DYNAMIC ENGINEERING

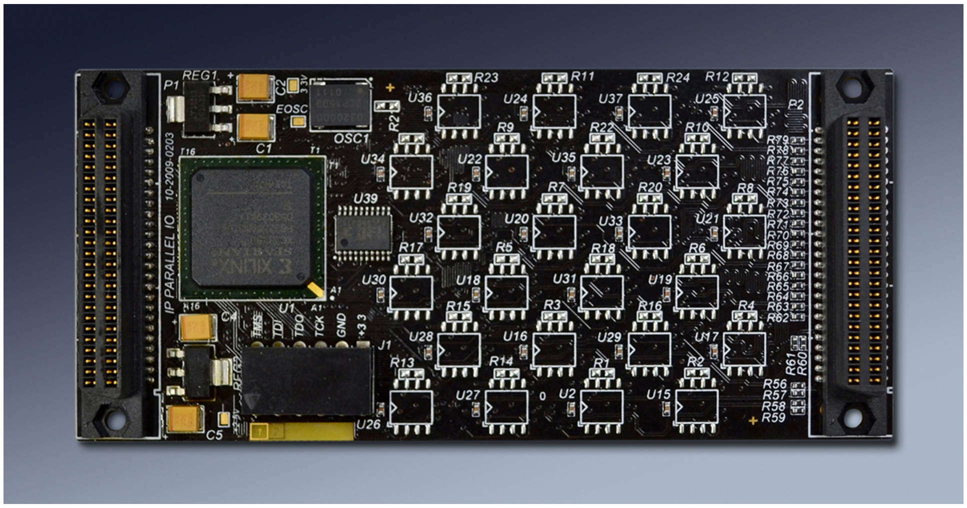
150 DuBois St. Suite B/C Santa Cruz, CA 95060
831-457-8891
sales@dyneng.com
https://www.dyneng.com
Est. 1988

User Manual

IP-Parallel-TTL-DPR

Dual Port RAM interface Plus GPIO with COS

IndustryPack® Module



Manual Revision: 1P0

FLASH revision: 0101 Fab: 10-2001-0104

IP-Parallel-TTL-DPR
Parallel Dual Port RAM interface plus GPIO
IndustryPack® Module

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.

Revised 10/11/22

Table of Contents

List of Figures

Product Description

IP-Parallel-TTL-DPR provides 48 IO with 40 allocated to the DPR interface and 8 allocated to GPIO. Each GPIO is independent with programming options for direction, polarity, level or edge triggered with separate rising and falling enables. The local 50 MHz oscillator is divided to provide the reference for the COS function. Filtered and direct IO access are also provided. Interrupts are programmable on a per line basis with 3 enables per line to cover the level, rising, and falling options.

Reference software for Win10 provide references for all of the modes of operation including setting up clocking, interrupts, using the parallel ports etc.

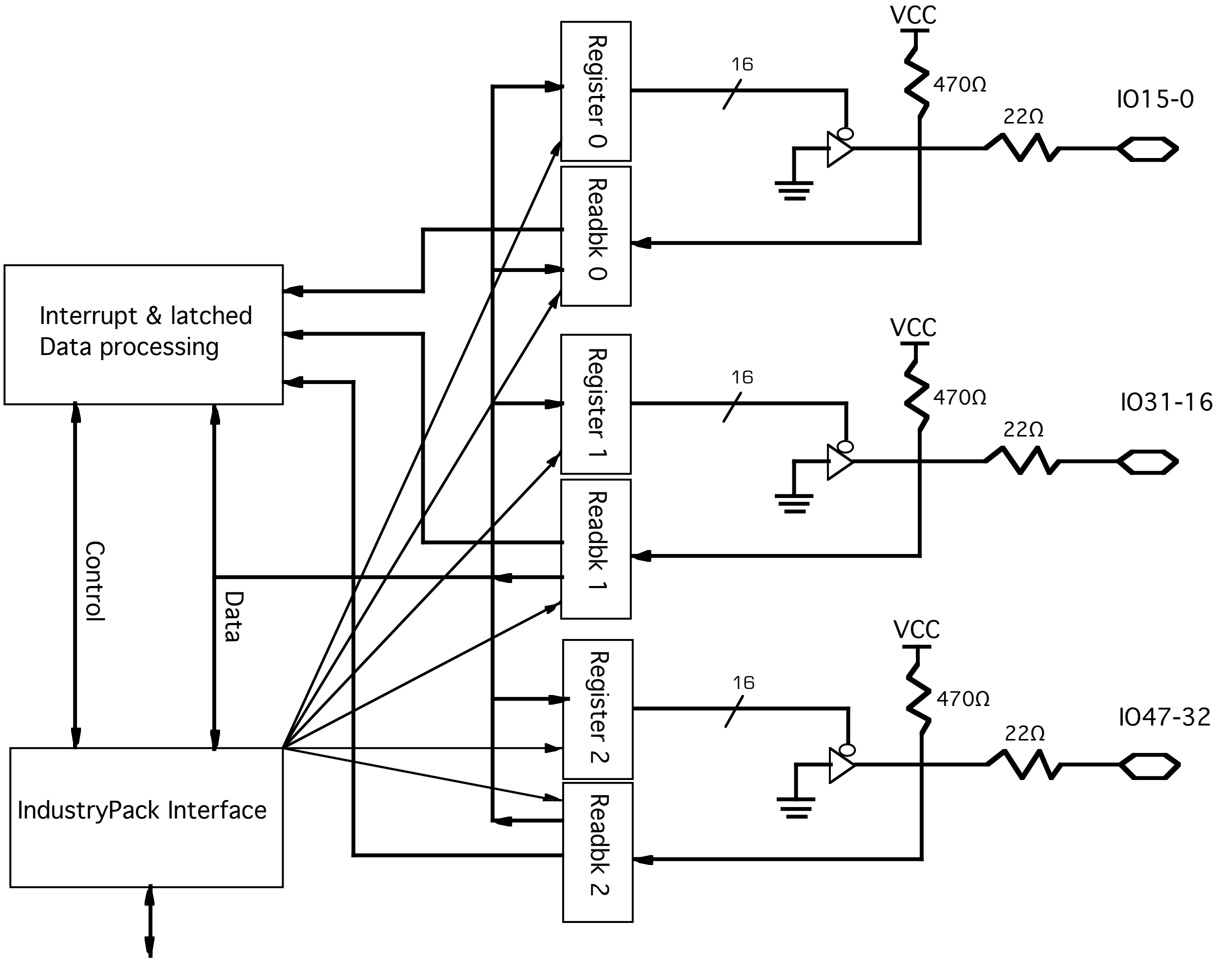


FIGURE 1 IP-PARALLEL-TTL-DPR BLOCK DIAGRAM

The Dual Port RAM interface uses the lower IO to provide Address, Data, and control for the interface. Commands are written to IP-Parallel-TTL-DPR and stored into the Command FIFO.   The commands include the Address, Type [Read or Write], and Data to transfer or the number of words to transfer for write or read cycles respectively.   For read cycles the data received from the external device is stored locally into a second FIFO for this purpose.

IP-Parallel-TTL-DPR is part of the IndustryPack® "IP" Module family of I/O components by Dynamic Engineering. IP-Parallel-TTL-DPR provides an IP Module type II compliant mechanical package with a Spartan 6 FPGA, FLASH, PLL (4 reference clocks to FPGA), 48 single ended IO each with separate direction controls. The FLASH is easily reprogrammable with the Xilinx Impact SW and USB adapter – a header is included for this purpose.

DPR is designed with a single level hierarchy – all logic at the Base level. The IP Bus interface and decoding, master interrupt control, clock generation, and Transmit and Receive logic.

The internal reference clock is programmable and derived from the 50 MHz oscillator.

IP-Parallel-TTL-DPR conforms to the VITA standard. This guarantees compatibility with multiple IP Carrier boards. Since the IP maintains plug and software compatibility while mounted on different form factors, system prototyping may be done on one IP Carrier board, with final system implementation done on a different one.

The 8 GPIO can be programmed to be transmitters or receivers with the direction selection. Data from the Tx register is driven onto the lines where the direction is set to Tx.

The received GPIO data can be read as Direct Data, Filtered Data, or COS status. Direct Data is the raw data after synchronization.   Filtered Data has Polarity and EdgeLevel applied.   Bits declared as edge are filtered out. Bits declared as inverted are reported after inversion. Active low signals on the IO can be read as active high in this manner.

The COS status is the captured change of state based on the settings in the enabled registers and the activity on the line. Separate Rising and Falling enables and status are supplied.

The sample rate for the COS is derived from the local oscillator. The HalfDiv register is used to select the divisor to set the sample rate.

Interrupts are supported by GPIO. The interrupt occurs when the programmed event occurs. Status is provided in the Interrupt Status register – summary of Direct, COS Rising and COS Falling.   The interrupts are individually maskable, and the interrupt vector is user programmable by a read/write register. The interrupt occurs on IntReq0. Status is available making it possible to operate in a polled mode. The COS interrupts are held until cleared by writing back to the respective status registers.

Product Details

IP-Parallel-TTL-DPR is designed for the purpose of acting as a generic Parallel Port or monitoring data with Change-Of-State capability plus providing the main function of Dual Port RAM interface.

IP-Parallel-TTL-DPR features a Xilinx FPGA. The FPGA contains all of the registers and protocol controlling elements of the DPR design.

The DPR interface is a specialized parallel port. Address 19-0, Data 15-0, WRITE, READ, ACK, BUSY signals are used. For a Read or Write cycle Address is asserted first.

For a write cycle the data is asserted second. The WRITE signal is asserted third. The Target device will respond with BUSY and or ACK. BUSY is a status indicating the transfer is delayed due to Address conflict. ACK assertion is used to signal the transfer is complete. When ACK is detected asserted the WRITE signal is disabled and the Data tri-stated.

Reads are slightly different with the data remaining tri-stated until the target drives the value onto the bus. Data is captured after ACK is detected. READ is used in place or WRITE for the transfer.

100 MHz is used internally to run the state-machine and control the timing of the interface. Ribbon cable was used to simulate the system wiring. Due to some cross talk and reflection from the medium delays were inserted into the transfer to reduce the number of bits switching at once [helped significantly with cross talk] and increasing the time from Address and Data assertion to WRITE to make sure the data is stable on the line. With READ cycles a delay is inserted after ACK is asserted before data is captured to help make the system more robust.

The reference is also used to track the total time of the transfer to make sure ACK follows access and the State-machine is not stuck waiting for that event. The cycle is declared a timeout error slightly over 1 uS after WRITE or READ is asserted. Status is available to see if a timeout has occurred.

OverFlow of the Data FIFO as well as empty, and full status for the command and data FIFOs are supplied in the status register. The counts of the commands or data stored are also available in separate registers.

The read cycle has a special feature – read multiple. To read from a single address set the data field to 0x00. To read more than 1 data set the field to N-1. For example, set

Address to 0x00 to start at the beginning of memory and the Data to 0xFE to read 255 data. The interrupt can be enabled to signal the end of the transfer or the Data FIFO count can be used. The Data FIFO is 1K deep and the command FIFO can store 255 commands. As long as the data is read out, multiple read commands can be stacked and then read back in parallel with the DPR accesses. With the Target simulator [second IP-Parallel-TTL with FLASH to implement the DPR interface] the cycle time is about 330 nS per read and runs continuously until completed.

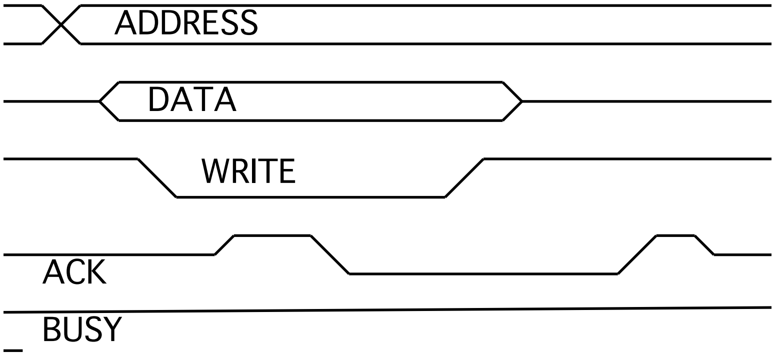


FIGURE 2 IP-PARALLEL-TTL-DPR WRITE TIMING

Address asserted to Data asserted = 10 nS to provide roughly ½ the edges changing at the same time. Data to WRITE asserted = 50 nS. Write asserted to ACK asserted is a max of 190 nS when not BUSY [as shown]. WRITE is deasserted following detection of ACK asserted leading to ACK deassertion. Data is tri-stated. Address is also tri-stated where local terminations will tend to cause 0xFFFFF.

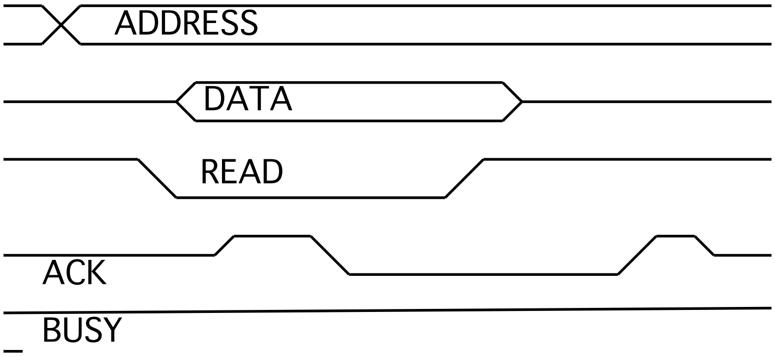


FIGURE 3 IP-PARALLEL-TTL-DPR READ TIMING

Address asserted to READ asserted = 50 nS. Data to READ asserted = 60 nS. READ asserted to ACK asserted is a max of 190 nS when not BUSY [as shown]. READ is deasserted following detection of ACK asserted leading to ACK deassertion.   Data is tri-stated by the target. Address is either tri-stated or driven to the next address in the case of read multiple operation.

ACK is shown coming out of tri-state, driven high and then driven low. The terminations are pull-ups meaning the tri-stated level is really the same as driven high. The local state-machine checks for ACK high before checking for ACK low to allow for situations with alternate termination schemes.   ACK is driven high for the off state before being tri-stated to ensure the end of the cycle can be determined.

The Spartan VI requires three voltages, derived via Buck regulators from the 5V IP standard voltage [3.3V, 2.5V, and 1.2V].   The 3.3V rail is also used for the TTL devices allowing mix and match assembly. [RS-485 and LVDS with this PCB for other implementations]

The FLASH is used to store the module firmware – in this case the DPR and GPIO functions.

The Base level of the design provides the IndustryPack interface, general decoding, card level status, IDPROM and Vector register plus board level features, interrupt masking, master channel enable.

DPR and GPIO are implemented as a flat design with the Base and IO features at the same level.   The decoded data bus is routed to each internal register to create control logic, status, data storage to support the serial IO and parallel ports.

Please refer the address/bit maps for details of the various registers and operation.

IP-Parallel-TTL-DPR conforms to the IndustryPack® standard. This guarantees compatibility with compliant carrier boards. Because IP-Parallel-TTL-DPR may be mounted on different form factors, while maintaining plug and software compatibility, system prototyping may be done on one Carrier board, with final system implementation on a different one. For example, the PCIe3IP – PCIe carrier for IP Modules can be used for development in a conventional PC. Later the hardware and software can be ported to the target. http://www.dyneng.com/PCIe3IP.html

With the Dynamic Engineering Windows driver collection for IP and carrier modules a Parent – Child architecture is employed. The IP portion of the driver is directly portable between the various Dynamic Engineering IP carriers [PCIe3IP, PCI3IP, PCIe5IP, PCI5IP, PC104pIP, PC104p4IP, cPCI2IP, cPCI4IP etc]. The parent portion of the driver contains the carrier specific design information.   This means that software developed for IP-Parallel-TTL-DPR on one platform can be directly ported to another. PCI to cPCI for example.

Designers can make use of the Dynamic Engineering carrier driver for non-Dynamic Engineering IP modules using the Generic IP capability built into the parent portion of the driver. IP modules that the carrier driver does not recognize are installed as “generic” and accessed with a address, data interface model. Software developed for the Generic mode can also be ported between modules.

IP-Parallel-TTL-DPR is tested with a combination of internal and external tests. The registers can be checked with R/W tests, the functions can be tested with the external loop-back fixture.

Interrupts are supported by IP-Parallel-TTL-DPR. A force interrupt for software development and test is provided, plus an interrupt from multiple received data options. Status is provided separate from the interrupt to allow use in a polled environment.

Change of State or “COS” is used to detect the transition of a signal from high to low or vice-versa. Each IO has a separate detector for the rising edge and falling edge. If both are enabled COS is implemented. Otherwise technically, it is a rising or falling detector. The transitions are based on the external signal. With an active low signal and falling edge enabled the leading edge of an active transition would be selected independent of the polarity setting. (polarity is only applied to filtered data).

When the edge is detected, the corresponding status bit is set and remains set until explicitly cleared by writing to the status register with the corresponding bit set. Reset also clears stored status.

If the interrupt for active bit is enabled, the ANDed enable and status are ORed together to create the rising or falling interrupt. If not enabled the status is available for polling. Please note the Master Interrupt Enable is also required for any of the programmed interrupts to create an interrupt to the host.

Level based interrupts are masked with the same interrupt enable bit and masked with the EdgeLevel selection. The Filtered data is the same data ahead of the interrupt enable being applied. Level data can also be polled in this manner. For level data the state of the IO line would need to change to remove the interrupt. In many cases the master or IO line level interrupt enable will need to be disabled until the line is in the ready state rather than the triggered state.

Address Maps

Address Map Base

```
IpParTtlDpr_Base            0x00  //0 Base control register
IpParTtlDpr_Vector          0x02  //1 Interrupt vector register
IpParTtlDpr_IntStatus       0x04  //2 Interrupt status read
IpParTtlDpr_Rev             0x06  //3 Revision, Major&Minor, read only
IpParTtlDpr_Info            0x08  //4 Spare port used by OS
IpParTtlDpr_CmdData         0x0A  //5 DPR Data 15-0
IpParTtlDpr_CmdAddr         0x0C  //6 DPR Address 15-0
IpParTtlDpr_CmdCntl         0x0E  //7 CmdRd, CmdWr, DPR Add 19-16
IpParTtlDpr_RdDataFIFO      0x10  //8 Read from Data FIFO
IpParTtlDpr_CmdFifoCnt      0x12  //9 Command FIFO Count
IpParTtlDpr_DataFifoCnt     0x14  //10 Data FIFO Count
IpParTtlDpr_spare           0x16  //11 unused
IpParTtlDpr_CosFallingEn    0x18  //12 COS Falling Enable 7-0
IpParTtlDpr_CosRisingEn     0x1A  //13 COS Rising Enable 7-0
IpParTtlDpr_CosEdgeLevel    0x1C  //14 Edge or Level 7-0
IpParTtlDpr_CosIntEn        0x1E  //15 Interrupt Enable 7-0
IpParTtlDpr_CosPolarity     0x20  //16 Polarity 7-0
IpParTtlDpr_GpioDir         0x22  //17 GPIO Direction 7-0
IpParTtlDpr_GpioOut         0x24  //18 GPIO Tx Data 23-16
IpParTtlDpr_GpioDirect      0x26  //19 Direct Read of IO 7-0
IpParTtlDpr_HalfDiv         0x28  //20 Set Divisor for COS Rate
IpParTtlDpr_CosRisingSt     0x2A  //21 COS Rising Status 7-0
IpParTtlDpr_CosFallingSt    0x2C  //22 COS Falling Status 7-0
IpParTtlDpr_CosFiltered     0x2E  //23 Filtered Read of IO 7-0
IpParTtlDpr_Test0           0x30  //24 15-0 – use for local test of DPR
IpParTtlDpr_Test1           0x32  //25 31-16 – IO bits.  Read Always
IpParTtlDpr_Test2           0x34  //26 39-32 -    Write when enabled
```

FIGURE 4 IP-PARALLEL-TTL-DPR ADDRESS MAP

Numbers following the // are the HW decode numbers based on the words – word 0, word 1 etc. There are 64 available in the IP IO space.

Programming

Programming IP-Parallel-TTL-DPR requires only the ability to read and write data from the host. The base address refers to the first user address for the slot in which the IP is installed.

The registers are organized as shown in the memory map and can be directly accessed with the offset to the IP Module position on the carrier. The base level has card level control and information. Channels when used, repeat with the same relative offsets to the start of each address range shown.

Depending on the software environment it may be necessary to set-up the system software with I IP-Parallel-TTL-DPR "registration" data. Other OS may be more "plug and play". The Dynamic Engineering Driver operates in a "plug and play" mode using parent ⇔ child architecture.

Interrupts are available to alert the local host when an event has happened.

The Dynamic Engineering drivers for Windows WFM (10, 11) and Linux, manage the interaction with the carrier level driver. Please refer to the driver manuals for more information. The manuals are downloadable from the webpage. Additional OS support can also be found on the website. Please contact Dynamic Engineering if you need additional SW options – VxWorks etc.

Register Definitions

IpParTtlDpr_Base

0x0000 // 0 base control register offset

DATA BIT	DESCRIPTION
15	MasterIntEn
14	TestIOEn
13-12	Spare
11	ForceInt
10-9	Spare
8	DPR IntEn
7-2	Spare
1	DPR Reset
0	DPR En

FIGURE 5 IP-PARALLEL-TTL-DPR BASE CONTROL REGISTER BIT MAP

DPR En: Set to enable DPR function. When cleared interface held in IDLE. Command and Data FIFOs can still be used.

DPR Reset when ‘1’ causes DPR state-machine to reset and clears associated FIFOs. ‘0’ for standard operation.

DPR Interrupt when ‘1’ used DPR state-machine status to cause interrupt request to host. Requires MasterIntEn to be enabled. Interrupt asserted at end of IO cycle. Most useful for read multiple commands.

ForceInt: when set causes an interrupt to be generated to the system. Requires MasterIntEn to be enabled. Useful for debugging and software test.

TestIoEn: when set selects the TEST0,1,2 registers to control the IO associated with the DPR function. Cleared for normal operation.

MasterIntEn: when set allows programmed interrupt sources to assert Int0. Required for all interrupt types. When ‘0’ no interrupts will be requested, all status can be used for polling.

IpParTtlDpr_VECTOR
0x0002 // 2 IP vector port
Vector Port

DATA BIT	DESCRIPTION
15-8	Undefined
7-0	vector

FIGURE 6 IP-BISERIAL-VI-GPIO VECTOR BIT MAP

If the system uses a vectored interrupt approach the vector port should be initialized to the value assigned to this device. IP-Parallel-TTL-DPR can be used as vectored or auto-vectored. When auto-vectored this port is unused. The Status port can be used to determine the source of any pending interrupts from P-Parallel-TTL-DPR.

Default is 0xFF for data.

IpParTtlDpr_IntStatus

0x0004 // 2 Interrupt Status register

DATA BIT	DESCRIPTION
15	DprIdleState
14	Locked100
13	IntReqM
12	IntReqUM
11	CosRisingIntReq
10	CosFallingIntReq
9	LvlIntReq
8	ForceInt
7	CmdFifoFull
6	CmdFifoMt
5	DataFifoFull
4	DataFifoMt
3	WriteErrorLat
2	ReadErrorLat
1	FifoOvrFlLat
0	DprIntLat

FIGURE 7 IP-PARALLEL-TTL-DPR INTERRUPT STATUS BIT MAP

DprIdleState: When set ‘1’ the DPR state-machine is in the idle state. When ‘0’ the statemachine is processing commands.

Locked100: When ‘1’ the local 100 MHz DCM has locked onto the 50 MHz [osc] reference and is generating the 100 MHz required for operation. The circuit has a watchdog reset control. If not locked within ~ 1mS the DCM is reset to retry locking on. If not set likely a problem with the oscillator.

IntReqUM: when set indicates an enabled interrupt request is pending. Signal level before the Master Interrupt Enable. “Unmasked”

IntReqM: When **IntReqUM** is set and the Master Interrupt Enable is set this bit will be set.

CosRisingIntReq when Set indicates at least one Rising Status bit is set along with the corresponding interrupt enable. When cleared no interrupts are pending from this source.

CosFallingIntReq when Set indicates at least one Falling Status bit is set along with the corresponding interrupt enable. When cleared no interrupts are pending from this source.

LvlIntReq when Set indicates at least one bit declared as level, plus polarity, plus interrupt enable are true. When cleared no interrupts are pending from this source.

Note: COS interrupts are cleared by writing back to the Rising and or Falling status registers. Level interrupts are not cleared per se. Level interrupts are masked off until the IO line returns to the ready state, SW can then rearm the interrupt.

ForceInt is set when ForceInt is active in the base register. Slightly redundant having in both the base read and IntStatus read, and helpful for developing ISR routines to have the ForceInt behave as much like the IO interrupts as possible.

CmdFifoFull: is set when the Command FIFO is full. Should be '0' before loading a new command. 255x48 are the dimensions of the Command FIFO.

CmdFifoMt: is set when the Command FIFO is empty. The DPR function when enabled automatically reads from the Command FIFO when not empty.

DataFifoFull: is set when the Data Storage FIFO is at capacity [1023 x16]. See Count register for direct read of current total. If Full is set when time to write more, the OverFlow bit will be set.

DataFifoMt: when set the Data FIFO is empty.

WriteErrorLat: When set a Write error has occurred. This bit is latched. Clear by writing back with this bit set. Write error happens when the target HW fails to respond.

ReadErrorLat: When set a Read error has occurred. This bit is latched. Clear by writing back with this bit set. Write error happens when the target HW fails to respond.

FifoOvFlLat: When set a the Data Storage FIFO was full when time to write more data. This bit is latched. Clear by writing back with this bit set.

DprIntLat: Set when the DPR interface completes a command. If the DprIntEn, and MasterInEn are also set an interrupt request can be generated. The IO side of the DPR interface is reasonably fast compared to interrupt processing. Recommend to use as status or for Read Multiple commands where longer read cycles can occur. 255 x 330 => 84 uS or so to complete.

IpParTtlDpr_INFO

0x0006 // 3 Location Register

DATA BIT	DESCRIPTION
15-11	spare
10-3	Carrier Switch
2-0	Carrier Slot

FIGURE 8 IP-PARALLEL-TTL-DPR INFO BIT MAP

The location register can be updated by the carrier driver during initialization. IP-Parallel-TTL-DPR Driver can access this information later to determine the carrier and location on the carrier that this node is installed into. Once the IP-Parallel-TTL-DPR Driver is started the user software can use this as a general purpose register. The IP Driver stores a local copy in RAM to allow the user software to determine which node it is talking to when multiple nodes are present in a PCI/PCIe based system with dynamic addressing. Please note that this function is supported on all Dynamic Engineering carriers and may not be supported on other products.

In a Windows system the user software will query for the installed devices and the devices will be returned in order. The issue is that the order can change and there is no way to directly tell with software which card you are controlling at the moment. The user software can retrieve the devices present and then match them up to the physical hardware based on the carrier switch setting and slot on the carrier.

In some systems, knowing which board is controlling which machine can be important. Please see the software manual and userap reference software for examples of working with multiple cards. The userap software prints out the device number and associated slot and switch settings. Your software can use the information without printing out for proper access control.

This is only important if you have multiple cards visible to the same CPU.

IpParTtlDpr_REV

0x0008 //4 Revision Register

DATA BIT	DESCRIPTION
15-8	RevisionMajor
7-0	RevisionMinor

FIGURE 9 IP-PARALLEL-TTL-DPR REVISION BIT MAP

RevisionMajor:
This field is reported via the IDPROM as well [revision]. It is rolled when major changes occur.

RevisionMinor:
This field is only read from this location. It is rolled when minor changes occur – usually during development to allow SW tracking of HW revisions without using the Major Revision field.

Revisions (Major P Minor)
1p1 original – New Design 9/22

IpParTtlDpr_CmdData

x0A // 5

DATA BIT	DESCRIPTION
15-0	CMD Data

FIGURE 10 IP-PARALLEL-TTL-DPR CMD DATA

Write the data pattern to load into DPR or the quantity of data to read from DPR. N-1 for Read count i.e. 0 for read 1, xFE for xFF etc.

IpParTtlDpr_CmdAddr

x0C // 6

DATA BIT	DESCRIPTION
15-0	CMD Address 15-0

FIGURE 11 IP-PARALLEL-TTL-DPR CMD ADDRESS

Write the Lower 16 bits of the address to access DPR. For a write this is the address. For a read this is the starting address and could be auto incremented with read multiple commands

IpParTtlDpr_CmdCntl

x0E // 7

DATA BIT	DESCRIPTION
7-6	Spare
5	DprReadCmd
4	DprWriteCmd
3-0	CMD Address 19-16

FIGURE 12 IP-PARALLEL-TTL-DPR CMD CNTL

Complete the address with the lower bits. Set either the read or the write and not both. The state-machine will ignore any commands with both set or neither set.

Note: 39-0 are the IO in use. The IO include the BUSY and ACK bits from the target [spares in this register]. The GPIO uses the upper bits – see the separate GPIO control register for this.

The three register's contents are transferred from the registers to the command FIFO automatically when the CmdCntl register is written to. If changing the data in the lower two write those first. Note: The CmdAddr and CmdCntl registers are on a LW boundary allowing a single write to cover both registers – Dynamic Engineering carriers support this operation – No need to adjust the address across the two registers in this mode.

If enabled the DPR state-machine will sense the not empty status and process the command. The front side runs at the IP clock rate – recommend 32 MHz for best performance. The back side runs at 100 MHz. for optimal IO side timing.

IpParTtlDpr_RdDataFifo

x10 // 8 Read Data FIFO

DATA BIT	DESCRIPTION
15-0	DPR Read Command Data

FIGURE 13 IP-PARALLEL-TTL-DPR READ DATA FIFO

Read this location to retrieve data returned from accesses to DPR. Same order as commanded. No address information stored.

IpParTtlDpr_CmdFifoCnt

x12 // 9 Command FIFO Count

DATA BIT	DESCRIPTION
15-0	DPR Command FIFO Count

FIGURE 14 IP-PARALLEL-TTL-DPR COMMAND FIFO COUNT

Read this location to determine the number of commands currently stored. xFF is max count.

IpParTtlDpr_DataFifoCnt

x14 // 10 Data FIFO Count

DATA BIT	DESCRIPTION
15-0	DPR Data FIFO Count

FIGURE 15 IP-PARALLEL-TTL-DPR DATA FIFO COUNT

Read this location to determine the number of data words currently stored. x3FF is max count.

IpParTtlDpr_CosFallingEn

x18 // 12 COS Falling Enable

DATA BIT	DESCRIPTION
7-0	COS Falling Enable

FIGURE 16 IP-PARALLEL-TTL-DPR COS FALLING ENABLE

The bits in the Falling Enable register are set to enable falling edge detections of the corresponding IO bit. When cleared the falling edges of the IO bit are ignored.

IpParTtlDpr_CosRisingEn

x1A // 13 COS Rising Enable

DATA BIT	DESCRIPTION
7-0	COS Rising Enable

FIGURE 17 IP-PARALLEL-TTL-DPR COS RISING ENABLE

The bits in the Falling Enable register are set to enable rising edge detections of the corresponding IO bit. When cleared the rising edges of the IO bit are ignored.

IpParTtlDpr_CosEdgeLevel

x1C // 14 Edge Level Select Control Register

DATA BIT	DESCRIPTION
7-0	EdgeLevel

FIGURE 18 IP-PARALLEL-TTL-DPR EDGELEVEL BIT MAP

Each Bit of the EdgeLevel Control Register corresponds to a bit in the Parallel Port. When set the bit is selected to be COS processed with Edge. When cleared the bit is selected for Level processing. The type of processing coupled with the interrupt enable selects how a particular IO is handled. The direct data is always available. The Rising and Falling status are always available. EdgeLevel is applied to filter which potential interrupters of each type are selected and the Interrupt Enable further selects which subset of those can actually generate an interrupt.

For example the direct data is filtered with EdgeLevel and only the level bits make it to the Filtered data [also treated with Polarity]. After the Filtered level the Interrupt Enable is applied to potentially allow interrupts from those sources.

IpParTtlDpr_CosIntEn

x1E // 15 Interrupt Enable Control Register

DATA BIT	DESCRIPTION
7-0	Interrupt Enable

FIGURE 19 IP-PARALLEL-TTL-DPR INTEN BIT MAP

Each bit of the IntEn registers corresponds to an input bit. When set the IO can cause an interrupt. When cleared that IO is masked out. Please note: EdgeLevel selects which type of pre-processing is applied to the Input prior to the interrupt enable. Polarity, Rising, Falling are used to select the active level or edge to process.

IpParTtlDpr_CosPolarity

x20 // 16 Polarity Control Register

DATA BIT	DESCRIPTION
7-0	Polarity

FIGURE 20 IP-PARALLEL-TTL-DPR POLARITY BIT MAP

Each Bit of the Polarity Control Register corresponds to a bit in the Parallel Port. When set the sense of the filtered data is inverted, when cleared the received data is left alone. For example, if the receive signal is active low the corresponding polarity bit is set to have it reflected as an active high signal in the filtered data. If the bit is to be used as an interrupt the enable is ANDed with the filtered data making the sense important to program. If not used for interrupt purposes it is up to the user to decide whether inversion is helpful or not. Please note: polarity does not affect the Direct Data port.

IpParTtlDpr_GpioDir

x22 // 17 Direction Control Register

DATA BIT	DESCRIPTION
7-0	Direction

FIGURE 21 IP-PARALLEL-TTL-DPR GPIO DIRECTION BIT MAP

Each Bit of the Direction Control Register corresponds to a bit in the Parallel Port. When set the IO is programmed to transmit, when cleared the IO is programmed to receive.

IpParTtlDpr_GpioOut

x24 // 18 Direct Output defininition

DATA BIT	DESCRIPTION
7-0	Tx Data Definition

FIGURE 22 IP-PARALLEL-TTL-DPR GPIO TX DATA

Data to transmit from the GPIO port is written to this register. The data is masked with the Direction register – only enabled bits are driven.

IpParTtlDpr_GpioDirect

x26 // 19 Direct IO Read

DATA BIT	DESCRIPTION
7-0	IO Definition

FIGURE 23 IP-PARALLEL-TTL-DPR GPIO DIRECT DATA

The current state of the IO lines are returned from this register. Synchronizing to the IP clock is the only filtering done with this port.

IpParTtlDpr_HalfDiv

x28 // 20

DATA BIT	DESCRIPTION
15-0	HalfDiv

FIGURE 24 IP-PARALLEL-TTL-DPR HALFDIV BIT MAP

HalfDiv is programmed with the divisor used for the first level of division within the user specified counter. The reference for the counter is the 100 MHz in this design.

The output of the user specified counter is further divided by 2 to create a square wave no matter the programmed divisor set.

Example: Desire 1 MHz. reference for Transmission. Last stage is /2 => divide 100 MHz to get 2 MHz. = 50. The first stage divider is preloaded with x1 and rolls over to x1 when the programmed count is reached.

IpParTtlDpr_CosRisingSt

x2A // 21 COS Rising Status

DATA BIT	DESCRIPTION
7-0	COS Rising Status

FIGURE 25 IP-PARALLEL-TTL-DPR RISING STATUS

The bits in the Rising Status register are set when the selected IO transitions from low to high. The programmed sample rate can affect how long it takes for the transition to be reflected in the status. See HalfDiv for more information. Please note: the corresponding CosRisingEn bit will need to be set in order to enable the capture of the corresponding rising edge. Status is cleared by writing to this port with the status bit to clear set.

IpParTtlDpr_CosFallingSt

x2C // 22 COS Falling Status

DATA BIT	DESCRIPTION
7-0	COS Falling Status

FIGURE 26 IP-PARALLEL-TTL-DPR FALLING STATUS

The bits in the Falling Status register are set when the selected IO transitions from high to low. The programmed sample rate can affect how long it takes for the transition to be reflected in the status. See HalfDiv for more information. Please note: the corresponding CosFallingEn bit will need to be set in order to enable the capture of the corresponding falling edge. Status is cleared by writing to this port with the status bit to clear set.

IpParTtlDpr_CosFiltered

x2E // 23 Filtered Data

DATA BIT	DESCRIPTION
7-0	Filtered Data

FIGURE 27 IP-PARALLEL-TTL-DPR FILTERED DATA

Read only port. The value of the port is the result of applying the EdgeLevel and Polarity to the IO. Please note: Direction is not a filter – transmitted bits can be filtered inputs if desired. The Polarity is applied on a bit-by-bit basis and then masked by the EdgeLevel definition to present the Level Selected data with the desired polarity. Please note: the output of the Filtered data is ANDed with the Interrupt Enable bits and then ORed together to create the level type interrupt request.

IpParTtlDpr_Test0

x30 // 24 Test port IO 15-0

DATA BIT	DESCRIPTION
15-0	IO15-0

FIGURE 28 IP-PARALLEL-TTL-DPR TEST DATA 0

R/W port with the read returning the state of the IO lines for the associated bit. When TestIoEn is set, the contents of this register are driven onto the IO lines to support BIT.

IpParTtlDpr_Test1

x32 // 25 Test port IO 31-16

DATA BIT	DESCRIPTION
15-0	IO31-16

FIGURE 29 IP-PARALLEL-TTL-DPR TEST DATA 1

R/W port with the read returning the state of the IO lines for the associated bit. When TestIoEn is set, the contents of this register are driven onto the IO lines to support BIT.

IpParTtlDpr_Test2

x34 // 26 Test port IO 39-32

DATA BIT	DESCRIPTION
7-0	IO39-32

FIGURE 30 IP-PARALLEL-TTL-DPR TEST DATA 2

R/W port with the read returning the state of the IO lines for the associated bit. When TestIoEn is set, the contents of this register are driven onto the IO lines to support BIT.

Interrupts

IP-Parallel-TTL-DPR interrupts are treated as auto-vectored on many carriers.   When the software enters into an exception handler to deal with an IP-Parallel-TTL-DPR interrupt the software must read the status register(s) to determine the cause(s) of the interrupt, clear the interrupt request(s) and process accordingly. Power on initialization will provide a cleared interrupt request and interrupts disabled.

The interrupt is mapped to INT0 on the IP connector, which is mapped to a system interrupt via the host [carrier] device. The source of the interrupt is obtained by reading the Interrupt Status registers. The status remains valid until that bit in the status register is cleared.

When an interrupt occurs, the Master interrupt enable should be cleared and the status register read to determine the cause of the interrupt. Next perform any processing needed to remove the interrupting condition, clear the status and enable the interrupt again.

The individual enables operate after the interrupt holding latches, which store the interrupt conditions for the CPU. This allows for operating in polled mode simply by monitoring the Interrupt Status register.

The base level has a master interrupt enable that affects all interrupt sources.

An interrupt Vector can be read for systems requiring vectored interrupt processing.

The Windows and Linux SW packages provide both straight Interrupt Status read and ISR status read. The ISR when activated by an interrupt request, disables the master interrupt enable and reads the Rising, Falling, and Filtered data. The Rising and Falling status is then cleared within the handler. Reading the ISR structure returns all of the interrupt status saving multiple application level calls. The standard Interrupt Status read only returns the status without any other operations.

Loop-back

The Engineering kit has reference software, which includes an external loop-back test. The test requires external connections. The Acceptance Test Procedure [ATP] for this device uses the internal tests without a fixture attached. Once those tests have completed the Target interaction tests are run. A second IP-Parallel-TTL [50 MHz Osc installed] board with a DPR Target simulator design is used for this purpose. 32Kx16 of RAM are used to provide live testing of the interface. We use IP-Debug-IO and IP-Debug-Bus to allow connection to the carrier and access to the testpoints on IO signals. For development a logic analyzer was attached to the header connections on IP-Debug-IO. All timing was verified in this manner.

The Target board is a standalone interface with no SW required. It does have an IP address assigned so you can identify in the system. It resets hot and waits for commands from IP-Parallel-TTL-DPR.

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 be present, it may check for it directly. The revision is also readable from the base revision register where both the major [reported in the PROM] and minor fields are available.

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-TTL-DPR is shown in the figure below. For more information on IP ID PROM's refer to the IP Module Logic Interface Specification.

Address	Data	
01	ASCII "I"	($49)
03	ASCII "P"	($50)
05	ASCII "A"	($41)
07	ASCII "H"	($48)
09	Manufacturer ID	($1E) Dynamic Engineering
0B	Model Number	($03) IP-Parallel-TTL
0D	Revision	($01) RevisionMajor
0F	reserved	($05) Customer Number
11	Driver ID, low byte	($0C) Design Number-DPR
13	Driver ID, high byte	($00)
15	No of extra bytes used	($0C)
17	CRC	($D6)

FIGURE 31 IP-PARALLEL-TTL-DPR ID PROM

IP-Parallel-TTL-DPR Logic Interface Pin Assignment

The figure below gives the pin assignments for the IP Module Logic Interface on IP-Parallel-TTL-DPR. Pins marked n/c below are defined by the specification, but not used on the IP-Parallel-TTL-DPR. 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
D0	IDSEL*	4	29
D1	n/c	5	30
D2	MEMSEL*	6	31
D3	n/c	7	32
D4	INTSEL*	8	33
D5	n/c	9	34
D6	IOSEL*	10	35
D7	n/c	11	36
D8	A1	12	37
D9	n/c	13	38
D10	A2	14	39
D11	n/c	15	40
D12	A3	16	41
D13	INTREG0*	17	42
D14	A4	18	43
D15	n/c	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. This table may be used to easily locate the physical pin corresponding to a desired signal. Pin 1 is marked on the IP Module.

FIGURE 32 IP-PARALLEL-TTL-DPR LOGIC INTERFACE

IP-Parallel-TTL-DPR IO Pin Assignment

The figure below gives the pin assignments for the IP Module IO Interface on IP-Parallel-TTL-DPR. Also see the User Manual for your carrier board for more information. IO#(signal name). Schematic shows IO#. Pairs are defined to be compatible with IP Standard Differential wiring – used on PCIe3IP, PCIe5IP etc.

IO0	IO24	1	26
IO1	IO25	2	27
IO2	IO26	3	28
IO3	IO27	4	29
IO4	IO28	5	30
IO5	IO29	6	31
IO6	IO30	7	32
IO7	IO31	8	33
IO8	IO32	9	34
IO9	IO33	10	35
IO10	IO34	11	36
IO11	IO35	12	37
IO12	IO36	13	38
IO13	IO37	14	39
IO14	IO38	15	40
IO15	IO39	16	41
IO16	IO40	17	42
IO17	IO41	18	43
IO18	IO42	19	44
IO19	IO43	20	45
IO20	IO44	21	46
IO21	IO45	22	47
IO22	IO46	23	48
IO23	IO47	24	49
IO_GND	IO_GND	25	50

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

FIGURE 33 IP-PARALLEL-TTL-DPR CONNECTOR PINOUT

Notes:
IO_GND = DC ground for revision 04 PCB.
DPR Address(19-0) = IO 19-0
DPR Data(15-0) = IO 35-20
DPR WRITE = IO(36)
DPR READ = IO(37)
DPR ACK = IO(38)
DPR BUSY = IO(39)
GPIO(7-0) = IO(47-40)

Applications Guide

Interfacing

The pin-out tables are displayed with the pins in the same relative order as the actual connectors. Ideally the interconnecting cable is shielded and if very long it is recommended to separate the Address, Data, Control signals into separate groups to reduce switch induced cross talk.

We provide the components. You provide the system. Safety and reliability can be achieved only by careful planning and practice. Inputs can be damaged by static discharge, or by applying voltage outside of the TTL devices rated voltages. TTL signals have small series resistors located near the connector to provide some protection.

If induced noise is causing errors, please check the cabling and make sure the shields are properly tied to ground on one side. It may be necessary to go to higher grade cable.

Construction and Reliability

IP Modules are conceived and engineered for rugged industrial environments. IP-Parallel-TTL-DPR is constructed out of 0.062 inch thick high temp ROHS compliant FR4 material.

Options are available for ROHS and standard processing

Through hole and surface mounting of components are used.

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.

The IP Module provides a low temperature coefficient of .89 W/^{O}C for uniform heat. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-^{O}C, and taking into account the thickness and area of the IP. The coefficient means that if .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-TTL-DPR 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 Watt is required to be dissipated due to external loading forced air cooling is recommended. With one-degree differential temperature to the solder side of the board external cooling is easily accomplished.

Warranty and Repair

Please refer to the warranty page on our website for the current warranty offered and options.

https://www.dyneng.com/warranty.html

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.   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
150 DuBois St Suite B/C
Santa Cruz, CA 95060
831-457-8891
support@dyneng.com

Specifications

Host Interface:	IP Module 8 and 32 MHz capable
IO Interface:	48 bit parallel port with 40 allocated to DPR interface and 8 as GPIO with COS.
COS sample rates generated:	100 MHz for COS reference (SW Selectable).
Software Interface:	Control Registers, Status Ports
Initialization:	Hardware Reset forces all registers [except vector] to 0.
Access Modes:	IO, Memory, ID, INT spaces (see memory map)
Wait States:	minimized based on programmed clock rate
Interrupt:	Programmable per bit/function
Onboard Options:	Most Options are Software Programmable. Shunt for IO ground reference: open, DC, AC
Interface Options:	48 IO plus reference on P2.
Dimensions:	Type II
Construction:	High temp ROHS compatible FR4 Multi-Layer Printed Circuit, Through Hole and SMT.
Temperature Coefficient:	.89 W/^{o}C for uniform heat across IP
Power:	Typical **180** mA @ 5V typical.
Temperature Range	–40C⇔85C or better rated components. Conformal Coating option for condensing environments

Order Information

IP-Parallel-TTL-DPR	IP Module with 48 TTL IO. Programmable GPIO function with COS or Level based interrupts and status, Direct and Filterered data. Bit level Direction, Polarity, Rising and Falling control. 40 bit Dual Port RAM interface with 20 bits of address, 16 data, Read, Write, Ack and Busy controls.
-CC	Conformal Coating option
-ROHS	Change to ROHS processing. Without this option, standard leaded solder will be used.
Eng Kit–IP-Parallel-TTL-DPR	IP-Debug-IO - IO connector breakout IP-Debug-Bus - IP Bus interface extender IP-Parallel-TTL-DPR Driver and reference software

Note: The Engineering Kit is strongly recommended for first time purchases.