Register Descriptions

This appendix is a quick reference that describes registers most commonly accessed by the software driver. For all registers as well as further details, refer to the specific user guides.

The registers implemented in hardware are mapped to base address register (BAR0) in PCI Express®. Table B-1 shows the mapping of multiple Packet DMA channel registers across the BAR.

Table B-1: **Packet DMA Channel Register Address**

Packet DMA Channel	Offset from BAR0
DMA Channel 0 S2C	`0x0`
DMA Channel 1 S2C	`0x100`
DMA Channel 0 C2S	`0x2000`
DMA Channel 1 C2S	`0x2100`

Registers for interrupt handling in the Packet DMA are grouped under a category called common registers, which are offset from BAR0 by `0x4000`.

Figure B-1 shows the layout of registers.
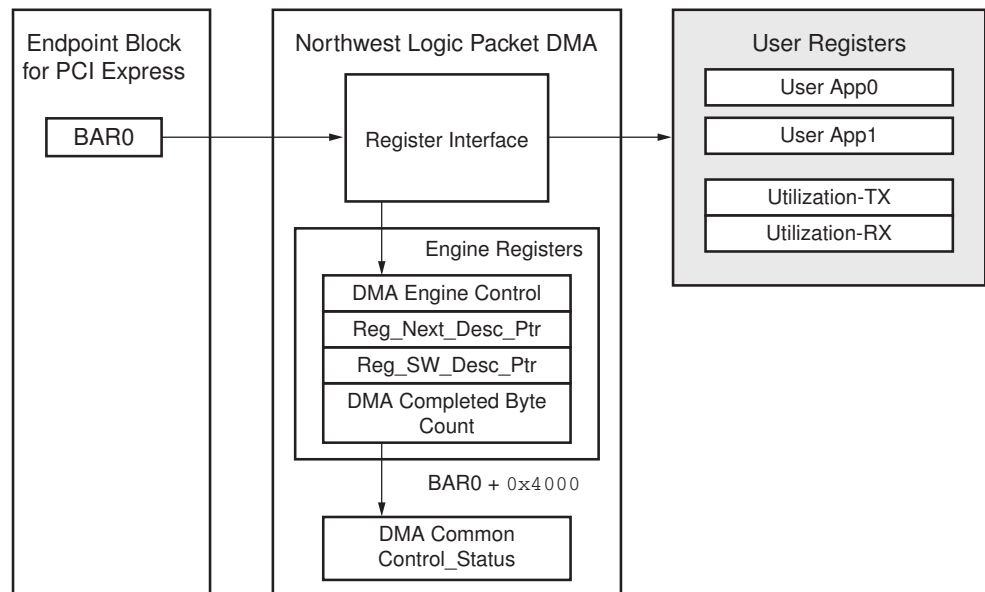
UG379_aB_01_090910

Figure B-1: **Register Map**

The user logic registers are mapped as shown in Table B-2.

Table B-2: **User Register Address Offsets**

User Logic Register Group	Range (Offset from BAR0)
Utilization Registers	`0x8200 - 0x82FF`
User App0 Registers	`0x9000 - 0x90FF`
User App1 Registers	`0x9100 - 0x91FF`

Packet DMA Registers

This section describes the prominent Packet DMA registers used frequently by the software driver. For a detailed description of all registers available, refer to the Northwest Logic *Back End Core User Guide* [Ref 9].

Packet DMA Channel-Specific Registers

The registers described in this section are present in all Packet DMA channels. The address of the register is the DMA channel address offset from BAR0 (refer to Table B-1) plus the register offset.

Engine Control (`0x0004`)

Table B-3 defines the bits within the DMA Engine Control register.

Table B-3: **DMA Engine Control Register**

Bit	Field	Mode	Default Value	Description
0	Interrupt Enable	RW	0	This bit enables interrupt generation.
1	Interrupt Active	RW1C	0	This bit is set whenever an interrupt event occurs. Write a 1 to clear this bit.
2	Descriptor Complete	RW1C	0	This bit is set when the interrupt on completion bit is set in the descriptor.
3	Descriptor Alignment Error	RW1C	0	This bit is set when the descriptor address is unaligned and that DMA operation is aborted.
4	Descriptor Fetch Error	RW1C	0	This bit is set when the descriptor fetch errors out. That is, the completion status is not successful.
5	SW_Abort_Error	RW1C	0	This bit is set when the software aborts the DMA operation.
8	DMA Enable	RW	0	When set, this bit enables the DMA engine. Once enabled, the engine compares the next descriptor pointer and software descriptor pointer to begin execution.
10	DMA_Running	RO	0	This bit indicates the DMA is in operation.
11	DMA_Waiting	RO	0	This bit indicates the DMA is waiting for software to provide more descriptors.
14	DMA_Reset_Request	RW	0	When set, this bit issues a request to user logic connected to the DMA to abort the outstanding operation and prepare for reset. This bit is cleared when the user acknowledges the reset request.
15	DMA_Reset	RW	0	When set, this bit resets the DMA engine and issues a reset to user logic.

Next Descriptor Pointer (`0x0008`)

Table B-4 defines the fields within the DMA Next Descriptor Pointer register.

Table B-4: **DMA Next Descriptor Pointer Register**

Bit	Field	Mode	Default Value	Description
[4:0]	Reserved	RO	5'b00000	This field is required for 32-byte alignment.
[31:5]	Reg_Next_Desc_Ptr	RW	0	The Next Descriptor Pointer is writable when the DMA is not enabled. It is read only when the DMA is enabled. This field should be written to initialize the start of a new DMA chain.

Software Descriptor Pointer (`0x000C`)

Table B-5 defines the fields within the DMA Software Description Pointer register.

Table B-5: **DMA Software Descriptor Pointer Register**

Bit	Field	Mode	Default Value	Description
[4:0]	Reserved	RO	5'b00000	This field is required for 32-byte alignment.
[31:5]	Reg_SW_Desc_Ptr	RW	0	The Software Descriptor Pointer contains the location of the first descriptor in the chain, which is still owned by the software.

Completed Byte Count (`0x001C`)

Table B-6 defines the fields within the DMA Completed Byte Count register.

Table B-6: **DMA Completed Byte Count Register**

Bit	Field	Mode	Default Value	Description
[1:0]	Sample Count	RO	0	This sample count is incremented every time a sample is taken at 1 second intervals.
[31:2]	DMA_Completed_Byte_Count	RO	0	The completed byte count field records the number of bytes that transferred in the previous 1 second. This field has a four-byte resolution.

Common Registers

The registers described in this section are common to all engines. These are located at the given offsets from BAR0.

Common Control and Status (`0x4000`)

Table B-7 defines the fields within the DMA Common Control and Status register.

Table B-7: **DMA Common Control and Status Register**

Bit	Field	Mode	Default Value	Description
0	Global DMA Interrupt Enable	RW	0	This bit globally enables or disables interrupts for all DMA engines.
1	Interrupt Active	RO	0	This bit reflects the state of the DMA interrupt hardware output when the state is global interrupt enable.
2	Interrupt Pending	RO	0	This bit reflects the state of the DMA interrupt output without regard to the state of the global interrupt enable.
3	Interrupt Mode	RO	0	`0`: MSI mode `1`: Legacy interrupt mode
4	User Interrupt Enable	RW	0	This bit enables generation of user interrupts.
5	User Interrupt Active	RW1C	0	This bit indicates user interrupts are active.
[23:16]	S2C Interrupt Status	RO	0	Bit [i] indicates the interrupt status of S2C DMA engine [i]. If the S2C engine is not present, this bit reads as `0`.
[31:24]	C2S Interrupt Status	RO	0	Bit [i] indicates the interrupt status of C2S DMA engine [i]. If the C2S engine is not present, this bit reads as `0`.

User Application Registers

This section describes the user application registers in detail. All registers are 32 bits wide. Bit-fields not defined are considered to be reserved, where a read always returns a value of zero.

Design Version Register

This subsection defines the register used to identify the design version being used.

Design Version (`0x8000`)

This registers allows the driver to determine the design version, the device the design is targeted at, and whether it uses AXI interfaces or non-AXI interfaces.

Table B-8: **Design Version Register**

Bit	Field	Mode	Default Value	Description
[3:0]	Sub-version number	RO	`0000`	`0000`: for designs using the PCIe TRN (Transaction) interface `0001`: for designs using the PCIe AXI4-Stream interface
[11:4]	Version number	RO	Matches the ZIP file version	Example: for v1.3 of the ZIP file, version number is `b'0001_0011`.
[27:12]	Reserved	RO	0	These bits are reserved and return zero on a READ.
[31:28]	Targeted Device	RO	`0001`	`0001` for Virtex®-6 FPGAs

Performance Monitor Registers

This subsection defines the registers implemented for measuring PCIe transaction utilization.

Transmit Utilization Byte Count (`0x8200`)

This register counts the utilization on the transmit signals of the AXI4-Stream interface of the Virtex-6 FPGA Integrated Block for PCI Express (see Table B-9). It increments every clock cycle when both s_axis_tx_tvalid and s_axis_tx_tready are asserted.

Table B-9: **Transmit Utilization Byte Count Register**

Bit	Field	Mode	Default Value	Description
[1:0]	Sample Count	RO	0	This two-bit sample count increments once every second.
[31:2]	Transmit Utilization Count	RO	0	This field contains the utilization count when the signals on the AXI4-Stream interface in the transmit direction are active. This register has a resolution of four bytes. To get the byte count, multiply the value obtained by 4 to get the byte count.

Receive Utilization Byte Count (`0x8204`)

This register counts the utilization on the receive signals of the AXI4-Stream interface of the Virtex-6 FPGA Integrated Block for PCI Express (see Table B-10). It increments every clock cycle when both m_axis_rx_tvalid and m_axis_rx_tready are asserted.

Table B-10: **Receive Utilization Byte Count**

Bit	Field	Mode	Default Value	Description
[1:0]	Sample Count	RO	0	This two-bit sample count increments once every second.
[31:2]	Receive Utilization Count	RO	0	This field contains the utilization count when the signals on the AXI4-Stream interface in the receive direction are active. This register has a resolution of four bytes. To get the byte count, multiply the value obtained by 4 to get the byte count.

Upstream Memory Write Byte Count (`0x8208`)

This register counts the payload of memory write transactions sent upstream on the transmit path of the AXI4-Stream interface of the Virtex-6 FPGA Integrated Block for PCI Express (see Table B-11).

Table B-11: **Upstream Memory Write Byte Count**

Bit	Field	Mode	Default Value	Description
[1:0]	Sample Count	RO	0	This two-bit sample count increments once every second.
[31:2]	MWR Payload Count	RO	0	This field contains the number of MWR payload bytes sent across the AXI4-Stream interface in the transmit direction. This register has a resolution of four bytes. To get the byte count, multiply the value obtained by 4.

Downstream Completion Payload Byte Count (`0x820C`)

This register counts the payload of completion transactions received at the endpoint on the receive path of the AXI4-Stream interface of the Virtex-6 FPGA Integrated Block for PCI Express.

Table B-12: **Downstream Completion Payload Byte Count**

Bit	Field	Mode	Default Value	Description
[1:0]	Sample Count	RO	0	This two-bit sample count increments once every second.
[31:2]	CplD Payload Count	RO	0	This field contains the number of CplD payload bytes received across the AXI4-Stream interface in the receive direction. This register has a resolution of four bytes. To get the byte count, multiply the value obtained by 4.

Initial Flow Control Credits for Completion Data for the PCIe Downstream Port (`0x8210`)

This register reports the initial flow control credits of the host system (see Table B-13).

Table B-13: **Initial Flow Control Credits for Completion Data for the Host System**

Bit	Field	Mode	Default Value	Description
[11:0]	INIT_FC_CPLD	RO	0	After link training, the host system advertises its initial flow control credits. The flow control credits for completion data is captured in this register

Initial Flow Control Credits for Completion Header for the PCIe Downstream Port (`0x8214`)

This register reports the initial flow control credits of the host system (see Table B-14).

Table B-14: **Initial Flow Control Credits for Completion Header for the Host System**

Bit	Field	Mode	Default Value	Description
[7:0]	INIT_FC_CPLH	RO	0	After link training, the host system advertises its initial flow control credits. The flow control credits for completion header is captured in this register.

Initial Flow Control Credits for Non-Posted Data for the PCIe Downstream Port (`0x8218`)

This register reports the initial flow control credits of the host system (see Table B-15).

Table B-15: **Initial Flow Control Credits for Non-Posted Data for the Host System**

Bit	Field	Mode	Default Value	Description
[11:0]	INIT_FC_NPD	RO	0	After link training, the host system advertises its initial flow control credits. The flow control credits for non-posted data is captured in this register.

Initial Flow Control Credits for Completion Non-Posted Header for the PCIe Downstream Port (`0x821C`)

This register reports the initial flow control credits of the host system (see Table B-16).

Table B-16: **Initial Flow Control Credits for Non Posted Header for the Host System**

Bit	Field	Mode	Default Value	Description
[7:0]	INIT_FC_NPH	RO	0	After link training, the host system advertises its initial flow control credits. The flow control credits for non-posted header is captured in this register.

Initial Flow Control Credits for Posted Data for the PCIe Downstream Port (`0x8220`)

This register reports the initial flow control credits of the host system (see Table B-17).

Table B-17: **Initial Flow Control Credits for Posted Data for the Host System**

Bit	Field	Mode	Default Value	Description
[11:0]	INIT_FC_PD	RO	0	After link training, the host system advertises its initial flow control credits. The flow control credits for posted data is captured in this register.

Initial Flow Control Credits for Posted Header for the PCIe Downstream Port (`0x8224`)

This register reports the initial flow control credits of the host system.

Table B-18: **Initial Flow Control Credits for Posted Header for the Host System**

Bit	Field	Mode	Default Value	Description
[7:0]	INIT_FC_PH	RO	0	After link training, the host system advertises its initial flow control credits. The flow control credits for posted header is captured in this register.

User App0 Registers

This section defines the registers specific to the XAUI application connected to DMA channel 0.

XAUI Error (`0x9000`)

This register indicates fatal and non-fatal errors that might happen on the XAUI path (see Table B-19). The software needs to reset the DMA in case of fatal errors because the TRD cannot recover after the error has occurred. The register clears when the software reads it.

Table B-19: **XAUI Error**

Bit	Field	Mode	Default Value	Description
0	TX ERR - Fatal	RO	0	If XGMII alignment logic reads from an empty virtual FIFO or reads less than eight bytes on a clock, then the current packet on the XAUI path is corrupted, and this bit is set.
1	Packet Length ERR - Fatal	RO	0	This bit is set if the length field of the XAUI packet does not match the actual payload. This bit is also set when SOP on a packet is set and EOP is not set. The length field includes payload size + four bytes for CRC. Thus this check is based on the length field – four bytes.
2	Dropped Packet - Non-Fatal	RO	0	If the header CRC does not check out on a packet, packet segments might have been dropped. This can happen if there is congestion on the path.

XAUI IFG (`0x9004`)

This register value increases the inter-frame gap (IFG) between consecutive XAUI transmit packets by inserting Idles on the lanes (see Table B-20). The software can program this value if there is a lot of congestion and packets are dropped.

Table B-20: **XAUI IFG**

Bit	Field	Mode	Default Value	Description
[15:0]	Inter-Frame Gap	RW	0	This field determines the number of Idle cycles to be inserted between consecutive XAUI transmit packets.

XAUI Config (`0x9008`)

This register is tied to the configuration vector bits on the XAUI LogiCORE™ block (see Table B-21). The TRD uses only bit 0 of this register. For more details on the configuration vector bits, refer to the XAUI LogiCORE IP block [Ref 10].

Table B-21: **XAUI Config Register**

Bit	Field	Mode	Default Value	Description
0	Loopback	RW	0	This bit sets the serial loopback in the device-specific transceivers

XAUI Status (`0x900C`)

This register is tied to status vector bits on the XAUI LogiCORE block. This register provides information on receiver alignment, link status, and errors on the XAUI transmit and receive paths. For more details on the status vector bits, refer to the XAUI LogiCORE IP block [Ref 10].

User App1 Registers

This section defines the register specific to the Raw Data application connected to DMA channel 1.

Enable Generator (`0x9100`)

This register is used to enable the data generator in the loopback static module (see Table B-22). The generator allows receive operations to run independent of transmit operations on the Raw Data path.

Table B-22: **Enable Generator Register**

Bit	Field	Mode	Default Value	Description
1	Enable Generator	RW	0	This bit enables the generator logic on the Raw Data path. The logic generates data for the Raw data receive path. If the Enable Generator bit is set to 1, the Enable Loopback bit must be 0.

Packet Length (`0x9104`)

This register configures the fixed length of packets on the Raw Data path (see Table B-23).

Table B-23: **Raw Data Packet Length Register**

Bit	Field	Mode	Default Value	Description
[15:0]	Packet Length	RW	d'768	This field contains the length of packets on the Raw Data path.

Enable Checker or Loopback (`0x9108`)

This register enables the data checker or enable loopback mode on the receive path (see Table B-24). If loopback is enabled, the transmit data is not verified by the data checker.

Table B-24: **Enable Checker or Loopback Register**

Bit	Field	Mode	Default Value	Description
0	Enable Checker	RW	0	This bit enables the checker logic on the Raw Data path. The logic checks the data transmitted by the host. It reports any data mismatches. If the Enable Checker bit is set to 1, the Enable Loopback bit must be 0.
1	Enable Loopback	RW	0	If this bit is set, the data transmitted by the host is looped back and sent out on the receive path. If this bit is set to 1, the Enable Checker and the Enable Generator bits must be 0.

Data Mismatch (`0x910C`)

This register reports data integrity failures on the Transmit path of the Raw Data path.

Table B-25: **Data Mismatch on Raw Data Transmit Register**

Bit	Field	Mode	Default Value	Description
0	Data Mismatch	RO	0	If the data checker on the Raw Data path finds a mismatch between the expected data and data transmitted by the host, it sets the Data Mismatch flag. This bit is cleared when Enable Checker is set to 0.