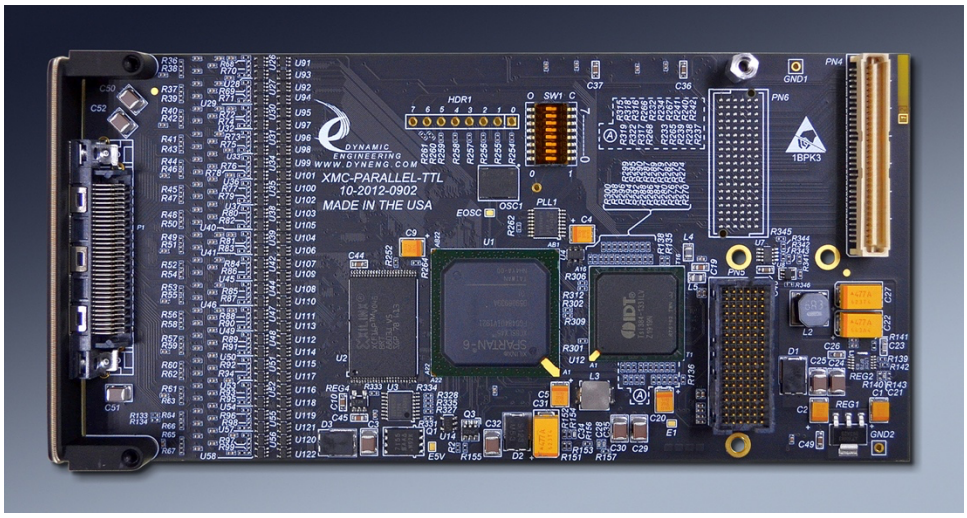
DYNAMIC ENGINEERING
150 DuBois St. Suite C, Santa Cruz, Ca 95060
831-457-8891 https://www.dyneng.com
sales@dyneng.com
Est. 1988

User Manual

XMC-Parallel-TTL-GPIO
PMC-Parallel-TTL-GPIO

Digital Parallel Interface
XMC and PMC Modules
64 bit General Purpose 3V & 5V IO with COS



XMC model Rev 02 shown with Bezel and Pn4 IO

Revision 01P1
Corresponding Hardware: Revision 2
PMC 10-2007-0102, XMC 10-2012-0902
FLASH 0101

PARALLEL-TTL-GPIO

Digital Parallel Interface

Dynamic Engineering
150 DuBois St. Suite C, Santa Cruz CA 95060
831-457-8891

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 PMC Module carriers and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.

Revised 4/21/21

Table of Contents

List of Figures

Product Description

In embedded systems many of the interconnections are made with single ended TTL or CMOS level signals. Depending on the system architecture an IP, PMC, or XMC will be the right choice to make the connection. With most architectures you have a choice as there are carriers for PCIe, PCI, cPCI, VPX, VME, PC/104p and other buses for XMC, PMC, and IP mezzanine modules.

Usually the choice is based on other system constraints as XMC, PMC, and IP can provide the IO you require. Dynamic Engineering would be happy to assist in your decision regarding architecture and other trade-offs with the XMC/ PMC / IP decision. Dynamic Engineering has carriers for XMC, IP, and PMC modules for most architectures, and is adding more as new solutions are requested and required by our customers.

The XMC compatible XMC-Parallel-TTL-GPIO and PMC compatible PMC-Parallel-TTL have 64 independent digital IO. Hereafter referred to as Parallel-TTL-GPIO. The high density makes efficient use of slot resources. The IO is available for system connection through the front panel, via the rear [Pn4] connector, or both. A high density 68 pin VHDCI front panel connector provides the front panel IO. The IO lines can be protected with optional transorbs. The rear panel IO has a PIM and PIM Carrier available for rear panel wiring options in some systems.

With the revision 02 PCB a larger industrial temperature FPGA is standard along with temperature sensor, 50 MHz PCI operation, programmable IO voltage, and on-board FLASH reprogramming. See SW package for examples of how to use the new features.

Parallel-TTL-GPIO provides 64 IO. Each IO is independent with programming options for direction, polarity, level or edge triggered with separate rising and falling enables. In addition, a choice of reference clocks with the local 50 MHz oscillator or programmable PLL. 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 and Linux provide references for all of the modes of operation including setting up clocking, interrupts, using the parallel ports etc.

The HDEterm68 https://www.dyneng.com/HDEterm68.html
can be used as a breakout for the front or rear panel IO. The HDEcabl68 provides a convenient cable. https://www.dyneng.com/HDEcabl68.html VHDCI to SCSI adapter cables are available. A VHDCI based “Term68” is planned. Custom cables can be manufactured to your requirements. Please contact Dynamic Engineering with your specifications.

Each channel is programmable to be input or output on a channel-by-channel basis. All 64 IO channels can be used as interrupt generators. Interrupts are programmable to be based on rising, falling and change of state [both] conditions. The interrupts are

maskable to allow polled operation as well.

The inputs are available unfiltered and after the transition detection. The transition detection is programmable for clock rate. The local 50 MHz oscillator, PCI or external clocks can be selected as the reference to the clock divider. The clock divider is programmable to use the reference rate or to divide it to a lower frequency.

All of the IO are routed through the FPGA to allow for custom applications that require hardware intervention or specific timing- for example an automatic address or data strobe to be generated.

The IO are driven with open-drain high current drivers. When enabled, the high side is driven with the device and augmented with pull-up resistors. When disabled the output is pulled high with the resistors unless another device on the line is driving that line low.

The driver can source & sink 24+ mA. The pull-up default value is for 470 ohms. The resistors are referenced to either 5V or 3.3V based on software selection. For custom models with alternate values please contact Dynamic Engineering.

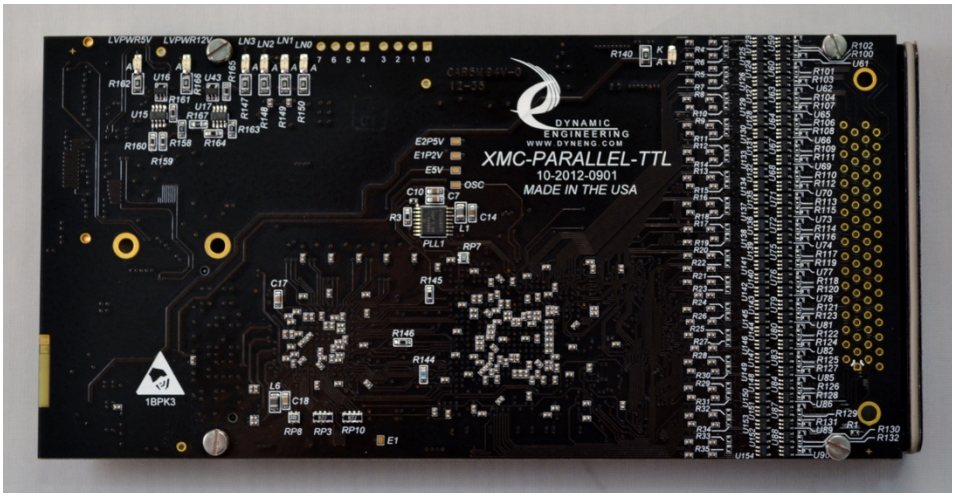


Figure 1 XMC-PARALLEL-TTL-GPIO REAR VIEW

Each line driver is a separate Single Bus Buffer Gate – LVC125. By using separate gates all lines can be enabled or disabled without overheating a package with multiple drivers. The separated gates also allow for individual control for the enable signal.

Each signal also has a separate LVC17 receiver with hysteresis. Internal loop-back is supported and can be used for BIT purposes. Please note: external connections can affect the value read-back when performing internal loop-back.

The registers are mapped as 32 bit words. All registers are read-writeable. The Linux and Windows® compatible drivers are available to provide the system level interface for this design. Use standard C/C++ to control your hardware or use the Hardware manual to make your own software interface. The software manuals are also available on-line. The Linux documentation is provided in-line with the source code.

The basic functions of parallel IO and COS capture are designed into the “GPIO” model.

XMC-Parallel-TTL and PMC-Parallel-TTL are part of the Mezzanine Module family of modular I/O components. XMC-Parallel-TTL conforms to the XMC standard. PMC-Parallel-TTL conforms to the PMC standard. This guarantees compatibility with multiple Carrier boards. Because the mezzanine 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, use PCIe8LXMCX1 to develop your SW in a standard PCIe slot and then port to your target HW. [XMC model]

Theory of Operation

Parallel-TTL-GPIO can be used for multiple purposes with applications in telecommunications, control, sensors, IO, test; anywhere multiple independent or coordinated IO are useful.

The designs feature a Xilinx Spartan 6 FPGA, high current LVC drivers, and LVC Schmidt trigger receivers with hysteresis. The design utilizes a bridge to convert between PCIe and PCI for the XMC model allowing quick porting of designs between the two platforms. The FPGA contains the PCI interface and control required for the parallel interface.

The Xilinx design incorporates the “PCI Core” and additional modules for DMA in parallel with a direct register decoded programming model. The initial implementation provides an enhanced feature set based on the PMC-Parallel-TTL design. Designs can be ported between the PMC and XMC implementations. Additional FLASH updates will provide new features.

The drivers are initialized to the off state and pull-ups on board hold the IO lines in the ‘high’ state. The direction registers are used to program the channel to be a driver or not. The receivers are always enabled allowing local read-back of the transmitted data.

Data written to the IO registers can be placed on the bus. The master enable allows all 64 bits to be synchronized. The master enable can be programmed “on” to allow direct updates if 64 bit synchronization is not required.

For an IO with the direction bit set and master enabled: When a ‘0’ is written to any IO line register position the corresponding line is driven low. When a ‘1’ is written to any IO line register position that line is driven high by the local driver. The 470 pull-up resistor to 3.3/5 will provide additional “source current”, and level control when in “open drain” mode [programmed for receive].

If the direction bit is set to input, the level will be controlled by external devices and the attached pull-ups. The control register is read-writeable. The data register read corresponds to the IO side. The register read-back is at an alternate address offset. The register read-back is independent of the bus; the data read will always match the data written. The IO data read will reflect the state of the bus and not necessarily the state of the on-board drivers.

The read-back registers are clocked at a programmable rate with an internal clock generator. If desired the internal clock can be replaced with an external source and an enable. The basic option is available under SW control. If special programming is needed please contact Dynamic Engineering for a custom FPGA implementation.

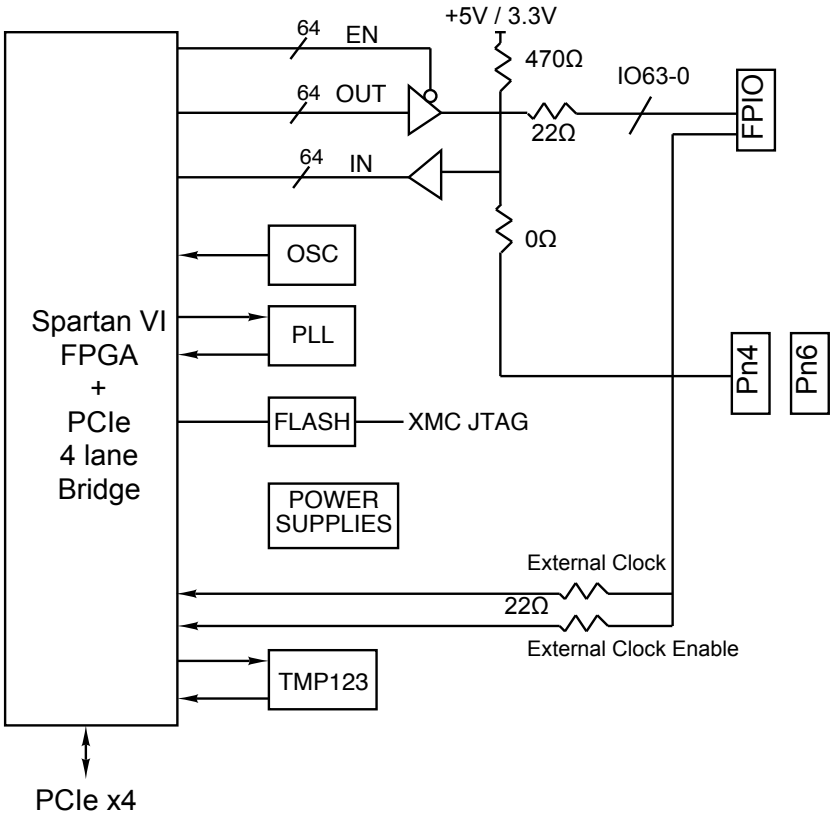


Figure 2 **XMC-Parallel-TTL-GPIO Block Diagram**

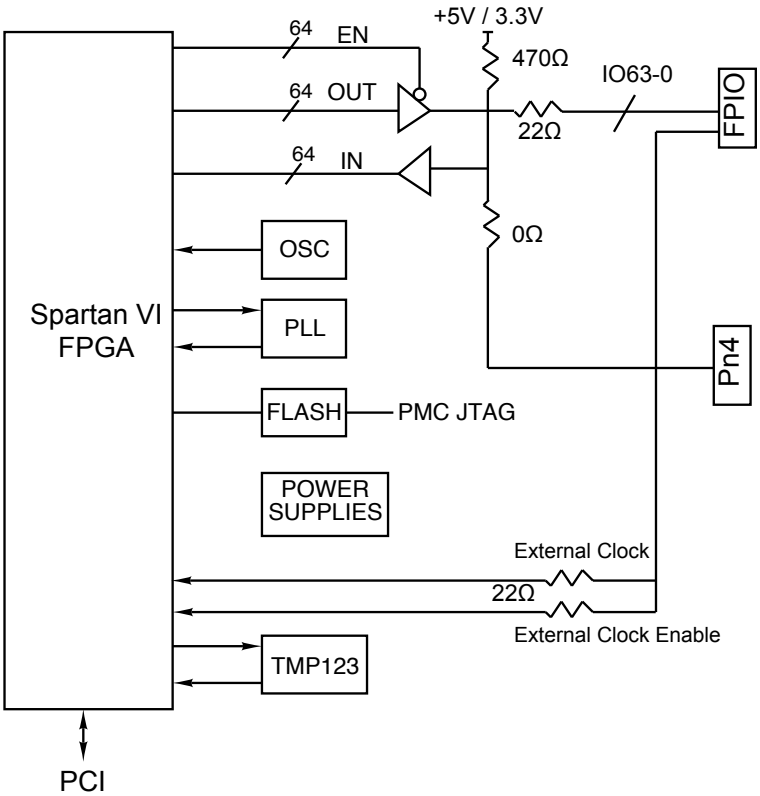


Figure 3 **PMC-Parallel-TTL-GPIO Block Diagram**

Additional detail applies to both block diagrams.
1. The 5V/3.3V selection is SW programmable. FETs etc. are not shown.
2. The isolation resistors are strategically placed to eliminate stubs to the unused connections. The lengths are matched to both connectors.
3. The receivers are hysteresis types.

Address Map

Function	Offset
// Parallel-TTL-GPIO definitions	
#define PAR_TTL_GPIO_BASE	0x0000 // 0 Base control register offset
#define PAR_TTL_GPIO_BASE1	0x0004 // 1 2nd Base control with Master Interrupt En
#define PAR_TTL_GPIO_STATUS	0x0008 // 2 Status Register
#define PAR_TTL_GPIO_SWITCH	0x000C // 3 Type, Revision, Dip Switch
#define spare	0x0010 // 4 unused currently
#define spare	0x0014 // 5 unused currently
#define PAR_TTL_GPIO_DataOut0	0x0018 // 6 Data Out 31-0
#define PAR_TTL_GPIO_DataOut1	0x001C // 7 Data Out 63-32
#define PAR_TTL_GPIO_DataEn0	0x0020 // 8 Data Enable 31-0 '1' = Tx
#define PAR_TTL_GPIO_DataEn1	0x0024 // 9 Data Enable 63-32
#define PAR_TTL_GPIO_Polarity0	0x0028 // 10 Polarity Selection 31-0 '1' = inverted
#define PAR_TTL_GPIO_Polarity1	0x002C // 11 Polarity Selection 63-32
#define PAR_TTL_GPIO_EdgeLevel0	0x0030 // 12 Edge or Level 31-0 '1' = Edge
#define PAR_TTL_GPIO_EdgeLevel1	0x0034 // 13 Edge or Level 63-32
#define PAR_TTL_GPIO_IntEnable0	0x0038 // 14 Interrupt Enables 31-0 '1' = Enabled
#define PAR_TTL_GPIO_IntEnable1	0x003C // 15 Interrupt Enables 63-32
#define PAR_TTL_GPIO_Rising0	0x0040 // 16 Rising Edge Capture 31-0 '1' = Enabled
#define PAR_TTL_GPIO_Rising1	0x0044 // 17 Rising Edge Capture 63-32
#define PAR_TTL_GPIO_Falling0	0x0048 // 18 Falling Edge Capture 31-0 '1' = Enabled
#define PAR_TTL_GPIO_Falling1	0x004C // 19 Falling Edge Capture 63-32
#define PAR_TTL_GPIO_CosRising0	0x0050 // 20 Status set when Rising Edge Captured 31-0
#define PAR_TTL_GPIO_CosRising1	0x0054 // 21 Status set when Rising Edge Captured 63-32
#define PAR_TTL_GPIO_CosFalling0	0x0058 // 22 Status set when Falling Edge Captured 31-0
#define PAR_TTL_GPIO_CosFalling1	0x005C // 23 Status set when Falling Edge Captured 63-32
#define PAR_TTL_GPIO_Direct0	0x0060 // 24 IO Data Synchronized and unfiltered 31-0
#define PAR_TTL_GPIO_Direct1	0x0064 // 25 IO Data Synchronized and unfiltered 63-32
#define PAR_TTL_GPIO_Filtered0	0x0068 // 26 IO Data Synchronized, Polarized, Masked 31-0
#define PAR_TTL_GPIO_Filtered1	0x006C // 27 IO Data Synchronized, Polarized, Masked 63-32
#define PAR_TTL_GPIO_HalfDiv	0x0070 // 28 Divisor for COS clock reference
#define PAR_TTL_GPIO_TempData	0x0074 // 29 Current Temperature
#define PAR_TTL_GPIO_FIFO	0x0078 // 30 Single Word Access ⇔ FIFO 8Kx32
#define spare	0x007C // 31 Currently unused
#define PAR_TTL_WR_DMA_PNTR	0x0080 // 32 Write Physical Address to start DMA Tx
#define PAR_TTL_RD_DMA_PNTR	0x0084 // 33 Write Physical Address to start DMA Rx
#define spare	0x0088 // 34 reserved for Tx FIFO Count
#define spare	0x008C // 35 reserved for Rx FIFO Count
#define spare	0x0090 // 36 Reserved for Tx FIFO Total Count
#define PAR_TTL_GPIO_RX_FIFO_TOTAL_COUNT	0x0094 // 37 Number of Data Available

Figure 4 PARALLEL-TTL-GPIO Address Map

The address map provided is for the local decoding performed within Parallel-TTL-

GPIO. The addresses are all offsets from a base address. The upstream device provides the base address. Dynamic Engineering prefers a long-word oriented approach because it is more consistent across platforms.

The map is presented with the #define style to allow cutting and pasting into many compilers “include” files.

The host system will search the PCI bus to find the assets installed during power-on initialization.

PMC-Parallel-TTL-GPIO: VendorId = 0xDCBA and the CardId = 0x006E
XMC-Parallel-TTL-GPIO: VendorId = 0xDCBA and the CardId = 0x006F

The Type field has a unique number allowing SW to read the type currently connected to. The UserAp package makes use of this feature to print the correct board type on the menu. The OS uses the CardId to index into the INF and supply PMC or XMC in the card description in the device manager [Windows]. Linux and other systems also make use of the data.

Programming

Programming the Parallel-TTL-GPIO requires the ability to read and write data in the host's memory space.

Once the initialization process has occurred, and the system has assigned addresses to Parallel-TTL-GPIO, software will need to determine what the address space is for the PCI interface [BAR0]. The offsets in the address table are relative to the system assigned BAR0 base address.

The next step is to initialize Parallel-TTL-GPIO. If the basic mode of direct read and write operations is to be used the default settings can be used except for setting the master output enable and the direction bits corresponding to the channels to transmit on.

If COS inputs are to be used the reference and divisor clocks may require programming. In many cases the default settings will work. In addition, the Rising, Falling, and Interrupt capabilities need to be programmed. Once the settings are in place it is recommended the receive state registers are written to for clearing purposes as the programming steps may cause phantom events to be captured.

One additional programming step will be to initialize the PLL to the user desired frequency if selected instead of the local oscillator.

For Windows™ and Linux systems the Dynamic Driver can be used. The driver will take care of finding the hardware and provide an easy-to-use mechanism to program the hardware. The Driver comes with reference software showing how to use the card and reference frequency files to allow the user to duplicate the test set-up used in manufacturing at Dynamic Engineering. Using simple, known to work routines is a good way to get acquainted with new hardware.

DMA is provided to support test of Dynamic Engineering carriers using DMA. The DMA function is not normally used with the current GPIO implementation. DMA could be added to the Tx, Rx or both to provide consistent rate of update for output or sample for input. Please contact Dynamic Engineering [engineering@dyneng.com] for any modifications we can make to support your project.

Register Definitions

ParTtlGpio_BASE

[$00 Base Control Register Port read/write]

DATA BIT	DESCRIPTION
31	VIOSEL
30-18	Spare
17	DMA_RdEn
16	DMA_WrEn
15	PLL_SDAT [write to PLL, read-back from PLL]
14	PLL_SCLK
13	PLL_EN
12	ClkCosSel
11	TMS_IP
10	TCK_IP
9	TDI_IP
8	JTAG_MUX_SEL
7	TDOS
-5	spare
4	Force Interrupt
3	spare
2	spare
1	IoRst
0	MasterTxEn

Figure 5 PARALLEL-TTL-GPIO Control port 0 Bit Map

The base control register for Parallel-TTL-GPIO is used for programming the PLL and other design level controls. Unused bits are reserved for additional new features. Unused bits should be programmed '0' to allow for future commonality.

MasterTxEn is used to allow the upper and lower data to be synchronized. The upper 32 bits and the lower 32 bits are not accessed at the same time. If the user wants to have the upper and lower data change at the same time the Master enable can be cleared to '0', both halves of the data written and then the enable set '1'. If synchronization is not an issue; program to '1' as part of initialization.

IoRst When set '1' resets various HW functions. FIFOs, State-machines etc. In most cases the control registers are unaffected. Clear to '0' for normal operation.

Force Interrupt when '1' and the master enabled will cause an interrupt request. The interrupt can be cleared by clearing this bit or disabling the master interrupt enable or both. Force Interrupt is used for test and software development purposes.

PLL_EN: When this bit is set to a one, the signals used to program and read the PLL are enabled. PLL_SDAT is driven low when PLL_EN is '1' and PLL_SDAT is written '0'

else Tristated. Place in tristate mode to read data back from the PLL.

PLL_SCLK/PLL_SDAT : These signals are used to program the PLL over the I^2C serial interface. SCLK is always an output whereas SDATA is bi-directional. Park the Data bit '1' to read back with SCLK.

The PLL is programmed with the output file generated by the Cypress PLL programming tool. [CY3672 R3.01 Programming Kit or CyberClocks R3.20.00 Cypress may update the revision from time to time.]

The .JED file is used by the Dynamic Driver to program the PLL. Programming the PLL is fairly involved and beyond the scope of this manual. For clients writing their own drivers it is suggested to get the Engineering Kit for this board including software, and to use the translation and programming files ported to your environment. This procedure will save you a lot of time. For those who want to do it themselves the Cypress PLL in use is the 22393. The output file from the Cypress tool can be passed directly to the Dynamic Driver [Linux or Windows] and used to program the PLL without user intervention.

The reference frequency for the PLL is 50 MHz.

JTAG signals are used to reprogram, verify, erase etc. the FLASH used to load the FPGA. JTAG Mux Sel when set causes the JTAG mux to connect the FPGA control to the FLASH device. When '0' the external programming header is selected. TDI, TMS, and CLK are used to control the actions of the state-machine within the FLASH and to transfer data. TDO is used to read the serial data returned from the FLASH. We are adding this feature to our drivers. Currently not supported.

VIOSEL is used to set the reference for the TTL IO. Each IO has a pull-up resistor and the resistor can be tied to 3.3V or 5V. When cleared 3.3V is used as the reference. For 5V set this bit.

ParTtlGpio_BASE1

[$04 Master Interrupt Control]

DATA BIT	DESCRIPTION
31-1	spare
0	Master Interrupt Enable

Figure 6 PARALLEL-TTL-GPIO Master Interrupt Enable

Master Interrupt Enable when '1' gates active interrupt requesting conditions onto Interrupt Request A. When set to '0' the interrupting functions are available as status but no interrupt request is generated by the card to allow for polled operation.

ParTtlGpio_STATUS

[$08 Board level Status Port read only]

DATA BIT	DESCRIPTION
31	Interrupt Status
30-20	spare
19	Dma_RdInt
18	DMA_WrInt
17	DMA_RdErr
16	DMA_WrErr
15-5	spare
4	IntForce
3	LevelIntReq
2	CosFallingIntReq
1	CosRisingIntReq
0	local interrupt

Figure 7 PARALLEL-TTL-GPIO Status Port Bit Map

Local Interrupt is set when any of the interrupt types is set – unmasked for non DMA interrupt types.

CosRisingIntReq - This is the logical OR of the COS outputs for the Rising Edge condition. The RISING register will select which bits are enabled. If any of the enabled bits are active this bit is set. The status is captured before the master interrupt enable. If the master interrupt enable is set an interrupt will be generated if this condition is true.

CosFallingIntReq - This is the logical OR of the COS outputs for the Falling Edge condition. The Falling register will select which bits can be active [enabled]. If any of the enabled bits capture a falling edge this bit will be set. The status is captured before the master interrupt enable. If the master interrupt enable is set an interrupt will be generated if this condition is true.

LevelIntReq is set when an IO bit meets the level interrupt criterion specified by the Polarity, Edge/Level, and Level Interrupt enable. The status is captured before the master interrupt enable. If the master interrupt enable is set an interrupt will be generated if this condition is true.

IntForce is a copy of the IntForce bit from the Base Control Register. This status bit is included to allow a single register read to determine all interrupt types for the design. Clear in the base register. If the master interrupt enable is set an interrupt will be generated if this condition is true.

Write DMA Error Occurred: When a one is read, a write DMA error has been detected. This will occur if there is a target or master abort or if the direction bit in the next pointer of one of the chaining descriptors is a one. A zero indicates that no write DMA error has occurred. This bit is latched and can be cleared by writing back to the Status register with a one in this bit position.

Read DMA Error Occurred: When a one is read, a read DMA error has been detected. This will occur if there is a target or master abort or if the direction bit in the next pointer of one of the chaining descriptors is a zero. A zero indicates that no read DMA error has occurred. This bit is latched and can be cleared by writing back to the Status register with a one in this bit position.

Write DMA Interrupt Occurred: When a one is read, a write DMA interrupt is latched. This indicates that the scatter-gather list for the current write DMA has completed, but the associated interrupt has yet to be completely processed. A zero indicates that no write DMA interrupt is pending.

Read DMA Interrupt Occurred: When a one is read, it indicates that a read DMA interrupt is latched. This indicates that the scatter-gather list for the current read DMA has completed, but the associated interrupt has yet to be completely processed. A zero indicates that no read DMA interrupt is pending.

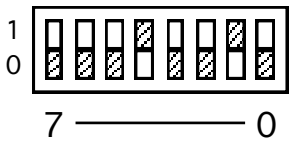
Interrupt Status – Set if the PCI interrupt is asserted. This bit can be checked to determine if this card is causing an interrupt to the system. If set the other bits can be checked to see which feature(s) of the board need to be serviced. Secondary reads to the COS etc. to determine the exact type of interrupt.

ParTtlGpio _SWITCH
[$0C Switch and Revision number port read only]

DATA BIT	DESCRIPTION
31-24	Type
23-16	Revision Major
15-8	Revision Minor
7-0	DIP switch

Figure 8 PARALLEL-TTL-GPIO Revision, Switch

The DIP Switch is labeled for bit number and '1' '0' in the silk screen. The DIP Switch can be read from this port and used to determine which Parallel-TTL-GPIO is being addressed in a system with multiple cards installed. The DIPswitch can also be used for other purposes – software revision etc. The switch shown would read back 0x12.



RevisionMajor and RevisionMinor are stored and allow SW to determine the feature set of the installed card. Major revisions are updated when large changes occur – new feature etc. Minor Revisions are updated anytime a new FLASH is released. The GPIO Major revision is x01 and the minor revision is x01 currently.

The Type field is provided to allow a read in user space to determine the card type the GPIO is implemented on.

Type	PCB
1	PMC-Parallel-TTL
2	XMC-Parallel-TTL

Revision History.
0x0101 initial major release, add GPIO function 2/2/21

Figure 9 PARALLEL-TTL-GPIO Revision History

ParTtlGpio_DataOut0

[$18 Data IO Port read/write]

DATA BIT	DESCRIPTION
31-0	Data Out 31-0

Figure 10 PARALLEL-TTL-GPIO Data IO Lower Bit Map

ParTtlGpio_DataOut1

[$1C Data IO Port read/write]

DATA BIT	DESCRIPTION
31-0	Data Out 63-32

Figure 11 PARALLEL-TTL-GPIO Data IO Upper Bit Map

The data to be transmitted is written to the Data Output Port side of the Data Register. Reading from this port will return the value of the port independent of other settings. Please see the Direct and Filtered ports for IO side data read.

The output bits are driven onto the IO for the bits that are enabled with the Enable control register, and when the master parallel enable is set. For bits without the Enable register bit set there are no side effects. The Enable register will act as a mask for the data register.

ParTtlGpio_DataEn0

[$20 Enable Register bits 31-0 read – write]

DATA BIT	DESCRIPTION
31-0	En31-0

Figure 12 PARALLEL-TTL-GPIO Enable Lower Bit Map

The lower 32 bits of the parallel port direction are controlled with this port. When reset this port is cleared 0x00000000. All IO are set to read [inputs]. To use one or more of the IO for outputs; program the corresponding Enable bit(s) to '1'.

ParTtlGpio_DataEn1

[$24 Enable Register bits 63-32 read – write]

DATA BIT	DESCRIPTION
31-0	En63-32

Figure 13 PARALLEL-TTL-GPIO Enable Upper Bit Map

The upper 32 bits of the parallel port direction are controlled with this port. When reset this port is cleared 0x00000000. All IO are set to read [inputs]. To use one or more of the IO for outputs; program the corresponding direction bit(s) to '1'.

Once a Direction bit is set to output the data in the corresponding output holding register bit is broadcast on that IO line. The data in the holding register will match the data in the data output register if the master parallel enable bit is set. If initial states are important you may want to program the initial data and enable it before enabling the direction bits.

ParTtlGpio_Polarity0

[$28 Polarity Reg Port read only]

DATA BIT	DESCRIPTION
31-0	Polarity 31-0

Figure 14 PARALLEL-TTL-GPIO Polarity Reg Lower Bit Map

ParTtlGpio_EdgeLevel1

[$2C Polarity Reg Port read only]

DATA BIT	DESCRIPTION
31-0	Polarity IO 63-32

Figure 15 PARALLEL-TTL-GPIO Polarity Reg Upper Bit Map

Data written to the Polarity registers can be read back through this port. Each Polarity control bit corresponds to the associated IO bit. When '1' Inverted processing is selected. When '0' non-inverted processing of the IO bit is selected.

ParTtlGpio_EdgeLevel0

[$30 EdgeLevel Reg Port read only]

DATA BIT	DESCRIPTION
31-0	EdgeLevel 31-0

Figure 16 PARALLEL-TTL-GPIO EdgeLevel Reg Lower Bit Map

ParTtlGpio_EdgeLevel1

[$34 EdgeLevel Reg Port read only]

DATA BIT	DESCRIPTION
31-0	EdgeLevel IO 63-32

Figure 17 PARALLEL-TTL-GPIO EdgeLevel Reg Upper Bit Map

Data written to the EdgeLevel registers can be read back through this port. Each EdgeLevel control bit corresponds to the associated IO bit. When '1' Edge processing is selected. When '0' level processing is selected. See Rising and Falling control registers when Edge is selected. See Polarity registers when level is selected.

ParTtlGpio_IntEnable0

[$38 IntEnable Reg Port read only]

DATA BIT	DESCRIPTION
31-0	IntEnable 31-0

Figure 18 PARALLEL-TTL-GPIO IntEnable Reg Lower Bit Map

ParTtlGpio_IntEnable1

[$3C IntEnable Reg Port read only]

DATA BIT	DESCRIPTION
31-0	IntEnable IO 63-32

Figure 19 PARALLEL-TTL-GPIO IntEnable Reg Upper Bit Map

Data written to the IntEnable registers can be read back through this port. Each IntEnable control bit corresponds to the associated IO bit. When '1' the interrupt from that IO is enabled. Affects both Edge and Level selected IO types.

ParTtlGpio_Rising0

$40 Rising Lower Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Rising 31-0

Figure 20 PARALLEL-TTL-GPIO Rising Lower Bit Map

ParTtlGpio_Rising1

$44 Rising Upper Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Rising 63-32

Figure 21 PARALLEL-TTL-GPIO Rising Upper Bit Map

The Rising control register bits correspond to the input data bits. All IO can be set-up for COS activity even if defined as an output. Please see EdgeLevel definition register. In most cases the output bits will be set to '0' for the Rising register. When set '1' and the corresponding input bit transitions from low to high the COS register of rising activity will have the corresponding bit set. If the separate interrupt enable bit is also set an interrupt can be generated. The Rising register is a control register. The COS data is read back separately.

ParTtlGpio_Falling0

$48 Falling Lower Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Falling 31-0

Figure 22 PARALLEL-TTL-GPIO Falling Lower Bit Map

ParTtlGpio_Falling1

$4C Falling Upper Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Falling 63-32

Figure 23 PARALLEL-TTL-GPIO Falling Upper Bit Map

The Falling control register bits correspond to the input data bits. All IO can be set-up for COS activity even if defined as an output. In most cases the output bits will be set to '0' for the Falling register. When set '1' and the corresponding input bit transitions from High to Low the COS register of falling activity will have the corresponding bit set. If the separate interrupt enable bit is also set an interrupt can be generated. The Falling register is a control register. The COS data is read back separately.

ParTtlGpio_CosRising0

$50 COS Rising Lower Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	COS Rising 31-0

Figure 24 PARALLEL-TTL-GPIO COS Rising Lower Bit Map

ParTtlGpio_CosRising1

$54 COS Rising Upper Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	COS Rising 63-32

Figure 25 PARALLEL-TTL-GPIO COS Rising Upper Bit Map

The COS Rising Edge data is available to read from this port. Write back to clear latched data.

ParTtlGpio_CosFalling0

$58 COS Falling Lower Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	COS Falling 31-0

Figure 26 PARALLEL-TTL-GPIO COS Falling Lower Bit Map

ParTtlGpio_CosFalling1

$5C COS Falling Upper Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	COS Falling 63-32

Figure 27 PARALLEL-TTL-GPIO COS Falling Upper Bit Map

The COS Falling Edge data is available to read from this port. Write back to clear latched data.

ParTtlGpio_Direct0

$60 Direct Data Lower Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Direct 31-0

Figure 28 PARALLEL-TTL-GPIO Direct Read Lower Bit Map

ParTtlGpio_Direct1

$64 Direct Data Upper Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Direct 63-32

Figure 29 PARALLEL-TTL-GPIO Direct Read Upper Bit Map

The IO is synchronized to the PCI reference clock and made available from this port. No filtering is performed.

ParTtlGpio_Filtered0

$68 Filtered Data Lower Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Filtered 31-0

Figure 30 PARALLEL-TTL-GPIO Filtered Read Lower Bit Map

ParTtlGpio_Filtered1

$6C Filtered Data Upper Control Register Port read/write

DATA BIT	DESCRIPTION
31-0	Filtered 63-32

Figure 31 PARALLEL-TTL-GPIO Filtered Read Upper Bit Map

The IO is synchronized to the PCI reference clock, polarized and masked [POL and EdgeLevel].

ParTtlGpio_HalfDiv

[$70 COS clock definition port read -write]

DATA BIT	DESCRIPTION
31-16	spare
15-0	DIVISOR

Figure 32 PARALLEL-TTL-GPIO COS Clk Control Bit Map

DIVISOR[15-0] are the clock divisor select bits. The clock source is divided by a 16-bit counter. The output frequency is {reference / [2*N], n≥1. The counter operates from 1 ⇔ N. A pulse is generated when the counter reaches the end point, and the pulse used to create the output clock. The output clock is a square wave as a result.

Oscillator is the reference for the divider.

For a desired frequency of 1 MHz. The required divisor is 50 => N = 25.

Please see ClkCosSel to use this reference or a user programmed PLLA output. The Windows and Linux SW packages support user defined .jed files to program the PLL to custom frequencies. See PLL [Base Reg] for more information.

ParTtlGpio_TempData

[$74 Temperature Interface port read -write]

DATA BIT	DESCRIPTION
31-14	spare
13	Ready
12	Sign
11-0	Data

Figure 33 PARALLEL-TTL-GPIO TMP123 Interface Bit Map

TMP123 is a 10 MHz SPI device. The oscillator is used as a reference to the controller built into the GPIO. Write any value to this port to start the state machine. Poll the Ready bit, and when '1' read the data. 0.0625C is the bit value. 2's compliment data. -40 ⇔ +125C with 2C accuracy. 640 mS max conversion period. 320 mS max conversion time. Do not over sample as the conversion is restarted if a new request is presented before the previous one completes. The conversion to the next value happens after the previous one is read. If reading as a monitor – once a second for example the data will not be stale, meet the conversion times etc. If reading more sporadically the first data should be tossed as stale and the 2nd used to make sure current.

ParTtlGpio_FIFO

[$78 FIFO Interface port read -write]

DATA BIT	DESCRIPTION
31-0	FIFO Data

Figure 34 PARALLEL-TTL-GPIO FIFO Interface Bit Map

GPIO incorporates an 8Kx32 FIFO for DMA testing purposes. Data can be written to this port and later read from the same port. In addition, this FIFO is used with the DMA engine to load and unload data. Caution, Reading data from the port removes it as it is a FIFO.

This port can be redirected to support IO requirements. Please contact Dynamic Engineering for this option

ParTtlGpio_WR_DMA_PNTR

[$80 Write DMA Pointer (write only)

DMA Pointer Address Register

Data Bit	Description
31-2	First Chaining Descriptor Physical Address
1	direction [0]
0	end of chain

Figure 35 Parallel-TTL-GPIO Write DMA pointer register

This write-only port is used to initiate a scatter-gather write [TX] DMA. When the address of the first chaining descriptor is written to this port, the DMA engine reads three successive long words beginning at that address. Essentially this data acts like a chaining descriptor value pointing to the next value in the chain.

The first is the address of the first memory block of the DMA buffer containing the data to read into the device, the second is the length in bytes of that block, and the third is the address of the next chaining descriptor in the list of buffer memory blocks. This process is continued until the end-of-chain bit in one of the next pointer values read indicates that it is the last chaining descriptor in the list.

All three values are on LW boundaries and are LW in size. Addresses for successive parameters are incremented. The addresses are physical addresses the HW will use on the PCI bus to access the Host memory for the next descriptor or to read the data to be transmitted. In most OS you will need to convert from virtual to physical. The length parameter is a number of bytes, and must be on a LW divisible number of bytes.

Status for the DMA activity can be found in the channel control register and channel status register.

Notes:

1. Writing a zero to this port will abort a write DMA in progress.
2. End of chain should not be set for the address written to the DMA Pointer Address Register. End of chain should be set when the descriptor follows the last length parameter.
3. The Direction should be set to ‘0’ for Burst In DMA in all chaining descriptor locations.

ParTtlGpio_RD_DMA_PNTR

[$84] Read DMA Pointer (write only)

DMA Pointer Address Register

Data Bit	Description
31-2	First Chaining Descriptor Physical Address
1	direction [1]
0	end of chain

Figure 36 Parallel-TTL-GPIO Read DMA pointer register

This write-only port is used to initiate a scatter-gather read [RX] DMA. When the address of the first chaining descriptor is written to this port, the DMA engine reads three successive long words beginning at that address. Essentially this data acts like a chaining descriptor value pointing to the next value in the chain.

The first is the address of the first memory block of the DMA buffer to write data from the device to, the second is the length in bytes of that block, and the third is the address of the next chaining descriptor in the list of buffer memory blocks. This process is continued until the end-of-chain bit in one of the next pointer values read indicates that it is the last chaining descriptor in the list.

All three values are on LW boundaries and are LW in size. Addresses for successive parameters are incremented. The addresses are physical addresses the HW will use on the PCI bus to access the Host memory for the next descriptor or to read the data to be transmitted. In most OS you will need to convert from virtual to physical. The length parameter is a number of bytes, and must be on a LW divisible number of bytes.

Status for the DMA activity can be found in the channel control register and channel status register.

Notes:

1. Writing a zero to this port will abort a write DMA in progress.
2. End of chain should not be set for the address written to the DMA Pointer Address Register. End of chain should be set when the descriptor follows the last length parameter.
3. The Direction should be set to ‘1’ for Burst Out DMA in all chaining descriptor locations.

ParTtlGpio_PAR_TTL_GPIO_RX_FIFO_TOTAL_COUNT

[$94] Read Rx FIFO Total Count

RX and TX FIFO Port	
Data Bit	Description
31-0	FIFO Total Count

Figure 37 Parallel-TTL-GPIO FIFO Total Count

Read the number of words available in the Rx FIFO path. This includes the DMA pipeline.

Loop-back

The Engineering kit has reference software, which includes external loop-back tests. Parallel-TTL-GPIO utilizes a 68 pin VHDCI front panel connector. The tests require an external cable with the following pins connected. Rear IO connection options are also available.

For Parallel Port loop-back the full IO0-31 is cross connected with IO32-63.

External GPIO function Loop-Back

Signal	From	To	Signal
IO_0	pin 33	pin 67	IO_32
IO_1	pin 32	pin 66	IO_33
IO_2	pin 31	pin 65	IO_34
IO_3	pin 30	pin 64	IO_35
IO_4	pin 29	pin 63	IO_36
IO_5	pin 28	pin 62	IO_37
IO_6	pin 27	pin 61	IO_38
IO_7	pin 26	pin 60	IO_39
IO_8	pin 25	pin 59	IO_40
IO_9	pin 24	pin 58	IO_41
IO_16	pin 17	pin 51	IO_48
IO_17	pin 16	pin 50	IO_49
IO_18	pin 15	pin 49	IO_50
IO_19	pin 14	pin 48	IO_51
IO_20	pin 13	pin 47	IO_52
IO_21	pin 12	pin 46	IO_53
IO_22	pin 11	pin 45	IO_54
IO_23	pin 10	pin 44	IO_55
IO_24	pin 9	pin 43	IO_56
IO_25	pin 8	pin 42	IO_57
IO_26	pin 7	pin 41	IO_58
IO_27	pin 6	pin 40	IO_59
IO_28	pin 5	pin 39	IO_60
IO_29	pin 4	pin 38	IO_61
IO_30	pin 3	pin 37	IO_62
IO_31	pin 2	pin 36	IO_63

PMC/XMC Module Front Panel IO Interface Pin Assignment

The figure below gives the pin assignments for the XMC and PMC Module IO Interface on the Parallel-TTL-GPIO. Installed for –FP and –FRP models. Also see the User Manual for your carrier board for more information.

EXT_CLK_EN	EXT_CLK	1	35
IO_31	IO_63	2	36
IO_30	IO_62	3	37
IO_29	IO_61	4	38
IO_28	IO_60	5	39
IO_27	IO_59	6	40
IO_26	IO_58	7	41
IO_25	IO_57	8	42
IO_24	IO_56	9	43
IO_23	IO_55	10	44
IO_22	IO_54	11	45
IO_21	IO_53	12	46
IO_20	IO_52	13	47
IO_19	IO_51	14	48
IO_18	IO_50	15	49
IO_17	IO_49	16	50
IO_16	IO_48	17	51
IO_15	IO_47	18	52
IO_14	IO_46	19	53
IO_13	IO_45	20	54
IO_12	IO_44	21	55
IO_11	IO_43	22	56
IO_10	IO_42	23	57
IO_9	IO_41	24	58
IO_8	IO_40	25	59
IO_7	IO_39	26	60
IO_6	IO_38	27	61
IO_5	IO_37	28	62
IO_4	IO_36	29	63
IO_3	IO_35	30	64
IO_2	IO_34	31	65
IO_1	IO_33	32	66
IO_0	IO_32	33	67
GND	GND	34	68

Figure 38 PARALLEL-TTL-GPIO FRONT PANEL Interface

PMC/XMC Module Backplane Pn4 IO Interface Pin Assignment

The figure below gives the pin assignments for the PMC/XMC Module IO Interface on the Parallel-TTL-GPIO and routed to Pn4. Pn4 installed for –RP and –FRP models. Also see the User Manual for your carrier board for more information.

IO_0	IO_1	1	2
IO_2	IO_3	3	4
IO_4	IO_5	5	6
IO_6	IO_7	7	8
IO_8	IO_9	9	10
IO_10	IO_11	11	12
IO_12	IO_13	13	14
IO_14	IO_15	15	16
IO_16	IO_17	17	18
IO_18	IO_19	19	20
IO_20	IO_21	21	22
IO_22	IO_23	23	24
IO_24	IO_25	25	26
IO_26	IO_27	27	28
IO_28	IO_29	29	30
IO_30	IO_31	31	32
IO_32	IO_33	33	34
IO_34	IO_35	35	36
IO_36	IO_37	37	38
IO_38	IO_39	39	40
IO_40	IO_41	41	42
IO_42	IO_43	43	44
IO_44	IO_45	45	46
IO_46	IO_47	47	48
IO_48	IO_49	49	50
IO_50	IO_51	51	52
IO_52	IO_53	53	54
IO_54	IO_55	55	56
IO_56	IO_57	57	58
IO_58	IO_59	59	60
IO_60	IO_61	61	62
IO_62	IO_63	63	64

Figure 39 PARALLEL-TTL-GPIO PN4 Interface

XMC Module Backplane Pn6 IO Interface Pin Assignment

The figure below gives the pin assignments for the XMC Module IO Interface on the Parallel-TTL-GPIO and routed to Pn6. Pn6 installed for –XIO and –XIOexc models. Also see the User Manual for your carrier board for more information.

IO_0	IO_1	A1	D1
IO_2	IO_3	B1	E1
IO_4	IO_5	C1	F1
IO_6	IO_7	C2	F2
IO_8	IO_9	A3	D3
IO_10	IO_11	B3	E3
IO_12	IO_13	C3	F3
IO_14	IO_15	C4	F4
IO_16	IO_17	A5	D5
IO_18	IO_19	B5	E5
IO_20	IO_21	C5	F5
IO_22	IO_23	C6	F6
IO_24	IO_25	A7	D7
IO_26	IO_27	B7	E7
IO_28	IO_29	C7	F7
IO_30	IO_31	C8	F8
IO_32	IO_33	A9	D9
IO_34	IO_35	B9	E9
IO_36	IO_37	C9	F9
IO_38	IO_39	C10	F10
IO_40	IO_41	A11	D11
IO_42	IO_43	B11	E11
IO_44	IO_45	C11	F11
IO_46	IO_47	C12	F12
IO_48	IO_49	A13	D13
IO_50	IO_51	B13	E13
IO_52	IO_53	A15	D15
IO_54	IO_55	B15	E15
IO_56	IO_57	A17	D17
IO_58	IO_59	B17	E17
IO_60	IO_61	A19	D19
IO_62	IO_63	B19	E19

36 Grounds on standard pins(A2,B2,D2,E2, A4,B4,D4,E4, A6,B6,D6,E6, A8,B8,D8,E8, A10,B10,D10,E10, A12,B12,D12,E12, A14,B14,D14,E14, A16,B16,D16,E16

Figure 40 XMC-PARALLEL-TTL-GPIO PN6 Interface

Interfacing

The pin-out tables are displayed with the pins in the same relative order as the actual connectors. 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.

Power all system power supplies from one switch. Open Drain interface devices provide some immunity from, and allow operation when part of the circuit is powered on and part is not. It is better to avoid the issue of going past the safe operating areas by powering the equipment together and by having a good ground reference.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. Series resistors are used and can be specified to be something other than the 22 ohm standard value. The connector is pinned out for a standard VHDCI cable to be used. Alternatively VHCDI to SCSI cables are available.

Terminal Block. We offer a high quality 68 screw terminal block that directly connects to the SCSI II/III cable. The terminal block can mount on standard DIN rails. HDEterm68 [http://www.dyneng.com/HDEterm68.html] Use with a VHDCI to SCSI cable.

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 particular device's rated voltages.

Construction and Reliability

XMC and PMC Modules were conceived and engineered for rugged industrial environments. Parallel-TTL-GPIO is constructed out of 0.062 inch thick high temperature ROHS compliant material.

The traces are matched length from the FPGA ball to the IO pin. The options for front panel and rear panel are isolated with series resistor packs to eliminate bus stubs when one of the connectors is not in use.

Surface mounted components are used.

The XMC Module connectors are keyed and shrouded with Gold plated pins on both plugs and receptacles. They are rated at 1 Amp per pin, 50 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

The PMC connectors are rated at 1 Amp per pin, 100 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

The module is secured against the carrier with the connectors and front panel. It is recommended to install the mounting screws supplied with the device.

The Module provides a low temperature coefficient of 2.17 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 module. The coefficient means that if 2.17 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

PARALLEL-TTL-GPIO design consists of CMOS circuits. The power dissipation due to internal circuitry is very low. It is possible to create 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 the one degree C 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. http://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 C
Santa Cruz, CA 95060
831-457-8891
support@dyneng.com

Specifications

Logic Interface:	XMC Logic Interface [PCIe ⇔ PCI] 32/50 at PCI. PMC Logic interface is 32 bit 50 MHz capable PCI. 33 standard operating rate.
Digital Parallel IO:	64 discrete IO channels. Each IO can be programmed to be input, output, or both. Inputs can be treated as Edge or Level sensitive. Filtering options include COS, Inversion, Selective interrupts. Software programmable 3.3V or 5V reference for IO.
CLK rates supported:	Osc (50 MHz) & PLL coupled with 16 bit divider to allow user programmed sample rate for COS.
Software Interface:	Control Registers, IO registers, IO Read-Back registers
Initialization:	Programming procedure documented in this manual
Access Modes:	LW to registers, read-write to most registers
Access Time:	Frame to TRDY 4 PCI clocks or burst mode DMA – 1 word per PCI clock transferred. 50/33 MHz reference.
Interrupt:	All IO lines can be used as interrupt sources with programmable rising and or falling activity on IO line “COS”, DMA interrupts
Onboard Options:	All Options are Software Programmable
Interface Options:	68 Pin VHDCI connector at front bezel User IO routed to Pn4/Pn6
Dimensions:	Standard Single XMC/PMC Module.
Construction:	Multi-Layer Printed Circuit, Through Hole and Surface Mount Components.
Temperature Coefficient:	2.17 W/^{o}C for uniform heat across PMC
Power:	TBD mA @ 5V outputs off Add 10 mA per active low output for pull-up current drivers support +/- 24 mA per IO line.

Ordering Information

standard temperature range -40-80ºC

XMC-Parallel-TTL-GPIO	XMC Module with 64 GPIO, programmable IO handling, COS, Level, polarity, interrupts. Temperature sensor. PLL and Oscillator references. https://www.dyneng.com/XMC-Parallel-TTL.html
PMC-Parallel-TTL-GPIO	PMC Module with 64 GPIO, programmable IO handling, COS, Level, polarity, interrupts. Temperature sensor. PLL and Oscillator references. https://www.dyneng.com/PMC-Parallel-TTL.html

Order Options:

–FP for front panel IO only [default if no selection made]
-RP for rear panel IO PN4 only
-**XIO** for rear panel both Pn4 and Pn6 [XMC only option]
-**XIOexc** for Pn6 only [XMC only option]
-CC to add conformal coating
-ROHS to switch to ROHS compliant solder

Adding -RP, -XIO, -XIOexc will remove the default VHDCI connector and install the rear IO options.

Related:

PCIe8LXMCX1: PCIe to XMC adapter to allow installation of XMC-Parallel-TTL-GPIO into a PCIe system.
https://www.dyneng.com/PCIe8LXMCX1.html

PCIeBPMCX1: PCIe to PMC adapter to allow installation of PMC-Parallel-TTL-GPIO into a PCIe system. Note, the 50 MHz PCI option can be selected on this carrier.
https://www.dyneng.com/pciebpmcx1.html

HDEterm68: 68 position terminal block with two SCSI II/III connectors. Parallel-TTL-GPIO compatible with VHDCI to SCSI cable.
https://www.dyneng.com/HDEterm68.html

XMC-UNIV-Test: PCIe to XMC adapter to allow installation of Parallel-TTL-GPIO into a PCIe system. Vertical adapter for debugging etc.
https://www.dyneng.com/XMC-UNIV-TEST.html

PMC-UNIV-Test: PCIe to PMC adapter to allow installation of Parallel-TTL-GPIO into a PCI system. Vertical adapter for debugging etc.
https://www.dyneng.com/PMC-UNIV-TEST.html