Chain FPGA IP Catalog

Justen G. Di Ruscio

August 15, 2019

Abstract

This document contains operational descriptions for the intellectual property created by Justen Di Ruscio in the Summer of 2019 for use in Jay-1 ionosonde. However, this document is not a complete list of Justen's tasks during this time. All intellectual property listed is packaged hardware defining code, Verilog or SystemVerilog, to be implemented on a field programmable gate array. The designs illustrated are components of the Jay-1 ionosonde created by the Canadian High Arctic Ionospheric Network.

Relevant Information

Throughout this catalog, *IP* will be used as the acronym for *intellectual property*, and *IF* will refer to *interface*. In addition, *FPGA* describes a *field programmable gate array*, *BRAM* stands for *Block Random Access Memory*, and *AXI* refers to the *Advanced eXtensible Interface*, of which, AXI-Lite and AXI-Stream are used. For all listed IP, at least one test bench and post-synthesis test has been completed. Vivado was the development tool used for all components ,and the Genesys2 development board, using a Kintex-7 FPGA, is the target device.

Contents

1 Signal Paths **4**
- 1.1 Input Signal Path . . . 4
- 1.2 Output Signal Path . . . 4

2 power_meter **5**
- 2.1 Diagram . . . 5
- 2.2 Parameters . . . 5
- 2.3 Pin Description: . . . 5
- 2.4 AXI Register Mapping . . . 6
- 2.5 AXI Register Description . . . 6
- 2.6 Usage . . . 7

3 IDELAYBUFDS **7**
- 3.1 Diagram . . . 7
- 3.2 Parameters . . . 7
- 3.3 Pin Description . . . 8
- 3.4 Usage . . . 8

4 custom_IBUFDS **9**
- 4.1 Diagram . . . 9
- 4.2 Parameters . . . 9
- 4.3 Pin Description . . . 10
- 4.4 Usage . . . 10

5 delay_FIFO **10**
- 5.1 Diagrams . . . 11
- 5.2 Parameters . . . 11
- 5.3 Pin Description . . . 13
- 5.4 AXI Register Mapping . . . 15
- 5.5 AXI Register Description . . . 15
- 5.6 Usage . . . 17

6 adc_IF_core **18**
- 6.1 Contained IP . . . 19
- 6.2 Diagrams . . . 20
- 6.3 Parameters . . . 20
- 6.4 Pin Description . . . 21
- 6.5 AXI Interface Mapping . . . 23
- 6.6 Usage . . . 24

7 AXI_ADC_ctrl **24**
7.1 Diagram . . . 25
7.2 Parameters . . . 25
7.3 Pin Description . . . 25
7.4 AXI Register Mapping . . . 26
7.5 AXI Register Description . . . 26
7.6 Usage . . . 26

8 AXI_DAC_CTRL **27**
8.1 Diagram . . . 27
8.2 Parameters . . . 27
8.3 Pin Description . . . 28
8.4 AXI Register Mapping . . . 29
8.5 AXI Register Description . . . 29
8.6 Usage . . . 29

9 shift_register **30**
9.1 Diagram . . . 30
9.2 Parameters . . . 31
9.3 Pin Description . . . 31
9.4 Operation . . . 32

10 PRBS_gen **33**
10.1 Diagram . . . 33
10.2 Parameters . . . 33
10.3 Pin Description . . . 34
10.4 Usage . . . 34

11 channel_delay **34**
11.1 Diagram . . . 35
11.2 Parameters . . . 35
11.3 Pin Description . . . 35
11.4 Usage . . . 36

12 dac_IF_core **36**
12.1 Contained IP . . . 37
12.2 Diagrams . . . 38
12.3 Parameters . . . 38
12.4 Pin Description . . . 39
12.5 AXI Interface Mapping . . . 41

12.6 Usage 42

13 VGA_SW_IF **42**
13.1 LMH6517 Block Diagram 43
13.2 Diagram 43
13.3 Parameters 43
13.4 Pin Description 44
13.5 AXI Register Mapping 45
13.6 AXI Register Description 45
13.7 Usage and Operation 46
13.8 Pulsed Interface 47

14 References and Links **48**

List of Tables

1 power_meter AXI-Lite Register Map 6
2 delay_FIFO AXI Register Map 15
4 Data Channel Mapping - axis_tdata 23
5 adc_IF_core AXI-Lite Address Mapping 24
6 AXI_ADC_ctrl Register Map 26
7 Control Line Mapping for Each ADC 26
8 AXI-Lite Register Mapping for AXI_DAC_CTRL 29
10 Data Channel Mapping - axis_tdata 41
11 dac_IF_core AXI-Lite Address Mapping 41
12 VGA_SW_IF Register Map 45

1 Signal Paths

To depict the purpose and use of the IP within the Jay-1 design, the input and output signal paths are shown below. It's important to understand the purpose and use of the blocks interfacing with the FPGA, which will be described in this document, but not the exact specifications of each block.

1.1 Input Signal Path

This is the path a received signal must travel, for a single input channel. Of

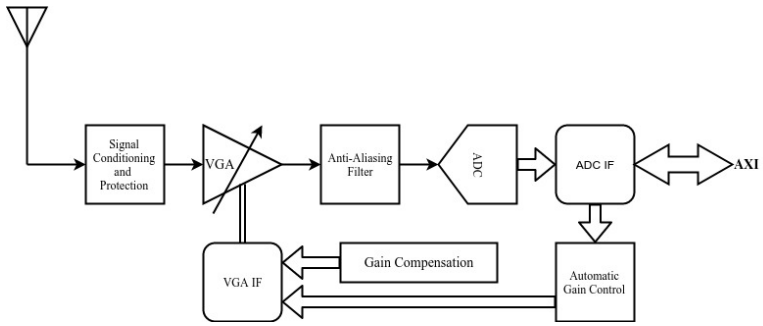


Figure 1: Input Signal Path Block Diagram

the above blocks, the VGA IF and ADC IF blocks, and all their contained IP, are components within this document.

1.2 Output Signal Path

This is the path a transmitted signal must travel, for a single output channel. Of the above blocks, the DAC IF, and all its contained IP, are components

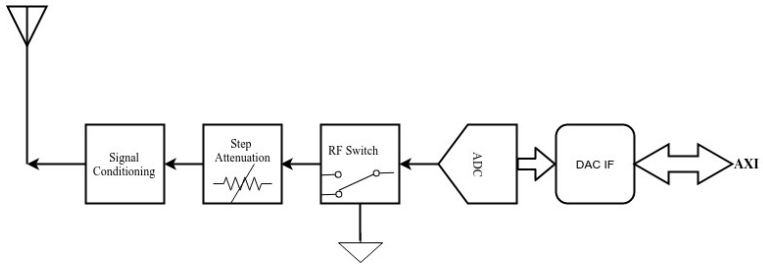


Figure 2: Output Signal Path Block Diagram

within this document.

2 power_meter

Although the Power Meter was specifically created to measure the input power of of the ADC, located within the ADC IF, it is flexible and parameterized, allowing it to measure any binary value. As as result, it is also implemented in the DAC_ IF. Measurements by the power_meter are done with respect to full scale input range, causing the measurements to actually report signal magnitude, but this extends to signal power measurements based on the signal termination used.

2.1 Diagram

Figure 3: IP Integrator Diagram of the power_meter

2.2 Parameters

Non-AXI related parameters:

sampleSize Number of bits composing samples. Scales the channel size of dataIn and aveVal. Should be between 1-16bits for current design.

2.3 Pin Description:

dataIn Parallel input channel for data

dataClk Clock used to govern rate at which data is sampled from dataIn

aveVal Parallel output channel containing the sample average over the specified integration window

s00_axi_aclk Positive edge sensitive clock governing data rate of AXI-Lite interface

s00_axi_aresetn Active low, asynchronous reset for AXI-Lite interface and power_meter logic

s00_axi Channel containing pins for the AXI-Lite specification. This IP acts as a slave

2.4 AXI Register Mapping

Table 1: power_meter AXI-Lite Register Map

<table>
<tr><th>Register Name</th><th>Address Offset</th><th colspan="3">Register Map (32bits per reg.)</th></tr>
<tr><td rowspan="2">slv_reg0</td><td rowspan="2">4'h0</td><td>Unused</td><td>Trigger</td><td>integrationWindowPower</td></tr>
<tr><td>bits 31-5</td><td>bit 4</td><td>bits 3-0</td></tr>
<tr><td rowspan="2">slv_reg1</td><td rowspan="2">4'h4</td><td colspan="2">Unused</td><td>calcDone</td></tr>
<tr><td colspan="2">bits 31-1</td><td>bit 0</td></tr>
<tr><td rowspan="2">slv_reg2</td><td rowspan="2">4'h8</td><td colspan="2">peakADCVal</td><td>aveADCVal</td></tr>
<tr><td colspan="2">bits 31-16</td><td>bits 15-0</td></tr>
<tr><td rowspan="2">slv_reg3</td><td rowspan="2">4'hC</td><td colspan="3">Unused</td></tr>
<tr><td colspan="3">bits 31-0</td></tr>
</table>

2.5 AXI Register Description

trigger (w) Active high flag to start the sampling cycle of the power_meter

integrationWindowPower (w) Specifies the size of the integration window as $2^{integrationWindowPower}$

calcDone (r) Active high flag indicating when the power_meter has completed its calculations

peakADCVal (r) Largest sample recorded on dataIn over integration window

aveADCVal (r) Average sample value over the integration window

2.6 Usage

A reset pulse should be delivered at boot up to initialize all registers in the device. Once reset is complete, the device will wait for the trigger bit to be asserted, where the power_meter will read the integration window power from the respective AXI register to calculate the integration window size. A valid trigger is a positive edge of the trigger bit while the device is not currently sampling data, indicated by the calcDone flag. Only at a valid trigger will the integration window power be sampled; however, both registers can be written to at any time. This property is useful for clearing the trigger bit while the Power Meter is busy.
On dataClk rising edges following a valid trigger, the calcDone bit will be asserted low and the Power Meter will sample the dataIn channel. After the integration window number of samples have been acquired, the calcDone flag will be asserted high, and the average and peak sample will be written to their respective AXI register and average output pins. At this point, the power meter can repeat its operation cycle.

3 IDELAYBUFDS

The IDELAYBUFDS contains a configurable number of IDELAYE2 and IBUFDS Xilinx primitive pairs. Information regarding these internal primitives can be found in the *UG953* reference guide, also by Xilinx. In this IP, the IDELAYE2 is used in fixed delay mode, placed after the IBUFDS.

3.1 Diagram

Figure 4: IP Integrator Diagram of the IDELAYBUFDS

3.2 Parameters

ioStandard String describing the I/O standard used in the IBUFDS. The default option is “LVDS_25”, but other options can be found in DS182

- the Kintex-7 datasheet linked below

dataWidth Number of IDELAYE2/IBUFDS combination channels to create in IP

lowPower_buf Enables or disables low power mode of the IBUFDS in exchange for performance. Available options are "TRUE" and "FALSE"

highPerformance_delay Enables or disables high performance mode of the IDELAYE2 in exchange for power savings. Available options are "TRUE" and "FALSE"

refClkFreq The frequency in MHz of the reference clock, provided through an IDELAYCTRL module. Available value are 190-210 MHz and 290-310 MHz for the 7-Series FPGAs

diffTermination Enables or disables the input differential termination resistance between the input terminals. Available options are "TRUE" and "FALSE"

delayTapVal Selectable tap value which selects the delay imposed upon the signal, equal to $1/(64 \times refClkFreq)\mu S$, where refClkFreq is in MHz

signalType Describes the signal type for the IDELAYE2. Available values are "CLOCK" and "DATA", and the default value is "DATA". This allows the timing analyzer to account for jitter in the delay-chain

3.3 Pin Description

data_in_p Positive input channal for all IBUFDS/IDELAYE2 pairs

data_in_n Negative input channal for all IBUFDS/IDELAYE2 pairs

data_out Single ended output channel for all IBUFDS/IDELAYE2 pairs

3.4 Usage

As this IP includes an IDELAYE2, it requires an IDELAYCTRL device to be instantiated in the same design in order to maintain calibration. Although there is not a physical wire routed between the two, there is an internal bus designated for the reference clock signals in the FPGA, and once both are instantiated, they will be placed on this bus upon implementation.
Each line in the input or output channel corresponds to the input or output

with the same address. For example data_in_p[2], data_in_n[2], and data _out[2] are the pins to the same IBUFDS/IDELAYE2 pair.
The inputs should be routed to physical input pins on the FPGA chip.

4 custom_IBUFDS

The custom_IBUFDS is simply an extensible wrapper around the IBUFDS primitive, from Xilinx. This not only allows multiple differential channels within a single IP, but also extends the internal IBUFDS's functionality to cooperate with data signals instead of just clock signals. Further information regarding the internal IBUFDS primitive from Xilinx can be found in the *UG953* reference guide.

4.1 Diagram

Figure 5: IP Integrator Diagram of the custom_IBUFDS

4.2 Parameters

diffTerm Enables or disables the input differential termination resistance between the inputs terminals. Available options are "TRUE" and "FALSE". The default is "FALSE"

lowPower Enables or disables low power mode of the IBUFDS in exchange for performance. Available options are "TRUE" and "FALSE". The default is "FALSE"

ioStandard String describing the I/O standard used in the IBUFDS. The default option is "LVDS_25", but other options can be found in DS182 - the Kintex-7 datasheet linked below

size Number of differential channels to instantiate internally

4.3 Pin Description

in_p Positive data inputs for all channels

in_n Negative data inputs for all channels

bufout Single ended outputs for all channels

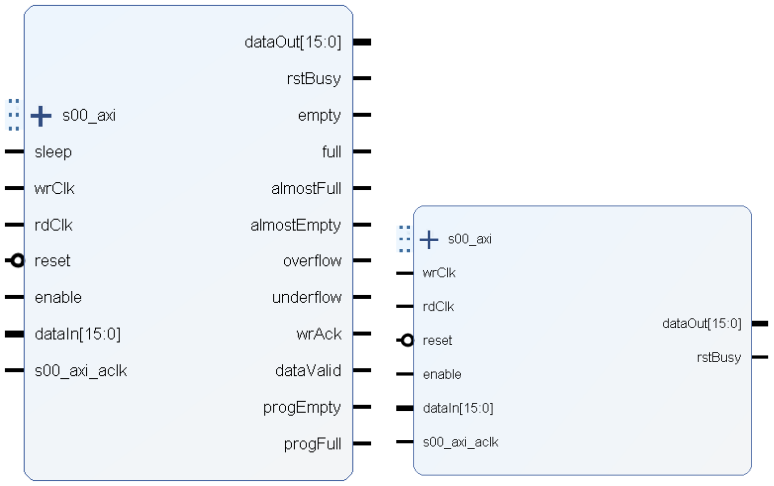
4.4 Usage

Compatible with data and clock signals. Ensure the I/O standard is set to meet the differential I/O standard driving the input lines. The inputs should be routed to physical input pins on the FPGA chip.

5 delay_FIFO

The delay_FIFO is essentially a hardware BRAM asynchronous FIFO which prevents reading until the specified number of words have been written. This delays the digital values by storing them internally.
The underlying FIFO is an xpm_fifo_async, which is a macro provided by Xilinx that instantiates the necessary BRAM elements, based on the parameters specified. These BRAM elements will be connected together to meet the size requirement, up to 150 Mbits. Most of the tight timing restrictions and operating conditions inherent to BRAM devices, including the xpm_fifo_async, are managed by the delay_FIFO wrapper. Status flags are also available on this device which fire under certain states, referred to as 'protection pins', as they can protect the device from misuse. All protection pins are accessible through the available AXI-Lite interface and are optionally enabled as discrete hardware lines. The IP is completely extensible and parameterized. As such, it can be configured to fit most applications of varying requirements. Refer the *UG953* reference guide for timing diagrams of the Xilinx macro in specific situations.

5.1 Diagrams



(a) All Pins Enabled (b) Default Settings

Figure 6: IP Integrator Diagram of the delay_FIFO

5.2 Parameters

Non-AXI related parameters:

fifoWidth Width, from 1 to 4096, of FIFO. This scales input and output channel sizes and the number of instantiated BRAM elements

fifoDepth Depth, from 16 to 4194304, of FIFO. This scales the maximum available delay value and number of instantiated BRAM elements

resetVal Hexadecimal string representing the value on the output channel during reset. Default is "0"

addrWidth Number of bits used to address elements within BRAM. Automatically calculated and is not available in the graphical tool, but may be overriden in RTL if required

readMode Selects the read mode. Available options are "std", which is the default, or "fwft", which selects first word fall through operation.

rd_clk_delay _en Single bit used to enable the read clock for delay timing instead of the write clock, which is the default. Available options are 1'b0, the default, or 1'b1

prog_full_empty_val Number of words away from completely full or empty at which the prog_full or prog_empty flags are respectively asserted. Only used if prog_full_empty_en is set to "1".

advancedFeatures Hexadecimal string representing a 13-bit word used to enable features of the hardware FIFO. The default value is "1B1B" and each bit within the word is active high. Locations of the features within the 13-bit configuration word is shown below:

- **bit 0** overflow flag
- **bit 1** prog_full flag (programmable full)
- **bit 3** almost_full flag
- **bit 4** wr_ack flag (write acknowledge)
- **bit 8** underflow flag
- **bit 9** prog_empty flag (programmable empty)
- **bit 11** almost_empty flag
- **bit 12** data_valid flag
  Note: Unlisted bits are unused and should be set to 0.

As an example, if a user only wants the overflow flag enabled in hardware, they should set this parameter to "1"

almost_full_empty_en Single bit enabling or disabling the external almostEmpty and almostFull pins

prog_full_empty_en Single bit enabling or disabling the external progEmpty and progFull pins

over_under_flow_en Single bit enabling or disabling the external underflow and overflow pins

wr_ack_en Single bit enabling or disabling the external wrAck (write acknowledge) pin

data_valid_en Single bit enabling or disabling the external dataValid pin

full_empty_en Single bit enabling or disabling the external empty and full pins

sleep_en Single bit enabling or disabling the external sleep input pin

5.3 Pin Description

sleep When either this line, or the sleep bit in the AXI register, go high, the FIFO enters a dynamic power saving mode

wrClk Write clock: rising edges of this line clock data into the FIFO. This is also the default delay timing clock when the rd_clk_delay_en parameter is set to 1'b1

rdClk Read clock: rising edges of this line clock data out of the FIFO once the specified number of words have been written into the FIFO as delay. This can also be used as the delay timing clock when the rd_clk_delay_en parameter is set to 1'b1

reset Active low reset. A negative edge of this line, or the negative edge of the reset bit in the AXI register, will cause the FIFO to enters its reset sequence, indicated by the rstBusy line.

enable Active high enable pin. When this pin, or the enable bit in the AXI register, go high, the FIFO is enabled, allowing writes and reads. However, the device will only be enabled when: the device is not currently in its reset sequence (indicated by the rstBusy line); both reset inputs are inactive; and when the disable bit in the AXI register is low. All of these take precedence over an asserted enable line

dataIn Input channel for data. So long as the device is enabled, the value on these lines will be written into the FIFO upon a rising wrClk (write clock) edge

dataOut Output channel for data. So long as the device is enabled, and the specified delay has been met, the eldest word written into the FIFO will be extracted and written to these lines.

rstBusy Reset busy: internally asserted high while the device is within its reset sequence

empty Asserted high when the FIFO is empty. A read from the FIFO while its empty will be ignored and is not destructive. This pin can be removed from the package using the full_empty_en parameter. Set until read flag

full Asserted high when the FIFO is full. A write to the FIFO while its full will be ignored and is not destructive. This pin can be removed from the package using the full_empty_en parameter. Set until read flag

almostFull Asserted high when the FIFO is one word away from being full. This pin can be removed from the package using the almost_full_empty_en parameter. Set until read flag

almostEmpty Asserted high when the FIFO is one word away from being empty (contains only a single word). This pin can be removed from the package using the almost_full_empty_en parameter. Set until read flag

overflow Asserted high on the following wrClk edge after the FIFO becomes full and the device is enabled. The overflow will be ignored and will not be destructive. This pin can be removed from the package using the over_under_flow_en parameter. Set until read flag

underflow Asserted high on the following rdClk edge after the FIFO becomes empty and the device is enabled. The underflow will be ignored and will not be destructive. This pin can be removed from the package using the over_under_flow_en parameter. Set until read flag

wrAck Write acknowledge: set high for a clock cycle indicating the write to the FIFO on the previous write clock edge was accepted. This pin can be removed from the package using the wr_ack_en parameter. Set until read flag

dataValid Flag asserted high while there is valid data on the output port, and low otherwise. This pin can be removed from the package using the data_valid_en parameter. Set until read flag

progEmpty Programmable empty: flag set high when FIFO reaches programmable empty threshold. This pin can be removed from the package using the prog_full_empty_en parameter. Set until read flag

progFull Programmable full: flag set high when FIFO reaches programmable full threshold. This pin can be removed from the package using the prog_full_empty_en parameter. Set until read flag

s00_axi Bus containing pins for AXI-Lite specification. This IP acts as a slave

5.4 AXI Register Mapping

Table 2: delay_FIFO AXI Register Map

Register Name	**Address Offset**	**Register Map (32bits per reg.)**						
slv_reg0	4'h0	Unused						...
		bits 31-26						...
slv_reg1	4'h4	wrAck	dataValid	overflow	full	almostFull	progFull	...
		bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	...
slv_reg2	4'h8	Unused						...
		bits 31-2						...
slv_reg3	4'hC	Unused						...
		bits 31-0						...

Table 2 Continued

	Register Map (32bits per reg.)						
...	disable	sleep	reset	enable	delayValue		
	bit 25	bit 24	bit 23	bit 22	bits 21-0		
...	progEmpty	almostEmpty	empty	underflow	fifoDepth		
	bit 25	bit 24	bit 23	bit 22	bits 21-0		
...	Unused				rstBusy	wrRstBusy	rdRstBusy
	bits 31-2				bit 2	bit 1	bit 0
...	Unused						
	bits 31-0						

5.5 AXI Register Description

disable (w) Overrides both of the enable inputs to disable the device. While disabled, both reads and writes are prevented

sleep (w) When either this register, or the dedicated sleep line, go high, the FIFO enters a dynamic power saving mode

reset (w) Active low reset. A negative edge of this register, or the negative edge of dedicated hardware line, will cause the FIFO to enters its reset sequence, indicated by the rstBusy line.

enable(w) Active high enable register. When this register, or the dedicated hardware line, go high, the FIFO is enabled, allowing writes and reads. However, the device will only be enabled when: the device is not currently in its reset sequence (indicated by the rstBusy line); when both reset inputs are inactive; and when the disable bit in the AXI register is low. All of these take precedence over an asserted enable line

delayValue (w) Sets the delayValue, the number of timer clock edges, chosen as the read clock or the write clock, to delay data through FIFO. This value can be set between three up to the fifoDepth; however, values above or below these thresholds will result in the limits being used, preventing any harm. At reset, three is written to this register, allowing operation with the minimum delay without having to first having to write to any AXI register. Moreover, delayValue can be modified at runtime, but it should be noted that increasing and decreasing the delayValue at runtime each have their own operation

wrAck (r) Write acknowledge: set high for a clock cycle indicating the write to the FIFO on the previous write clock edge was accepted. Set until read flag

dataValid (r) Flag asserted high while there is valid data on the output port, and low otherwise. Set until read flag

overflow (r) Asserted high on the following wrClk edge after the FIFO becomes full and the device is enabled. The overflow will be ignored and will not be destructive. Set until read flag

full (r) Asserted high when the FIFO is full. A write to the FIFO while its full will be ignored and is not destructive. Set until read flag

almostFull (r) Asserted high when the FIFO is one word away from being full. Set until read flag

progFull (r) Programmable full: flag set high when FIFO reaches programmable full threshold. Set until read flag

progEmpty (r) Programmable empty: flag set high when FIFO reaches programmable empty threshold. Set until read flag

almostEmpty (r) Asserted high when the FIFO is one word away from being empty (contains only a single word). Set until read flag

empty (r) Asserted high when the FIFO is empty. A read from the FIFO while its empty will be ignored and is not destructive. Set until read flag

underflow (r) Asserted high on the following rdClk edge after the FIFO becomes empty and the device is enabled. The underflow will be ignored and will not be destructive. Set until read flag

fifoDepth (r) Depth of the FIFO set in the parameter. Makes this available at runtime if needed

rstBusy (r) Active high flag indicating that the FIFO is still in its reset sequence

wrRstBusy (r) Active high flag indicating that the FIFO's write (input) port is specifically within a reset state in the overall reset sequence

rdRstBusy (r) Active high flag indicating that the FIFO's read (input) port is specifically within a reset state in the overall reset sequence

5.6 Usage

This IP requires an initial reset pulse for proper operation. Once the rstBusy line indicates that the reset sequence has completed, the FIFO's enable inputs become active; nevertheless, they can be driven to any state at any point without consequences. The FIFO can become enabled when either of the enable lines are high, both of the reset lines are inactive (high), the FIFO is not currently in a reset procedure (indicated by the rstBusy line being low), and the disable bit in the AXI register is low. Once enabled, the FIFO will start collecting samples at each rising wrClk edge. After the specified number of timing clock edges, which can be either the rdClk or wrClk, reading from the FIFO will be permitted at every following rdClk rising edge.
The delayValue bits within the AXI reg are used to select the number of timing clock edges used delay the input data. This delayValue can be three, the default value, up to the fifoDepth specified as a parameter. Changing the delayValue at runtime is permitted. Increasing the delayValue will cause the output data to remain fixed, and the dataValid flag to go low until the larger delayValue has been met. So long as there is not an overflow, no input data will be lost during this procedure. Decreasing the delayValue will cause the wrAck flag to be asserted low until the smaller delayValue has been met. To meet a smaller delay value, input samples must be ignored,

resulting in a small loss of data. Assuming a constant flow of input data, the samples following the write of the updated delayValue, to the samples before the assertion of wrAck, are ignored. The difference between the current delayValue and the smalleer delayValue is the number of samples that get ignored following a write of an updated delayValue.
At any point, the flags listed in Pin Description and AXI Register Description will fire to indicate state and possibly protect the FIFO from application specific errors. The flags which indicate they are *set until read* flags will be set at their associated state but will only be cleared upon a read of their respective register. These inform the user that the flag has been set since the last read of the register.
For optimal operation, the write clock and read clock should be of the same frequency with no preference of phase shift. This is not enforced, nor is it required.

6 adc_IF_core

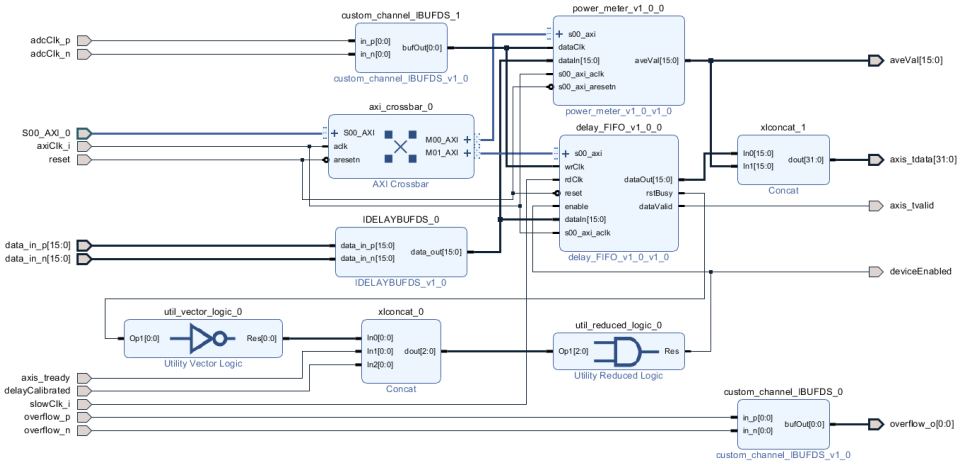
The adc_IF_core is an IP to specifically interface with the LTC2217 analog to digital converter within the input signal path of the Jay-1 Ionosonde. It reads the 16-bit LVDS 2.5 V parallel data from the *ADC*, acts as an AXI-Stream master to stream the samples to a slave, and provides an AXI-Lite interface for configuration.
The input samples have two locations for delay tuning, one in the IDELAYBUFDS for phase matching with the clock, and another in the delay_FIFO for phase matching between adc_IF_core devices. The AXI-Stream data channel is generated by the output of the delay_FIFO. In addition, a power_meter is contained within this IP to read the ADC power, directly after the IDELAYBUFDS delay element. All AXI-Lite registers of the power_meter and the delay_FIFO are available through a single AXI-Lite slave interface. Internally, this IP contains the logic to prevent data corruption from enablement or reset while the IP is busy, and logic to meet reset requirements of contained IP.
Finally, the IP converts the LVDS overflow signal from the ADC to a single-ended signal for other IP to utilize. For further information on the LTC2217, please refer to *LT0108*.

6.1 Contained IP

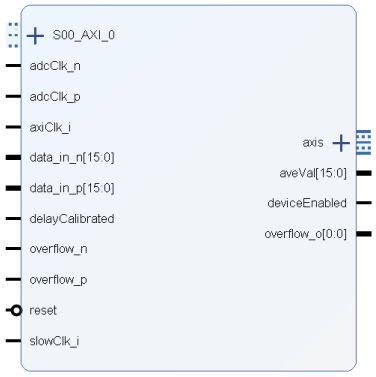
Table 3: Contained IP Within adc_IF_core

Custom IP	**Xilinx IP**
• custom_channel_IBUFDS • power_meter • delay_FIFO • IDELAYBUFDS	• axi_crossbar • util_reduced_logic • util_vector_logic • xlconcat • xlconstant

6.2 Diagrams



(a) Internal Diagram



(b) adc_IF_core Package

Figure 7: IP Integrator Diagrams of the adc_IF_core

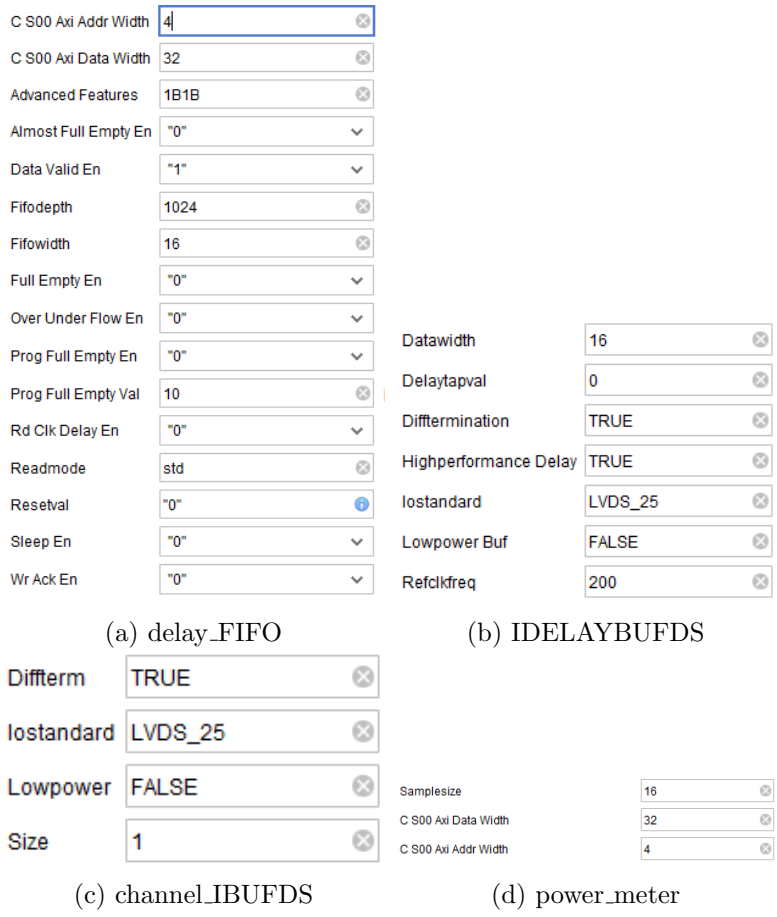
6.3 Parameters

Since this IP is very specific to its application, it does not contain any global parameters. All internal IP is pre-configured to meet the application requirements. Sadly, the IP flow does not support passing global parameters to internal IP, preventing externalizing composing IP's parameters. The delayTapVal parameter of the IDELAYBUFDS suffers most from this, as it may need modification to match the adcClk phase. If one would like to alter the default parameters, they can do so by modifying the design in the IP

Integrator and re-packaging.

6.3.1 Internal IP Default Parameters

Every IP has default parameters, most of which are obvious or are described by the IP operation. The important parameters are listed in the following figures.



(a) delay_FIFO (b) IDELAYBUFDS

(c) channel_IBUFDS (d) power_meter

Figure 8: Default Parameters of the adc_IF_core's Internal IP

6.4 Pin Description

adcClk_n Negative input terminal for LVDS 2.5 V data rate clock from ADC

adcClk_p Positive input terminal for LVDS 2.5 V data rate clock from ADC

axiClk_i Single-ended input terminal for AXI-Lite

data_in_n Negative input channel for LVDS 2.5 V data from ADC

data_in_p Positive input channel for LVDS 2.5 V data from ADC

delayCalibrated Flag from IDELAYCTRL module that must be instantiated within the same design as this IP. Informs the adc_IF _core that the delay filter is stable/calibrated. This IP is internally prohibited from reading or writing data to the AXI-Stream interface through the FIFO while this is low. The differential overflow buffer and power_meter may still run while this line is low, but that is not recommended.

overflow_n Negative input terminal for LVDS 2.5 V overflow signal from LTC2217 ADC

overflow_p Positive input terminal for LVDS 2.5 V overflow signal from LTC2217 ADC

overflow_o Single-ended output terminal for overflow signal from LTC2217 ADC

slowClk_i Single ended input for internally generated 'slow clock', which is a 105MHz clock that extracts data from the adc_IF_core through the AXI-Stream interface. This is essentially the AXI-Stream data clock and, as such, should be connected to the AXI-Stream input clock of the connected slave. The slow clock frequency should be that of the ADC sampling frequency, and should be connected to all adc_IF_cores, so they all extract data at the same time.

aveVal 16-bit, single-ended output channel from the internal power_meter, relaying the average power for other hardware devices

reset Active low reset for all device logic, including AXI-Lite interface

deviceEnabled Status flag indicating the internal FIFO is enabled; this IP is able to receive samples. Specifically, this flag indicates:

1. delayCalibrated is high

2. AXI-Stream interface is ready, indicated by a logic 1 on the ready pin
3. the FIFO is not within its reset procedure
4. active low reset line is deactivated

axis AXI-Stream interface. Contains necessary lines to match protocol. As these lines are simple enough to be implemented in a design with discrete logic, their purposes are listed below:

- **axis_tvalid** Indicates to the slave that valid data is available
- **axis_tready** Indicates to the master that the slave is ready for data
- **axis_tdata** 32-bit data channel

A slave must only implement the axis_tready signal and read the valid and data lines.

S00_AXI_0 AXI-Lite bus containing all necessary lines for the protocol

6.5 AXI Interface Mapping

Both the AXI-Stream data channel mapping and the AXI-Lite address mapping are included in this section. For a detailed mapping of each register, please refer to the respective IP description above.

6.5.1 AXI-Stream Interface

Table 4: Data Channel Mapping - axis_tdata

Channel: bits 31 - 0	
aveVal	dataOut
bits 31 - 16	bits 15-0
average value from power meter	data output of device containing ADC samples

Only dataOut, the bottom 16-bits of axis_tdata, is required for the Jay-1 ionosonde. Since there were 16 extra bits, the aveVal output channel from the power_meter is also routed to this data interface. It doesn't have to be used, and is simply there for extendability.

6.5.2 AXI-Lite Address Mapping

The following AXI-Lite devices are connected through a crossbar, allowing them to be accessible through the single S00_AXI_0 interface.

Table 5: adc_IF_core AXI-Lite Address Mapping

Internal IP	Address Bits	Offset Address	Range	High Address
delay_FIFO	32	0x44A00000	4K	0x44A00FFF
power_meter	32	0x44A10000	4K	0x44A10FFF

6.6 Usage

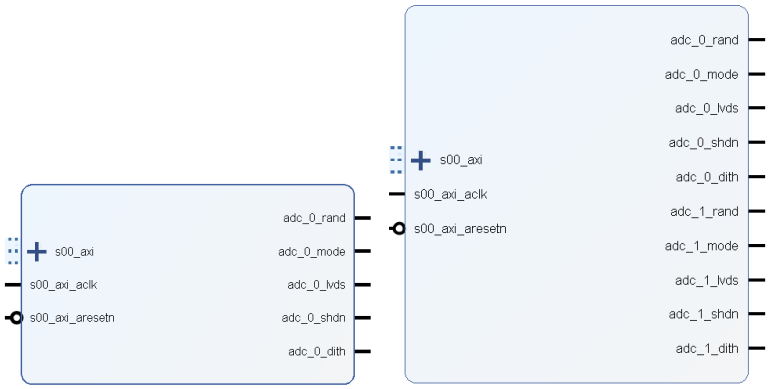
This IP requires a reset pulse to be delivered at start up. Once the device is ready for operation, the deviceEnabled line will go high, allowing the internal FIFO to begin reading and writing data. Most harmful events are prevented from existing in this IP, but is often done by ignoring input data and clocks while the device is in a busy state. So long as important data is not written while the deviceEnabled line is low, this should not be an issue. Once deviceEnabled becomes high, data will be read by this IP, and, after the appropriate delay has passed, written across the AXI-Stream interface. In addition, the delay_FIFO and power_meter will become accessible over AXI-Lite, where each can be written to and read from. Although they are not physically prevented, these AXI-Lite connected IP should not be communicated with while in a reset state.
In order to modify the internal IP's parameters, most notably the delayTapVal of the IDELAYBUFDS, one must edit the adc_IF_core block design and re-package, since the IP Integrator design flow does not support externalizing parameters. Finally, the slow clock frequency should be that of the ADC sampling frequency, and should be connected to all adc_IF_cores, so they all extract data simultaneously.

7 AXI_ADC_ctrl

This IP is an AXI-Lite controlled GPIO unit, specifically for driving the control GPIO lines of the LTC2217. Since there are only five GPIO inputs on each LTC2217, it's not space efficient to use a 32-bit AXI-Lite register for each ADC. The purpose of this IP is to increase space efficiency of the AXI registers and AXI-Lite address mapping, by putting the control lines for up to 32 ADCs into a single IP. Currently, the IP gets very large when

all outputs are enabled. A custom interface could be made for each ADC GPIO group, but has not yet done as this configuration met all requirements of the current Jay-1 design, especially since it will only contain two ADCs.

7.1 Diagram



(a) Default-Single ADC (b) Outputs for Two ADCs

Figure 9: IP Integrator diagram of the AXI_ADC_ctrl

7.2 Parameters

Non-AXI related parameters:

numChannels Specifies the number of ADCs that will be controlled by the IP. The necessary output pins will become visible to accommodate the specified number.

7.3 Pin Description

Where x is a place holder for the ADC number, from 0 to 31, that the pin is associated to.

s00_axi_aclk Active high AXI-Lite clock input

s00_axi_aresetn Active low reset

adc_*x*_mode Active high GPIO output for 'random' input of LTC2217

adc_*x*_mode Active high GPIO output for 'mode' input of LTC2217

adc_*x*_lvds Active high GPIO output for 'LVDS' input of LTC2217

adc_*x*_shdn Active high GPIO output for 'shutdown' input of LTC2217

adc_*x*_dith Active high GPIO output for 'dither' input of LTC2217

7.4 AXI Register Mapping

Table 6: AXI_ADC_ctrl Register Map

	slv_reg0	**slv_reg1**	**slv_reg2**	**slv_reg3**	**slv_reg4**	**slv_reg5**
bits 4-0	0	6	12	18	24	30
bits 9-5	1	7	13	19	25	31
bits 14-10	2	8	14	20	26	Unused
bits 19-15	3	9	15	21	27	Unused
bits 24-20	4	10	16	22	28	Unused
bits 29-25	5	11	17	23	29	Unused
bits 31-30	Unused	Unused	Unused	Unused	Unused	Unused

The integers above represent ADC number that the five enclosed bits are mapped to. For example, the register bits in 15, spanning bits 19-15, are mapped to the five outputs connected to the ADC 15, where each individual control line is mapped to the following bits:

Table 7: Control Line Mapping for Each ADC

adc_*x*_outputs				
bit 4	bit 3	bit 2	bit 1	bit 0
adc_*x*_rand	adc_*x*_mode	adc_*x* _lvds	adc_*x*_shdn	adc_*x*_dith

7.5 AXI Register Description

Each bit in the registers are active high, and associated to the pin with the same name.

7.6 Usage

This IP requires a result pulse at boot up. After this, registers can be written with a logical one to turn on the associated pin. The state of the pins will

immediately change once the registers are updated from an AXI write.
As an example, if the user wants to turn on the adc_5_mode pin, they should write a logical one into bit 28 of slv_reg0, since ADC five is located in bits 29-25 of slv_reg0 and the mode bit is four bits over.
In addition, the registers can be read to query the state of the control pins.

8 AXI_DAC_CTRL

This IP is an AXI-Lite controlled IO unit, specifically for driving the clock and control IO lines of the MAX5885 digital to analog converter *DAC*. Internally, a PRBS_gen IP, and an OBUFDS primitive are instantiated to generate the random bit stream and differential sample clocks required by the DAC. Unlike the AXI_ADC_ctrl, this IP is designed to be implemented within the DAC_IF _core, but can be independently implemented if desired.

8.1 Diagram

Figure 10: IP Integrator Diagram of the AXI_DAC_CTRL

8.2 Parameters

Non-AXI related parameters:

randomGenBits Number of bits used within internal random generator. Default is seven

reset_tapAddr Tap address for internal random number generator inserted in AXI register at reset. This will change the tap used when a reset is provided

reset_xorOn State of tx_dac_xor inserted in AXI register at reset. This will enable or disable the XOR feature when a reset is provided

8.3 Pin Description

tx_dac_xor XOR line on which a random bit stream will established when the XOR feature of this IP is enabled using the respective AXI register. The purpose of this line is to inform the DAC when the data has been randomly inverted so the DAC can re-invert it to its proper value. A logical one indicates that the data from the dac_IF_core has been inverted. For more information on the XOR function of the DAC, refer to the dac_IF_core.

tx_dac_reset Active high reset pin for DAC

tx_dac_sw_a Active high control line that switches on the signal path

tx_dac_clk_p Positive differential sample clock output. This is in phase with the dataClk input and the transmit data

tx_dac_clk_n Negative differential sample clock output. This is directly out of phase with the dataClk input and the transmit data

dataClk Single ended sample clock input

dataValid Input that indicates when the dac_IF_core has valid transmit data. When this is low, the PRBS_gen and DAC are reset, regardless of dedicated reset inputs

s00_axi_aclk Active rising edge AXI-Lite clock

s00_axi_aresetn Active low reset that resets the PRBS_gen, the AXI-Lite registers, and the DAC with the tx_dac_sw_

s00_axi AXI-Lite bus containing all necessary lines to meet the protocol

8.4 AXI Register Mapping

Table 8: AXI-Lite Register Mapping for AXI_DAC_CTRL

Register Name	**Address Offset**	**Register Map (32bits per reg.)**			
slv_reg0	4'h0	dynamic_tapAddr	tx_dac_xor	tx_dac_reset	tx_sw_ctrl_a
		bits 31-3	bit 2	bit 1	bit 0
slv_reg1	4'h4	Unused			
		bits 31-0			
slv_reg2	4'h8	Unused			
		bits 31-0			
slv_reg3	4'hC	Unused			
		bits 31-0			

8.5 AXI Register Description

dynamic_tapAddr (w) Output pin address of the random bit stream generator to connect to tx_dac_xor. This can be changed at runtime. Although 29 AXI register bits are allocated, in default operation, only the least significant four bits will modify the tap address and the rest will be ignored. If this IP is modified, the number of valid bits in this register becomes $log_2 randomGenBits+1$, the same number of address bits the internal PRBS_gen requires

tx_dac_xor (w) Active high bit enabling or disabling the random bit stream generation

tx_dac_reset (w) Active high reset for the DAC and PRBS_gen. The AXI-Lite transceiver will not be reset upon assertion of this bit

tx_dac_sw_ctrl_a (w) Active high bit that drives the tx_dac_sw_a pin to the same state

8.6 Usage

8.6.1 General

This IP is designed to be implemented within the DAC_IF_core IP; nevertheless, it may also be used independently. It requires a reset pulse at boot

up and. Once complete, the I/O pins can be driven to the desired state by setting the appropriate AXI registers.

8.6.2 Clocking

A single-ended sample clock is supplied to the AXI_DAC_CTRL where it is converted to 3.3 V differential CMOS lines for the MAX5885. The CLKP and CLKN input clock pins on the MAX5885 are LVPECL (*Low Voltage Positive Emitter Coupled Logic*), but the 7-Series FPGAs don't contain LVPECL logic. Luckily, these clock inputs are rated for 3.3 V and are AC coupled on the Jay-1 TRX board, allowing them to be driven with CMOS signals. In order to use the 3.3 V CMOS clock signals provided by this IP, VCLK on the MAX5885 needs to be 3.3 V.

9 shift_register

This IP is a completely configurable shift register, including a configurable size, configurable control pins, and configurable modes of operation. It should be noted that registers are implemented in FPGA fabric, not in dedicated BRAM. All of the following modes are supported:

- PIPO - *Parallel In, Parallel Out*
- PISO - *Parallel In, Serial Out*
- SIPO - *Serial In, Parallel Out*
- SISO - *Serial In, Serial Out*

9.1 Diagram

The default configuration, shown below, enables all available control pins in a PIPO configuration with output latches enabled. When the input or output mode is set to serial, the input/output channel will scale to a single bit wide.

Figure 11: IP Integrator Diagram of the shift_register

9.2 Parameters

size Number of bits/flip-flops to internally instantiate. Outputs and inputs scale accordingly

resetValue Decimal representation of the bit array desired at the output of the internal flip-flops. Default is zero

parallel_input_en Single bit which enables or disables the parallel input feature. When disabled, with a logical zero, the input mode will become serial. The default is 1'b1 (parallel input enabled)

parallel_output_en Single bit which enables or disables the parallel output feature. When disabled, with a logical zero, the output mode will become serial. The default is 1'b1 (parallel output enabled)

output_latch_en Single bit which enables or disables the output latch feature. When disabled, with a logical zero, no output latches are instantiated and the latch_pin is removed. The default is 1'b1

preset_en Enables the synchronous preset input pin

clear_en Enables the synchronous clear input pin

enable_en Enables the synchronous enable input pin

9.3 Pin Description

reset Active low reset input, driving output of all internal flip-flops to the value specified with resetValue. This pin has the highest control pin priority.

clear Active low clear input, driving output of all internal flip-flops to zero. This pin has the second highest control pin priority.

preset Active low preset input, driving output of all internal flip-flops to one. This pin has the third highest control pin priority.

enable Active low enable input, enabling all internal flip-flops to change state. This pin has the lowest control pin priority.

latch_control Active rising edge input that latches the flip-flop states to the output pins.

write_shift Input determining whether the shift_register will accept a write from the parallel input wires (logical zero), or will shift the data currently stored (logical one) on the following positive clock edge. This pin is only enabled when parallel _input_en is set to 1'b1

clock Active rising edge clock for loading and shifting data

dataIn Channel for input data. Adjusts size and connection based on the parameters specified

dataOut Channel for output data. Adjusts size and connection based on the parameters specified

9.4 Operation

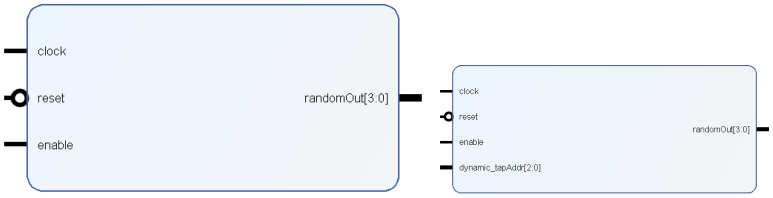
A reset pulse should be delivered at boot up to initialize the flip-flop states to the value specified with the resetValue parameter. In parallel input mode, the write_shift input selects whether to write the data on the input lines to the flip flops or shift the data currently there, if permitted by the control lines. Serial input mode will connect the dataIn channel to the first flip-flop's input, and serial output mode will connect the dataOut channel to the last flip-flop's non inverted output.
This device supports configurable output latches, which are enabled by setting output_latch_en to 1'b1. When enabled, a logic high on the latch_control input will latch whatever data is on the output(s) to the output pins. These latches are not affected by the control inputs, and will simply output the last latched value until the latch_control becomes high.

10 PRBS_gen

The PRBS_gen stands for *Pseudo Random Bit Sequence Generator*, and is a configurable source for random values. Internally, it contains a parallel output shift_register, used to create *Linear Feedback Shift Register* for random number generation. Exclusive NOR gates are used in the feedback path and are automatically instantiated and connected based on the parameters to maximize the sequence length. This IP supports variable seeds, single and parallel outputs, dynamic tap addressing, and variable size. Available bit lengths are 2-786, 1024, 2048, and 4096. When the output is serial, the *tap* refers to which shift register output the output pin is connected to.

10.1 Diagram

Both diagrams are shown with a bit length of four.



(a) Default Configuration (b) Dynamically Tapped

Figure 12: IP Integrator Diagram of the PRBS_gen

10.2 Parameters

numBits Number of bits/flip-flops used in *LFSR*

parallel_out_en Single bit to enable or disable parallel output. When disabled with a logical zero, the output becomes serial, tapped either statically or dynamically

seed Decimal value of the seed for random number generation. This is the value of the shift register output lines at reset

enable_en Single bit enabling the enable input terminal on the device

static_tapAddr Zero indexed address of the shift register output to which the serial output shall be connected in serial output mode. This value is zero by default, and only applies if dynamic_tap_en is inactive

dynamic_tap_en Single bit enabling or disabling the dynamic tap addressing feature. The dynamic_tapAddr input pins will be visible when this parameter is set to 1'b1

10.3 Pin Description

clock Rising edge active clock used to generate a new random value

reset Active low synchronous reset pin

enable Active high synchronous enable pin. While inactive, this pin will prevent the device from proceeding to the following random number. Clock signals while disabled will be ignored

randomOut Output channel for random values. Scales based on the parallel_out_en parameter

dynamic_tapAddr Input channel for dynamic tap address which is only visible when the dynamic tap is enabled and is used to select the desired tap between the serial output and shift register output. The width of this channel is automatically calculated as $log_2(numBits)+1$

10.4 Usage

This IP requires a reset pulse to be delivered at boot up to initialize the shift register output lines to the seed value. After this, with the enable pin high, a new random value will be generated at every rising positive edge. When using serial output, the output channel is connected to a single shift register output pin, which is selectable statically or dynamically using the appropriate parameters and addressing.

11 channel_delay

This IP is an extensible wrapper around the IDELAYE2 primitive, allowing multiple to be instantiated within a single IP, thereby creating a channel. In 7-series FPGAs, each I/O block contains an IDELAYE2 element, and each element can only access the input within its I/O block. As a result, each line within the channel_delay IP has an independent delay element. Nevertheless, the channel_delay IP only supports a single, fixed tap value for all delay elements. This design creates the perception of a single delay element which delays all lines within a channel to the same degree. Further

information regarding the internal IDELAYE2 primitive from Xilinx can be found in the *UG953* reference guide.

11.1 Diagram

Figure 13: IP Integrator Diagram of the channel_delay

11.2 Parameters

numElements Width of delay channel

refClkFreq The frequency in MHz of the reference clock, provided through an IDELAYCTRL module. Available value are 190-210 MHz and 290-310 MHz for the 7-Series FPGAs

highPerformance_delay Enables or disables high performance mode of the IDELAYE2 in exchange for power savings. Available options are "TRUE" and "FALSE"

delayVal Selectable tap value which selects the delay imposed upon the signal, equal to $1/(64 \times refClkFreq)\mu S$, where refClkFreq is in MHz

signalType Describes the signal type for the internal IDELAYE2. Available values are "CLOCK" and "DATA", and the default value is "DATA"

11.3 Pin Description

dataIn Input channel

dataOut Output channel

reset Active low reset

11.4 Usage

As this IP includes an IDELAYE2, it requires an IDELAYCTRL device to be instantiated in the same design in order to maintain calibration. Although there is not a physical wire routed between the two, there is an internal bus designated for the reference clock signals in the FPGA, and once both are instantiated, they will be placed on this bus upon implementation.

12 dac_IF_core

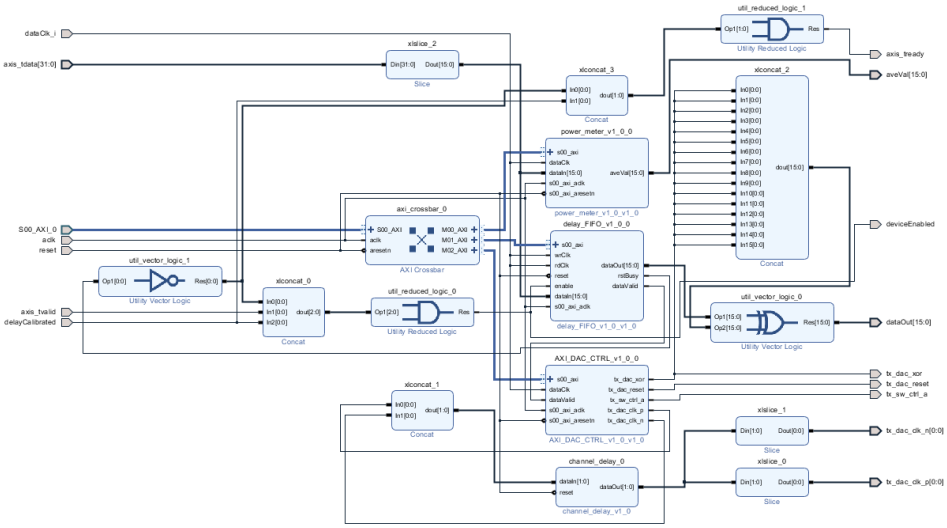
This IP is specifically designed to interface and control the MAX5885 digital to analog controller (DAC) within the transmit path of the Jay-1 ionosonde. It acts as an AXI-Stream slave, receiving 16-bit samples and directing them to the DAC over 16 parallel CMOS 3.3 V lines. In addition, this IP provides an AXI-Lite interface for configuration and settings during runtime.
The output signal passes through a delay_FIFO for delay tuning, and can be inverted for the XOR feature of the MAX5885. A channel_delay element is contained in the output path for the DAC sample clock, allowing phase tuning of the sample clock to meet the sample data.
In addition, a power_meter is housed within this IP, and directly reads the transmit power from the AXI-Stream input port. This power meter, along with the delay_FIFO, and AXI_DAC_CTRL can all be communicated with through a single AXI-Lite interface with the help of a contained crossbar.
Internally, this IP contains the logic to prevent data corruption from enablement or reset while the IP is busy, and logic to meet reset requirements of contained IP.
For further information on the MAX5885, please refer to its data sheet located below.

12.1 Contained IP

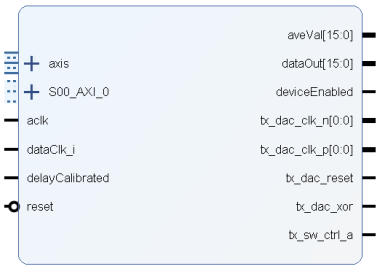
Table 9: Contained IP Within adc_IF_core

Custom IP	Xilinx IP
• channel_delay • AXI_DAC_CTRL • power_meter • delay_FIFO • IDELAYBUFDS	• axi_crossbar • util_reduced_logic • util_vector_logic • xlconcat • xlslice

12.2 Diagrams



(a) Internal Diagram



(b) dac_IF_core Package

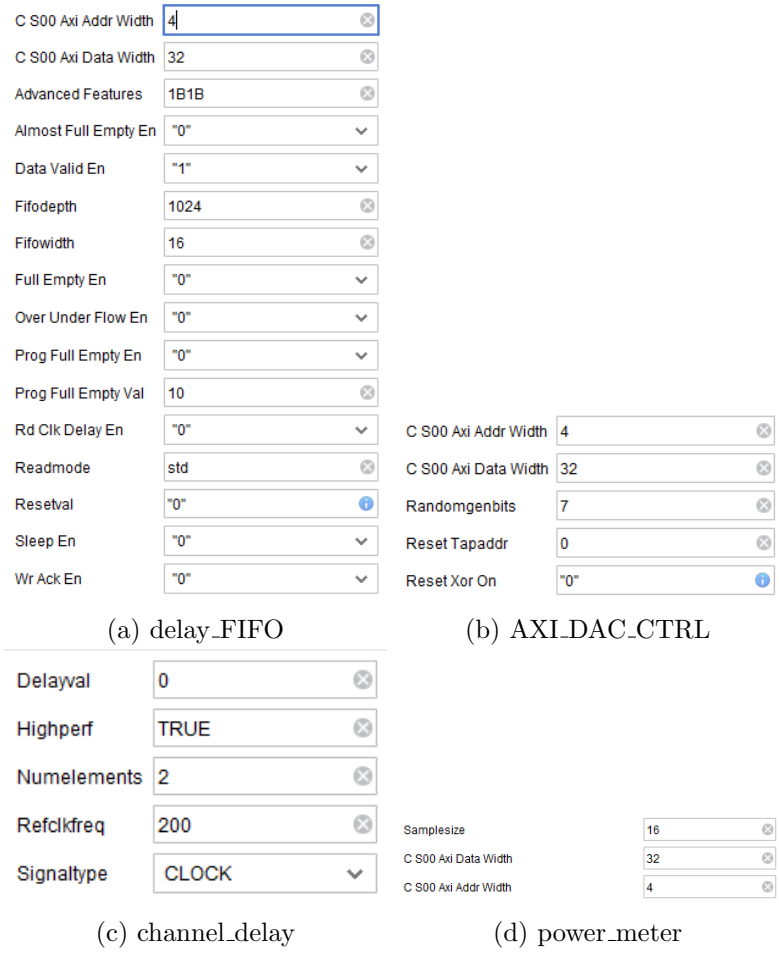
Figure 14: IP Integrator Diagrams of the dac_IF_core

12.3 Parameters

Since this IP is very specific to its application, it does not contain any global parameters. All internal IP is pre-configured to meet the application requirements. Sadly, the IP flow does not support passing global parameters to internal IP, preventing externalizing composing IP's parameters. The delayVal parameter of the channel_delay suffers most from this, as it may need modification to match the sample data phase. If one would like to alter these default parameters, they can do so by modifying the design in the IP Integrator and re-packaging.

12.3.1 Internal IP Default Parameters

Every IP has default parameters, most of which are obvious or are described by the IP operation. The important parameters are listed in the following figures.



(a) delay_FIFO (b) AXI_DAC_CTRL

(c) channel_delay (d) power_meter

Figure 15: Default Parameters of the dac_IF_core's Internal IP

12.4 Pin Description

aclk AXI-Lite active rising edge clock

reset Active low reset for all logic, including AXI-Lite interface

dataClk_i Single-ended, active rising edge sample. Essentially the AXI-Stream clock from the AXI-Stream master which is in phase with the input data. Writes and reads data to and from the dac_IF_core,

delayCalibrated Flag from IDELAYCTRL module that must be instantiated within the same design as this IP. Informs the dac_IF _core that the delay filter is stable/calibrated. This IP is internally prohibited from reading from the AXI-Stream interface or writing to the DAC through the FIFO while this is low.

aveVal 16-bit, single-ended output channel from the internal power_meter, relaying the average power for other hardware devices

dataOut 16-bit, single-ended output channel that sends the transmit data to the DAC. This data may be inverted for the XOR functionality, and delayed for phase tuning.

tx_dac_clk_n Negative CMOS output terminal with the data clock. This signal may be delayed for phase tuning with the internal channel_delay but will always be 180° out of phase with tx_dac_clk_p

tx_dac_clk_p Positive CMOS output terminal with the data clock. This signal may be delayed for phase tuning with the internal channel_delay but will always be 180° out of phase with tx_dac_clk_n

tx_dac_reset Active high reset pin for the DAC

tx_dac_xor Active high output which, when the feature is activated, contains a random bit stream for XOR inversion

tx_sw_ctrl_a Active high output which activates the transmit path switch

deviceEnabled Active high output flag which indicates the internal FIFO is enabled; the IP is able to receive data. Specifically, this indicates:

1. delayCalibrated is high
2. AXI-Stream interface contains valid data, indicated by a logic one on the valid pin
3. the FIFO is not within its reset procedure
4. active low reset line is deactivated

axis AXI-Stream interface. Contains necessary lines to match protocol. As these lines are simple enough to be implemented in a design with discrete logic, their purposes are listed below:

- **axis_tvalid** Indicates to the slave that valid data is available
- **axis_tready** Indicates to the master that the slave is ready for data
- **axis_tdata** 32-bit data channel

  A master must implement the valid and data signals and read the ready line.

S00_AXI_0 AXI-Lite bus containing all necessary lines for the protocol

12.5 AXI Interface Mapping

Both the AXI-Stream data channel mapping and the AXI-Lite address mapping are included in this section. For a detailed mapping of each register, please refer to the respective IP description above.

12.5.1 AXI-Stream Interface

Table 10: Data Channel Mapping - axis_tdata

Channel: bits 31 - 0	
Unused	dataOut
bits 31 - 16	bits 15-0
left unconnected	data input containing transmit samples

12.5.2 AXI-Lite Address Mapping

The following AXI-Lite devices are connected through a crossbar, allowing them to be accessible through the single S00_AXI_0 interface.

Table 11: dac_IF_core AXI-Lite Address Mapping

Internal IP	**Address Bits**	**Offset Address**	**Range**	**High Address**
AXI_DAC_CTRL	32	0x44A00000	4K	0x44A00FFF
delay_FIFO	32	0x44A10000	4K	0x44A10FFF
power_meter	32	0x44A20000	4K	0x44A20FFF

12.6 Usage

12.6.1 General

This IP requires a reset pulse to be delivered at start up. Once the device is ready for operation, the deviceEnabled line will go high, allowing the internal FIFO to begin reading and writing data. Most harmful events are prevented from existing in this IP, but is often done by ignoring input data and clocks while the device is in a busy state. So long as important data is not written while the deviceEnabled line is low, this should not be an issue. Once deviceEnabled becomes high, data will be read by this IP, and, after the appropriate delay has passed, written to the DAC. In addition, the AXI_DAC_CTRL, delay_FIFO, and power_meter will become accessible over AXI-Lite, where each can be written to and read from. Although they are not physically prevented, these AXI-Lite connected IP should not be communicated with while in a reset state.
In order to modify the internal IP's parameters, most notably the delayVal of the channel_delay, one must edit the dac_IF_core block design and re-package, since the IP Integrator design flow does not support externalizing parameters.

12.6.2 XOR Data Inversion

In order to decorrelate the input bit transitions from the DAC output, the MAX5885 supports XOR inversion. When the data signal is randomly inverted with XOR gates, the EMI produced will contain a wide frequency content of lower amplitude. To support this feature, bank of XOR gates is located on the MAX5885 data input and the dac_IF_core data output. When enabled over the AXI-Lite interface, the AXI_DAC_CTRL will generate a random bit stream which gets delivered to the output inversion bank, and to the MAX5885, over tx_dac_xor, so the data can be re-inverted.

13 VGA_SW_IF

This IP is an interface specifically built for the LMH6517 variable gain amplifier *VGA* and RF switch located on the receiver path of the Jay-1 ionosonde. It receives input from the gain compensation block, for long term gain bias, and an input from the automated gain control algorithm (AGC), which determines the optimal gain for the ADC based on the received signal power level. Both inputs to this IP are in tenths of a dB, or cB. This IP contains

an internal state machine that provides the LMH6517 with the pulse interface it requires to modify the VGA gain based the two inputs, and reset the device. Furthermore, this IP contains a AXI-Lite interface for configuration and software control, and a timing clock to meet the strict pulse timing requirements of the LMH6517's pulse interface. For more information on this VGA, please refer to *SNOSB19K*.

13.1 LMH6517 Block Diagram

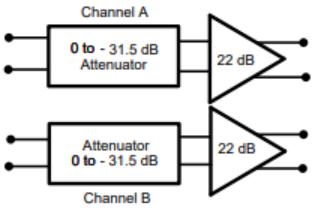


Figure 16: Block Diagram of LMH6517 VGA

13.2 Diagram

The default configuration is shown below, which has all pins enabled.

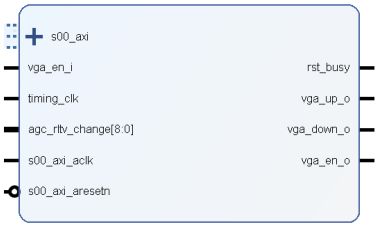


Figure 17: IP Integrator Diagram for the VGA_SW_IF

13.3 Parameters

Non-AXI related parameters:

max_gain Maximum gain possible by the VGA in tenths of a dB. Default value is 220 and this parameter must contain an integral multiple of five tenths of a dB

min_gain Minimum gain possible by the VGA in tenths of a dB. Default value is -95 and this parameter must contain an integral multiple of five tenths of a dB

vgaEnable_en Single active high bit that enables or disables the hardware vgaEnable pin on the IP

nominal_gain VGA gain desired at reset in tenths of a dB. This parameter must contain an integral multiple of five tenths of a dB and must be within max_gain and min_gain. The default value is 160

13.4 Pin Description

vga_en_i Active high input enable pin for VGA. Either a logic one on this input or the enable AXI register will cause the device to be enabled , so long as the disable pin is low, which has the highest priority. This hardware pin can removed with the vgaEnable_en parameter

timing_clk Clock used to ensure timing requirements for VGA pulses are met. This clock should have a period equal to the hold time of the pulse lines - 20 ns in this case. As such, in the current design this should be 50 MHz

agc_rltv_change Desired relative gain change from the AGC, with reference to nominal_gain, in tenths of a dB. These bits should be formatted in two's complement and must contain a integral multiple of five tenths of a dB

rst_busy Active high output flag indicating that the device is within its reset sequence. This should control the RF input switch

vga_up_o Active low *up* line for pulse interface to VGA. An independent low pulse of this line will raise the VGA gain by 0.5dB, if permitted

vga_down_o Active low *down* line for pulse interface to VGA. An independent low pulse of this line will reduce the VGA gain by 0.5dB, if permitted

vga_en_o Active high output enable for VGA

s00_axi_aresetn Active low reset for VGA and AXI-Lite. A low pulse of this pin or the reset AXI bit will cause the device to enter the reset sequence

s00_axi AXI-Lite bus that contains all signals for the protocol

13.5 AXI Register Mapping

Table 12: VGA_SW_IF Register Map

<table>
<tr><th></th><th>slv_reg0</th><th>slv_reg1</th><th>slv_reg2</th><th>slv_reg3</th></tr>
<tr><td>bits 0-8</td><td>comp_rltv_change</td><td>current_rltv_output</td><td rowspan="11">Unused</td><td rowspan="11">Unused</td></tr>
<tr><td>bit 9</td><td>reset</td><td rowspan="4">current_rltv_agc_val</td></tr>
<tr><td>bit 10</td><td>enable</td></tr>
<tr><td>bit 11</td><td>disable</td></tr>
<tr><td>bits 12-17</td><td rowspan="7">Unused</td></tr>
<tr><td>bits 18-26</td><td>nominal_gain</td></tr>
<tr><td>bit 27</td><td>rst_busy</td></tr>
<tr><td>bit 28</td><td>underflow</td></tr>
<tr><td>bit 29</td><td>overflow</td></tr>
<tr><td>bits 30-31</td><td>Unused</td></tr>
</table>

13.6 AXI Register Description

disable (w) Regardless of the enable AXI bit's or enable input pin's state, a logical one on this pin will disable the device

enable (w) Active high enable bit. This bit, or a logical one on the enable input pin, will enable device, so long as the disable bit is inactive. At reset, this bit is internally driven high so this IP can be used without having to perform an AXI write

reset (w) Active low reset bit. A logic zero on this bit or on the s00_axi_aresetn pin will cause this IP to enter its reset sequence and will reset the AXI-Lite interface. At reset, this bit is driven high so the device can be used without having to perform an AXI write

comp_rltv_change (w) Compensation Relative Change: these bits receive the desired relative change in gain from the gain compensation in tenths of a dB. These bits should be formatted in two's complement and must contain an integral multiple of five tenths of a dB.

overflow (r) Active high flag indicating an overflow condition has occurred

underflow (r) Active high flag indicating an underflow condition has occurred

rst_busy (r) Active high flag indicating when the IP is within its reset sequence. Also indicates the state of the RF input switch

nominal_gain (r) Nominal gain parameter set upon compilation. Allows software to read value if required. This is formatted as a two's complement number in tenths of a dB

current_rltv_val (r) Current relative difference between VGA_gain and nominal_gain in tenths of a dB. This will update as each pulse is delivered and is formatted in two's complement

current_rltv_agc_val (r) Current relative difference between agc_rltv_change and nominal_gain in tenths of a dB. This will update as the AGC value changes and is formatted in two's complement

13.7 Usage and Operation

13.7.1 General

This IP requires a reset pulse to be delivered at boot up. After the rstBusy line goes low, the reset sequence has completed, and the VGA gain will be modified in accordance with the two relative inputs. However, at any instance that the vga_en_i line goes low, the VGA_SW_IF will stop delivering pulses to the VGA and the vga_en_o will be driven low.
Each individual VGA channel requires an independent VGA_SW_IF. The LMH6517 contains two channels, and as such, if both are operational, two VGA_SW_IFs are required.
In addition, the agc_rltv_change and comp_rltc_change can be overwritten at any rate to any value. This IP will simply direct the VGA gain to the latest value, even if it has yet to reach the previous value.

13.7.2 Reset Sequence

Inherently, the LMH6517 sets its gain to the maximum possible value of 22.0 dB upon receiving a reset. This could be destructive to the ADC or other receiver components, as they may become overloaded at reset. In order to avoid this, the VGA_SW_IF generates a reset sequence, where it will turn off the RF input switch, reset the VGA, configure the gain down to the nominal value, and then turn on the RF input switch.

13.8 Pulsed Interface

A pulsed interface is employed by this IP to modulate the gain on the LMH6517, which it natively supports. This consists of an up and down line, each of which are active low. When either line goes low for 20 ns, with a minimum setup time of 20 ns, the VGA gain moves in the associated direction by 0.5dB. As a result, the slew rate for gain variation is very slow - 0.5dB/40ns.
The VGA_SW_IF ensures that this timing is met with the help of the timing_clk.

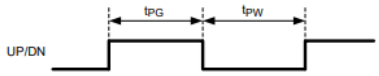


Figure 18: Pulse Timing Requirement

13.8.1 Overflow and Underflow Detection

The values in comp_rltv_change and agc_rltv_change get internally summed, where the result becomes the desired deviation from the nominal gain that the VGA will be driven to. This IP detects both over and underflow conditions when this sum produces an arithmetic binary over or underflow, and when the sum exceeds the bounds of $max\_gain - nominal\_gain$ and $min\_gain - nominal\_gain$. At any of these incidents, the gain will be driven to min_gain and the associated flags will be set until an acceptable value is provided.

14 References and Links

LT0108 Linear Technology. (n.d.). 16-Bit, 105Msps Low Noise ADC. Retrieved August 15, 2019, from https://www.analog.com/media/en/technical-documentation/data-sheets/2217f.pdf

DAC Maxim Integrated. (2003, December). 3.3V, 16-Bit, 200Msps High Dynamic Performance DAC with CMOS Inputs. Retrieved August 15, 2019, from https://datasheets.maximintegrated.com/en/ds/MAX5885.pdf

PRBS Pini, A. (2018, March 22). Use Readily Available Components to Generate Pseudo-Random Binary Sequences and White Noise. Retrieved August 15, 2019,from https://www.digikey.ca/en/articles/techzone/2018/mar/use-readily-available-components-generate-binary-sequences-white-noise

SNOSB19 Texas Instruments.(2013, March). Low Power, Low Noise, IF and Baseband, Dual 16 bit ADC Driver With Digitally Controlled Gain. Retrieved August 15, 2019, from https://www.ti.com/lit/ds/symlink/lmh6517.pdf

UG953 Xilinx. (2018, June 6). Vivado Design Suite 7 Series FPGA and Zynq-7000 SoC Libraries Guide. Retrieved August 15, 2019, from https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_2/ug953-vivado-7series-libraries.pdf

DS182 Xilinx. (2019, June 28). Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics. Retrieved August 15, 2019, from https://www.xilinx.com/support/documentation/data_sheets/ds182_Kintex_7_Data_Sheet.pdf