parallel-CDU requires only the ability to read and write data in the host's I/O space. The IP Carrier board determines the base address of its IP modules.

In order to receive data the software is only required to read from the "Direct" port. Alternatively the filtered data path can be programmed with the enable, polarity, level and edge and then the Filtered data used. If desired, the interrupt can be enabled and the interrupt vector written to the vector register.

A typical sequence would be to first write to the vector register with the desired interrupt vector. Please note that some carrier boards do not use the interrupt vector. The interrupt service routine should be loaded and the mask set. The Level and Edge conditions programmed then the enables set to receive data. The incoming data can be pulsed. The hardware will hold any pulse or level detected until the software reads the filter data registers.

This version of the IP-Parallel-IO has all 24 485 transceivers installed, each of which can be programmed to be either an input or an output. All 24 IO bits are connected to the direct and filtered data port regardless as to whether they are configured as inputs or outputs, therefore care must be taken to disregard the output bits when looking for changes to the input lines.

A 32-bit write/read with some CPUs will result in two 16-bit accesses to the hardware with automatic incrementing addresses. This 32-bit access is quite a bit faster than a software loop. The PCI3IP, PCI5IP, and cPCI2IP are examples of carriers that support 32-bit to 16-bit mapping automatically. The lower 32 bits of the output, "data in filtered", and "data in direct" are on long word boundaries to utilize this feature if available.

http://www.dyneng.com/pci_3_ip.html

<http://www.dyneng.com/pci5ip.html>

<http://www.dyneng.com/cpci2ip.html>



Register Definitions

CNTLO-485

\$00 Parallel Control Register Port read/write

CONTROL REGISTER 0	
DATA BIT	DESCRIPTION
15-0	485 data [I015-0]

FIGURE 4

IP-PARALLEL-485 CONTROL REGISTER 0 BIT MAP

CNTL1-485

\$02 Parallel Control Register Port read/write

CONTROL REGISTER 1	
DATA BIT	DESCRIPTION
7-0	485 data [I023-16]
15-8	485 Direction [I07-0]

FIGURE 5

IP-PARALLEL-485 CONTROL REGISTER 1 BIT MAP

CNTL2-485

\$04 Parallel Control Register Port read/write

CONTROL REGISTER 2	
DATA BIT	DESCRIPTION
15-0	485 direction [I023-8]

FIGURE 6

IP-PARALLEL-485 CONTROL REGISTER 2 BIT MAP

Bit Mapping Table

"IP-Parallel-485" 24 – RS-485 I/O lines

Control Bits	Bit#/Definition
47-0	47-24/485 direction control 23-0/485 data bits [I023-0]



Base_CNTL

\$06 BISERIAL Control Register Port read/write

CONTROL REGISTER BASE	
DATA BIT	DESCRIPTION
15	spare
14	hold timerB 0 = disabled, 1 = hold
13	clear timerB 0 = run, 1 = clear
12	interrupt enable timerB 1 = enabled
11	spare
10	square wave output 1 = output on msb
9	timerA load 0 = run, 1 = load
8	interrupt enable timerA 1 = enabled
7	spare
6	clock 32, 1 = 32 MHz IP clock, 0 = 8 MHz
5	parallel enable, 0 = CDU interface enabled
4,3	spare
2	force interrupt 1 = force
1	master interrupt enable 1 = enabled
0	output register control 1 = enabled

FIGURE 7

IP-PARALLEL-CDU BASE CONTROL REGISTER BIT MAP

0. Output Register Control is used to control when the three Control registers values are placed onto the output registers. If synchronization is needed set to '0' until the registers are written and then enable ['1']. The output bits will then be driven to the new state at the same time. Referenced to the IP Clock. If the bit is left in the '0' state, then the new control register values will not be output and the data will stay in the previous state. If the bit is left in the '1' state then the control outputs will change when the registers are independently updated.

1. INT_EN is the master interrupt enable. Default is 0. If set to 1 then if one or more of the filtered input data or timer interrupt conditions is met an interrupt will be generated on level 0.

2. Force Interrupt is used to create an interrupt for test and software development purposes. Set the bit to cause an interrupt and clear the bit to remove the interrupt. The IO bits can be used for the same purpose if the filter controls are properly set. Requires INT_EN to be enabled.

5. Parallel enable is used to enable the CDU full-duplex serial interface registers and IO bits (IO bits 6, 8, 18, and 20). The bit defaults to zero, which enables this



interface. When set to a one, the regular parallel interface is enabled

6. Clock 32 must be set to the proper value, matching the IP clock frequency being used, in order for the proper default frequency to be produced.

8. Interrupt Enable TimerA. When enabled and the Counter/Timer A is active the counter timer will create an interrupt stream. The period of the interrupts is determined by the pre-load register. The interrupt is generated when the Counter/Timer A counts down to "0". Requires INT_EN to be enabled.

9. Counter/Timer A load when '1' loads the value in the preload register into the down counter. When '0' the counter decrements until the terminal count of "0" is reached at which point the counter re-loads the pre-load value and repeats the cycle. The load bit is not needed because the counter will eventually roll over and reload anyway. If a large count was previously loaded or if the software wants to have a consistent period from a known point in time then the load bit is used.

10. WaveOut when enabled selects the waveform generated by Counter/Timer A to output on the MSBit instead of the data bit for that position. 1 = waveform, 0 = databit. The waveform is a square wave which is switched at each zero crossing – has 2x period of the interrupts generated by Counter/Timer A.

12. Interrupt Enable Counter/Timer B when '1' creates an interrupt stream based on the second counter timer. Counter/Timer B is a 32 bit up counter. The counter output is masked with the value in the mask register to pick off the period you desire for the interrupt stream. For example: with the IP clock set to 8 Mhz and the mask set to a 0x00000040 the 7th bit would be selected for a divide by $64 * 2 * 125 = 16 \text{ uS}$ period.

13. Clear Counter/Timer B when '1' forces the second counter timer to 0. This function is useful to restart the "real time clock" to a known value at a known time.

14. Hold Timer when '1' will stop the Counter/Timer B from updating the read-back register. Counter/Timer B will continue to run. The register will be stable allowing the two reads of the register data to happen without the stored count changing. "0" = update the counter read-back register.



INTerrupt Enable

Int_en0 \$08 Parallel Control Register Port read/write

DATA BIT	Interrupt Enable DESCRIPTION
15-0	int_en 15-0 Interrupt Enable 1 = enabled, 0 = disabled

FIGURE 8

IP-PARALLEL-485 INTERRUPT ENABLE 0 BIT MAP

Int_en1 \$0A Parallel Control Register Port read/write

DATA BIT	Interrupt Enable DESCRIPTION
15-8	spare
7-0	int_en 23-16 Interrupt Enable 1 = enabled, 0 = disabled

FIGURE 9

IP-PARALLEL-485 INTERRUPT ENABLE 1 BIT MAP

The data bits correspond to the IO lines. In the filtered path if the control register bit is set to 1 then the corresponding IO line is enabled to be a potential interrupter and to be captured by the hold circuit. The enable is applied after the inversion control. In 485 mode the bits corresponding to the direction control should be set to '0'.



INTerruptEdge_Lvl

Edg_Lvl 0 \$10 Parallel Control Register Port read/write

DATA BIT	EDGE_LVL	DESCRIPTION
15-0		Edg_Lvl 15-0 1 = edge, 0 = level

FIGURE 10

IP-PARALLEL-485 INTERRUPT EDG_LVL 0 BIT MAP

Edg_Lvl 1 \$12 Parallel Control Register Port read/write

DATA BIT	EDGE_LVL	DESCRIPTION
15-8		spare
7-0		Edg_Lvl 23-16 1 = edge, 0 = level

FIGURE 11

IP-PARALLEL-485 INTERRUPT EDG_LVL 1 BIT MAP

The data bits correspond to the IO lines. In the filtered path if the control register bit is set to 1 then the corresponding IO line is captured only if there is a transition from '0' to '1'. If set to '0' then anytime the IO line is detected to be '1' the hold circuit will be set. The hold circuit will retain the data until read by the corresponding data_in_fi(x) is accessed. The hold circuits are after the enable and inversion in the pipeline.



INTerrupt Polarity

Pol 0 \$18 Parallel Control Register Port read/write

DATA BIT	Polarity	DESCRIPTION
15-0		POL 15-0 1 = invert, 0 = not inverted

FIGURE 12

IP-PARALLEL-485 INTERRUPT POL 0 BIT MAP

Pol 1 \$1A Parallel Control Register Port read/write

DATA BIT	Polarity	DESCRIPTION
15-8		spare
7-0		POL 23-16 1 = invert, 0 = not inverted

FIGURE 13

IP-PARALLEL-485 INTERRUPT POL 1 BIT MAP

The data bits correspond to the IO lines. In the filtered path if the control register bit is set to 1 then the corresponding IO line is inverted. If set to '0', then no inversion is applied.



Data Input Filtered

Datain_filo \$20 Parallel Control Register Port read/write

DATA BIT	Filtered Data DESCRIPTION
15-0	DATAIN_FIL 15-0

FIGURE 14

IP-PARALLEL-485 INTERRUPT DATAIN_FILO BIT MAP

Datain_fil1 \$22 Parallel Control Register Port read/write

DATA BIT	Filtered Data DESCRIPTION
15-8	spare
7-0	DATAIN_FIL 23-16

FIGURE 15

IP-PARALLEL-485 INTERRUPT DATAIN_FIL1 BIT MAP

The data bits correspond to the IO lines after the filters have been applied. The data remains latched until the register is read. The three registers are independent for reading and clearing purposes. Read [clear] the registers after any control change to insure that no false positives are reported.



Data Input Direct

Datain_dir0 \$28 Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-0	DATAIN_DIR 15-0

FIGURE 16

IP-PARALLEL-485 INTERRUPT DATAIN_DIRO BIT MAP

Datain_dir1 \$2A Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-8	spare
7-0	DATAIN_DIR 23-16

FIGURE 17

IP-PARALLEL-485 INTERRUPT DATAIN_DIR1 BIT MAP

The data bits correspond to the IO lines without filters being applied. The data is a direct reflection of the current state of the IO lines. Metastable protection registers are in place but no hold registers.



Pre-Load Registers

Pre_load_l \$2E Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-0	pre-load 15-0

FIGURE 18

IP-PARALLEL-485 PRE-LOAD LOWER BIT MAP

Pre_load_u \$30 Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-0	pre_load 31-16

FIGURE 19

IP-PARALLEL-485 PRE-LOAD UPPER BIT MAP

The pre-load registers are combined internally to form a 32 bit pre-load value to use with Counter/Timer A. Counter/Timer A is loaded with the value in the Pre-Load registers when the counter reaches zero. The counter can also be loaded with the software command via the Base register. The counter will count from the value down to zero creating an N+1 total count. The counter will re-load the value on the next count.

The Counter Timer A has the option of creating an interrupt at each zero crossing – period N+1. The wave out option can also be enabled. The waveform is generated by switching each time the zero count is detected. A square wave is generated with a period of 2(N+1). The reference period is the IP clock which can be 125 nS or 31.25 nS depending on the carrier selection made.

Because of the architecture the period selected with Counter/Timer A is arbitrary. Most OS/CPU can not handle interrupts faster than 10 uS. The counter output is registered and then checked against a count of 2 and re-registered. The count value checked is '2' to account for the double pipeline delay. The counter will be at "0" when the check reaches "2". Because of the checking scheme the minimum count needs to be larger than '2' or the hardware will miss the first "2" and not see it until the counter rolls over. At 8 MHz, The roll over time is approximately 9 minutes.



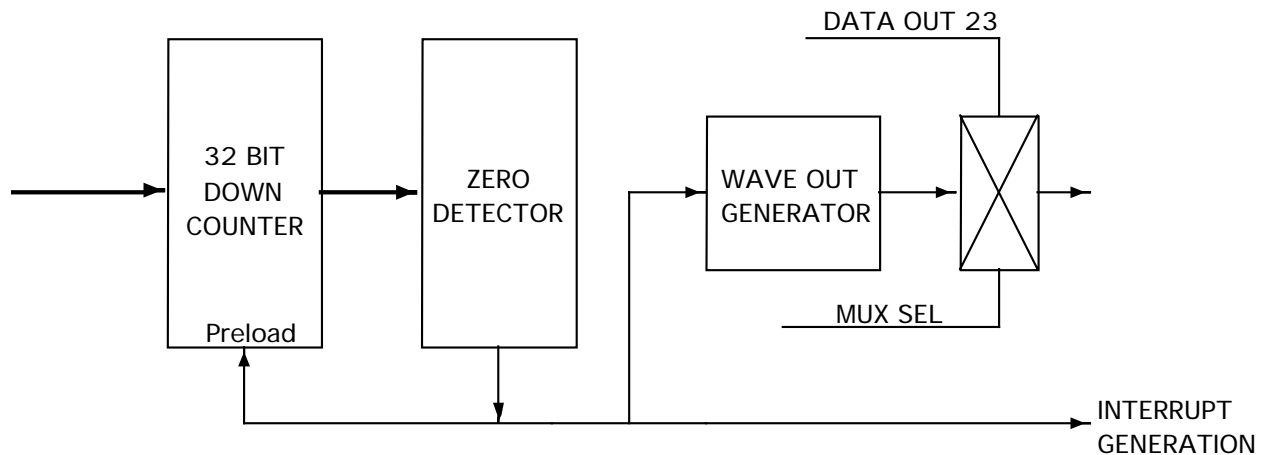


FIGURE 20

IP-PARALLEL-485 COUNTER/TIMER A

Mask Registers

MASK_L \$32 Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-0	mask 15-0

FIGURE 21

IP-PARALLEL-485 MASK LOWER BIT MAP

MASK_U \$34 Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-0	mask 31-16

FIGURE 22

IP-PARALLEL-485 MASK UPPER BIT MAP

The mask registers are combined internally to form a 32 bit mask value to use with Counter/Timer B. Counter/Timer B is a 32 bit up counter which can be cleared to '0'. The counter rolls over at the maximum count. Each counter bit is masked with the mask registers corresponding position. If the Mask(n) = '1' and



the counter(n) is also a '1' then the output will be '1'. The output from the AND array is or'd to determine if any of the counter bit – mask combinations are active. When a counter bit becomes active and the mask is set an interrupt is generated. The interrupt will be generated on the period of the counter selected with the mask. The counter uses the IP clock as a reference. The period selected is $2 * 2^n * [125 \text{ or } 31.25]$. For example a 0x40 and 8 MHz reference rate would create a periodic interrupt with a period of 16 μ S. [$2 * 64 * 125 \text{ nS}$]

Read-Back Registers

Rdbk_l \$32 Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-0	read-back 15-0

FIGURE 23

IP-PARALLEL-485 READ-BACK LOWER BIT MAP

Rdbk_u \$34 Parallel Control Register Port read/write

DATA BIT	Direct Data DESCRIPTION
15-0	read-back 31-16

FIGURE 24

IP-PARALLEL-485 READ-BACK UPPER BIT MAP

Counter/TimerB has a read-back port, which allows the count to be read by the host. The count is pipelined and stored into an output register. The register is updated with each count unless the Hold [see base register] bit is set. When the Hold bit is set the output register is disabled from updating; Counter/TimerB continues to count. The register is 32 bits wide. Two reads are required to get the entire 32-bit word via the 16 bit IP data bus. If only the lower or upper words are of interest then the Hold bit can be ignored.



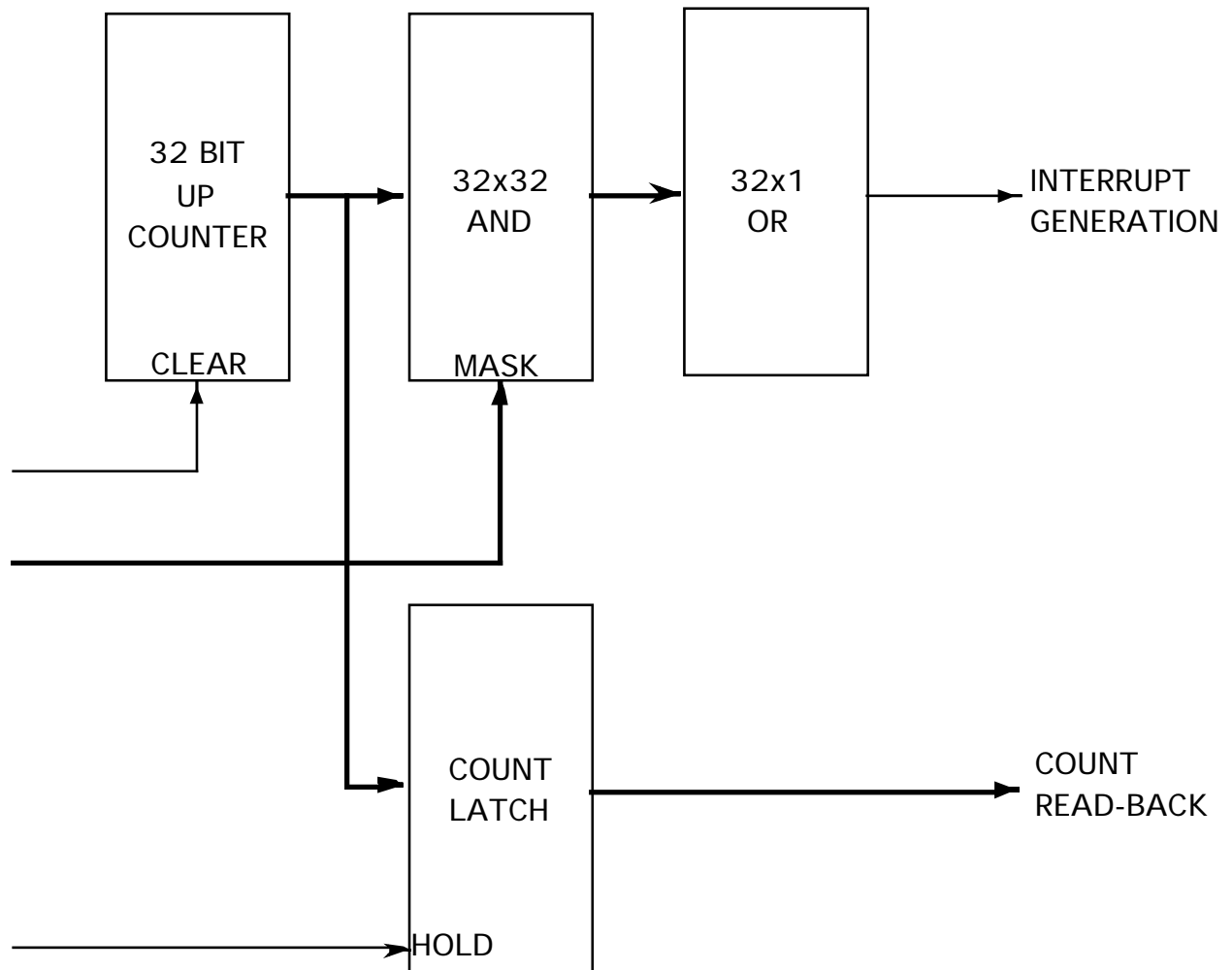


FIGURE 25

IP-PARALLEL-485 COUNTER/TIMER B



Status Register

Status \$1E Parallel Status read/write

DATA BIT	Interrupt Status	DESCRIPTION
15-8		spare
7	INTRO	1= interrupt condition active
6-4		spare
3	RX_INTR	1 = word received
2	INTR_DAT	1 = active IO Bit
1	INTRB	1 = active Counter/TimerB
0	INTRA	1 = active Counter/TimerA

FIGURE 26

IP-PARALLEL-CDU STATUS BIT MAP

When the interrupt status is set ('1') then the interrupt event has occurred. If the Interrupt Enable corresponding to the interrupt type and master interrupt enable are set then an interrupt will occur. If the Enable is not set then the interrupt will not be asserted and the status can be used to poll. The Counter/Timer and CDU Rx interrupts are cleared by writing to the status register with the corresponding bit(s) set. Writing to 15-4, 2 will have no affect. The write is transitory – no clear or re-enable is required. The data interrupt is cleared by reading the data from the filtered data ports.

INTRO is the state of the interrupt condition before the master enable. This bit is useful if operating in a polled mode.

BIS_VECTOR

\$OE Parallel Interrupt Vector Port

The Interrupt vector for the IP-Parallel-CDU is stored in this byte wide register. This read/write register is initialized to 'xFF' upon power-on reset. The vector is stored in the odd byte location [D7..0]. The vector should be initialized before the interrupt is enabled or the mask is lowered. The interrupt is automatically cleared when the CPU acknowledges the interrupt.



CDU specific registers

TX_DATA0

\$00 Transmit Data Register Port read/write (byte accessible)

CONTROL REGISTER 0		
DATA BIT	DESCRIPTION	
15-8	TX data [15-8]	byte select 1
7-0	TX data [7-0]	byte select 0

FIGURE 27

IP-PARALLEL-CDU TX DATA REGISTER 0 BIT MAP

TX_DATA1

\$02 Transmit Data Register Port read/write (byte accessible)

CONTROL REGISTER 0		
DATA BIT	DESCRIPTION	
15-8	TX data [31-24]	byte select 1
7-0	TX data [23-16]	byte select 0

FIGURE 28

IP-PARALLEL-CDU TX DATA REGISTER 1 BIT MAP

These two registers allow the Transmit data word to be specified. Each byte is individually accessible. Alternatively a 16 or 32-bit word can be written.



TIM_CNTL

\$04 CDU Timing Control Register Port read/write (byte accessible)

CONTROL REGISTER BASE	
DATA BIT	DESCRIPTION
15-8	Clock divide-by count
7	spare
6	receive interrupt enable – 1 = enabled
5	test mode – 1 = test mode enabled, 0 = normal
4	clear receive bit count – 1 = clear count
3	receiver enable – 1 = enabled
2	clear receive data
1	transmit start/busy
0	transmit inhibit 1 = inhibit shift

FIGURE 29

IP-PARALLEL-CDU TIMING/CONTROL REGISTER BIT MAP

This is the control/timing register for the CDU interface. The clock divisor field defaults to a frequency of 500 kHz if left reset or set to zero. The clock 32 bit in the base control register communicates the operational IP frequency, in order for the clock generator to produce the default frequency. If this field is not zero, then it determines how many IP clock counts the clock generator waits before transitioning to the opposite level. Therefore the resulting frequency is determined by the following formula: $F_{out} = F_{ip} / 2^n$ where F_{ip} is the IP frequency, n is the count, and F_{out} is the resulting clock frequency.



RX_DATA0

\$08 Receive Data Register Port read only (byte accessible)

CONTROL REGISTER 0		
DATA BIT	DESCRIPTION	
15-8	TX data [15-8]	byte select 1
7-0	TX data [7-0]	byte select 0

FIGURE 30

IP-PARALLEL-CDU RX DATA REGISTER 0 BIT MAP

RX_DATA1

\$0A Receive Data Register Port read only (byte accessible)

CONTROL REGISTER 0		
DATA BIT	DESCRIPTION	
15-8	TX data [31-24]	byte select 1
7-0	TX data [23-16]	byte select 0

FIGURE 31

IP-PARALLEL-CDU RX DATA REGISTER 1 BIT MAP

These two registers allow the Received data word to be read. Each byte is individually accessible. Alternatively a 16 or 32-bit word can be read.



RX_TX_CNT

\$SOC Transmit/Receive Bit Count Register Port read/write (byte accessible)

CONTROL REGISTER	
DATA BIT	DESCRIPTION
15-13	spare
12-8	RX bit count byte select 1
7-5	spare
4-0	TX bit count byte select 0

FIGURE 32

IP-PARALLEL-CDU TX/RX BIT COUNT REGISTER BIT MAP

The Rx and Tx bit counts when reset or set to zero default to 32 bits, otherwise the bit count specified (1-31) is used.



Interrupts

All IP Module interrupts are vectored. The vector from the IP-Parallel-CDU comes from a vector register loaded as part of the initialization process. The vector register can be programmed to any 8-bit value. The default value is \$FF which is sometimes not a valid user vector. The software is responsible for choosing a valid user vector.

The IP-Parallel-CDU state machines generate an interrupt request when a programmed condition is detected on the IO lines. The interrupt is mapped to interrupt request 0. The CPU will respond by asserting INT. The hardware will automatically supply the appropriate interrupt vector when accessed by the CPU. The interrupt type can be read from the Status register. There are four interrupt types; Data IO, Counter/Timer A, Counter/Timer B and CDU Rx. If more than one type of interrupt is in use, then the status register should be read first to determine which interrupt types are active. The exception handler can then respond to all of the current interrupt requests. The Data IO interrupt source line(s) are obtained by reading DATA_IN_FILO-1. The status remains valid until the registers are read. The interrupt status is auto-cleared when the registers are accessed.

Some carrier boards pre-fetch data. If your carrier board pre-fetches the interrupt status, then the status may be cleared when the SW goes to look at it. If this is an issue then be careful with the order of reading the registers to prevent the pre-fetching function from affecting operation.

The interrupt level seen by the CPU is determined by the IP Carrier board being used. The master interrupt and Counter/Timer interrupts can be disabled or enabled through the BASE_CNTL register. The CDU Rx interrupt is enabled in the CDU Timing/Control register. These enables operate after the interrupt status latch. The individual enables for IO lines are controllable through INT_ENO-1. The enable operates before the interrupt holding latch, which stores the request for the CPU. Once the interrupt request is set, the way to clear the request is to read the holding register [DATAIN_FILO-1], reset the board, or disable the interrupt. The Interrupt acknowledge cycle fetches the vector, but does not clear the interrupt request in this design.

If operating in a polled mode and making use of the interrupts for status then the master interrupt should be disabled.



ID PROM

Every IP contains an ID PROM, whose size is at least 32 x 8 bits. The ID PROM aids in software auto configuration and configuration management. The user's software, or a supplied driver, may verify that the device it expects is actually installed at the location it expects, and is nominally functional. The ID PROM contains the manufacturing revision level of the IP. If a driver requires that a particular revision to be present, it may check for it directly.

The location of the ID PROM in the host's address space is dependent on which carrier is used.

Standard data in the ID PROM on the IP-Parallel-CDU is shown in the figure below. For more information on IP ID PROMs refer to the IP Module Logic Interface Specification, available from Dynamic Engineering.

Each of the modifications to the IP-Parallel-IO board will be recorded with a new code in the DRIVER ID and reserved fields.

Address		Data
01	ASCII "I"	\$49
03	ASCII "P"	\$50
05	ASCII "A"	\$41
07	ASCII "H"	\$48
09	Manufacturer ID	\$1E
0B	Model Number	\$03
0D	Revision	\$A0
0F	reserved	\$00
11	Driver ID, low byte	\$00
13	Driver ID, high byte	\$06
15	No of extra bytes used	\$0C
17	CRC	\$C8

FIGURE 33

IP-PARALLEL-CDU ID PROM



IP-Parallel-CDU Logic Interface Pin Assignment

The figure below gives the pin assignments for the IP Module Logic Interface on the IP-Parallel-CDU. Pins marked n/c below are defined by the specification, but not used on the IP-Parallel-CDU. Also see the User Manual for your carrier board for more information.

GND		GND		1	26	
CLK		+5V		2	27	
Reset*		R/W*		3	28	
DO		IDSEL*		4	29	
D1		DMAReq0*		5	30	
D2		MEMSEL*		6	31	
D3		DMAReq1*		7	32	
D4		IntSel*		8	33	
D5		DMAAck*		9	34	
D6		IOSEL*		10	35	
D7		n/c		11	36	
D8		A1		12	37	
D9		DMAEnd*		13	38	
D10		A2		14	39	
D11		n/c		15	40	
D12		A3		16	41	
D13		IntReq0*		17	42	
D14		A4		18	43	
D15		IntReq1*		19	44	
BS0*		A5		20	45	
BS1*		n/c		21	46	
n/c		A6		22	47	
n/c		Ack*		23	48	
+5V		n/c		24	49	
GND		GND		25	50	

NOTE 1: The no-connect signals above are defined by the IP Module Logic Interface Specification, but not used by this IP. See the Specification for more information.

NOTE 2: The layout of the pin numbers in this table corresponds to the physical placement of pins on the IP connector. Thus this table may be used to easily locate the physical pin corresponding to a desired signal. Pin 1 is marked with a square pad on the IP Module.

FIGURE 34

IP-PARALLEL-CDU LOGIC INTERFACE



IP-Parallel-CDU Interface Pin Assignment

The figure below gives the pin assignments for the IP Module IO Interface on the IP-PARALLEL-485. Pins marked. Also see the User Manual for your carrier board for more information. There are 24 pairs of differential signals. IO23-0. See CNTLO-2 and the bit-mapping table for more information.

IO_0+	IO_12+		1	26	
IO_0-	IO_12-		2	27	
IO_1-	IO_13-		3	28	
IO_1+	IO_13+		4	29	
IO_2+	IO_14+		5	30	
IO_2-	IO_14-		6	31	
IO_3-	IO_15-		7	32	
IO_3+	IO_15+		8	33	
IO_4+	IO_16+		9	34	
IO_4-	IO_16-		10	35	
IO_5-	IO_17-		11	36	
IO_5+	IO_17+		12	37	
IO_6+	IO_18+		13	38	
IO_6-	IO_18-		14	39	
IO_7-	IO_19-		15	40	
IO_7+	IO_19+		16	41	
IO_8+	IO_20+		17	42	
IO_8-	IO_20-		18	43	
IO_9-	IO_21-		19	44	
IO_9+	IO_21+		20	45	
IO_10+	IO_22+		21	46	
IO_10-	IO_22-		22	47	
IO_11-	IO_23-		23	48	
IO_11+	IO_23+		24	49	
GND	GND		25	50	

NOTE 1: The layout of the pin numbers in this table corresponds to the physical placement of pins on the IP connector. Thus this table may be used to easily locate the physical pin corresponding to a desired signal. Pin 1 is marked with a square pad on the IP Module. Unused pins should not be connected.

FIGURE 35

IP-PARALLEL-CDU INTERFACE



Applications Guide

Interfacing

Some general interfacing guidelines are presented below. Do not hesitate to contact the factory if you need more assistance.

Watch the system grounds. All electrically connected equipment should have a fail-safe common ground that is large enough to handle all current loads without affecting noise immunity. Power supplies and power-consuming loads should all have their own ground wires back to a common point.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. Other than series resistors for the "TTL" interface the IP-Parallel does not contain special input protection.

We provide the components. You provide the system. Only careful planning and practice can achieve safety and reliability. Integrated circuits can be damaged by static discharge. Proper anti-static handling procedures must be followed.

Terminal Block. We offer a high quality 50-screw terminal block that directly connects to the flat cable. The terminal block mounts on standard DIN rails. [<http://www.dyneng.com/HDRterm50.html>]

Many flat cable interface products are available from third party vendors to assist you in your system integration and debugging. These include connectors, cables, test points, 'Y's, 50 pin in-line switches, breakout boxes, etc.

IndustryPacks® are mezzanine cards which require an adapter to work in any system. IP Modules are commonly used and frequently systems have "extra" slots where the modules can be located. Dynamic Engineering has carriers for the PCI and cPCI buses. IndustryPacks are portable and can be used on third party carriers when the hardware is compliant with the IP specification.

http://www.dyneng.com/pci_3_ip.html

<http://www.dyneng.com/pci5ip.html>

<http://www.dyneng.com/cpci2ip.html>

Different platforms have different operating system requirements. If you need a driver please contact Dynamic Engineering. Dynamic Engineering has driver expertise for Windows NT, 2000, and XP. Dynamic Engineering also writes



drivers for Linux and has plans for VxWorks and Labview. We can support your effort with driver and application software or help for your software designers. Dynamic Engineering hardware designs have features to help the integrator to write and test their software quickly and efficiently – we can help you.

Loop-back Connections

The ATP software we use to test the IP-Parallel-CDU includes a loop-back test. The Engineering Kit for the IP-Parallel-IO includes the source code for the ATP. The loop-back test is facilitated with an IP-Debug-IO card with added wire-wrapped interconnections.

485 loop-back connections (The underlined pins are for the CDU interface)

<u>+side</u>			<u>-side</u>		
49	24	I047 – I023	48	23	I046 – I022
46	21	I044 – I020	47	22	I045 – I021
45	20	I043 – I019	44	19	I042 – I018
<u>42</u>	<u>17</u>	<u>I040 – I016</u>	<u>43</u>	<u>18</u>	<u>I041 – I017</u>
41	16	I039 – I015	40	15	I038 – I014
<u>38</u>	<u>13</u>	<u>I036 – I012</u>	<u>39</u>	<u>14</u>	<u>I037 – I013</u>
37	12	I035 – I011	36	11	I034 – I010
34	9	I032 – I08	35	10	I033 – I09
33	8	I031 – I07	32	7	I030 – I06
30	5	I028 – I04	31	6	I029 – I05
29	4	I027 – I03	28	3	I026 – I02
26	1	I024 – I00	27	2	I025 – I01

Construction and Reliability

IP Modules were conceived and engineered for rugged industrial environments. The IP-Parallel-CDU is constructed out of 0.062 inch thick FR4 material.

Through hole and surface mounting of components are used. IC sockets use high quality plated screw machine pins. High insertion and removal forces are required, which assists in the retention of components. If the application requires unusually high reliability or is in an environment subject to high vibration, the user may solder the corner pins of each socketed IC into the socket, using a grounded soldering iron.



The IP Module connectors are keyed and shrouded with Gold plated pins on both plugs and receptacles. They are rated at 1 Amp per pin, 200 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

The IP is secured against the carrier with four metric M2 stainless steel screws. The heads of the screws are countersunk into the IP. The four screws provide significant protection against shock, vibration, and incomplete insertion. For most applications, they are not required. Dynamic Engineering IndustryPack Modules are shipped with a mounting kit.. [IP-MTG-KIT is available if you misplace the mounting hardware or if another IP was not shipped with the standoffs and screws]

The IP Module provides a low temperature coefficient of 0.89 W/°C for uniform heat. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-°C, and taking into account the thickness and area of the IP. The coefficient means that if 0.89 Watts are applied uniformly on the component side, then the temperature difference between the component side and solder side is one degree Celsius.

Thermal Considerations

The IP-Parallel-IO design consists of CMOS circuits. The power dissipation due to internal circuitry is very low. It is possible to create a higher power dissipation with the externally connected logic. If more than one a Watt is required to be dissipated due to external loading then forced air cooling is recommended. With the one degree differential temperature to the solder side of the board external cooling is easily accomplished.

Warranty and Repair

Dynamic Engineering warrants this product to be free from defects in workmanship and materials under normal use and service and in its original, unmodified condition, for a period of one year from the time of purchase. If the product is found to be defective within the terms of this warranty, Dynamic Engineering's sole responsibility shall be to repair, or at Dynamic Engineering's sole option to replace, the defective product. The product must be returned by the original customer, insured, and shipped prepaid to Dynamic Engineering. All replaced products become the sole property of Dynamic Engineering.

Dynamic Engineering's warranty of and liability for defective products is limited to that set forth herein. Dynamic Engineering disclaims and excludes all other product warranties and product liability, expressed or implied, including but not



limited to any implied warranties of merchandisability or fitness for a particular purpose or use, liability for negligence in manufacture or shipment of product, liability for injury to persons or property, or for any incidental or consequential damages.

Dynamic Engineering's products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of Dynamic Engineering.

Service Policy

Before returning a product for repair, verify as well as possible that the suspected unit is at fault. Then call the Customer Service Department for a RETURN MATERIAL AUTHORIZATION (RMA) number. Carefully package the unit, in the original shipping carton if this is available, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include a return address and the telephone number of a technical contact. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. Dynamic Engineering will not be responsible for damages due to improper packaging of returned items. For service on Dynamic Engineering Products not purchased directly from Dynamic Engineering contact your reseller. Products returned to Dynamic Engineering for repair by other than the original customer will be treated as out-of-warranty.

Out of Warranty Repairs

Out of warranty repairs will be billed on a material and labor basis. The current minimum repair charge is \$100. Customer approval will be obtained before repairing any item if the repair charges will exceed one half of the quantity one list price for that unit. Return transportation and insurance will be billed as part of the repair and is in addition to the minimum charge.

For Service Contact:

Customer Service Department
Dynamic Engineering

435 Park Dr.
Ben Lomond, CA 95005
831-336-8891
831-336-3840 fax
e-mail support@dyneng.com



Dynamic
Engineering

Specifications

Logic Interface:	IP Module Logic Interface
Features:	24 RS485 Transceivers two counter timers with waveform generator
Software Interface:	Control Registers, ID PROM, Vector Register, Status Ports
Initialization:	Hardware Reset forces all registers to 0.
Access Modes:	Word/Byte I/O Space (see memory map) Word in ID Space Vectored interrupt
Access Time:	back-to-back cycles in 500ns (8Mhz.) or 125 ns (32 Mhz.)
Wait States:	1 to all spaces
Interrupt:	Multiple interrupt filtering options available on each IO line. Enabled, Active hi or low, edge or level. Two counter/timer programmable interrupts. Receive CDU data interrupt
DMA:	No Logic Interface DMA Support implemented at this time.
Onboard Options:	All Options are Software Programmable
Interface Options:	50 pin flat cable 50 screw terminal block interface [HDRterm50] User cable
Dimensions:	Standard Single IP Module. 1.8 x 3.9 x 0.344 (max.) inches
Construction:	FR4 Multi-Layer Printed Circuit, Through Hole and Surface Mount Components. Programmable parts are socketed.
Temperature Coefficient:	0.89 W/°C for uniform heat across IP
Power:	Typical 52 mA @ 5V unloaded. Additional current will be required depending on the loads applied



Order Information

Tools for IP-Parallel-CDU

IP-Debug-Bus - IP Bus interface extender with testpoints, isolated power & quickswitch technology to allow hot swapping or power cycling without powering down the host.
<http://www.dyneng.com/ipdbgbus.html>

IP-Debug-IO II - IndustryPack IO connector breakout with testpoints, ribbon cable headers, and locations for user circuits. <http://www.dyneng.com/ipdbgio.html>

HDRterm50 - Ribbon cable compatible 50 pin header to 50 screw terminal header. Comes with DIN rail mounting capability. <http://www.dyneng.com/HDRterm50.html>

HDRribn50 - Ribbon cable in several standard lengths plus custom, with strain relief and cable pull attached.
<http://www.dyneng.com/HDRribn50.html>

PCI3IP - 1/2 length PCI card with 3 IP slots.
http://www.dyneng.com/pci_3_ip.html

PCI5IP - PCI card with 5 IP slots.
<http://www.dyneng.com/pci5ip.html>

cPCI2IP - cPCI card with 2 IP slots.
<http://www.dyneng.com/cpci2ip.html>

IP-MTG-KIT - 4 metric stainless screw and stand-off pairs to retain IP-Parallel-IO against the carrier board. Flat head screws match IP Specification mounting requirements.
<http://www.dyneng.com/IPHardware.html>

All information provided is Copyright Dynamic Engineering

