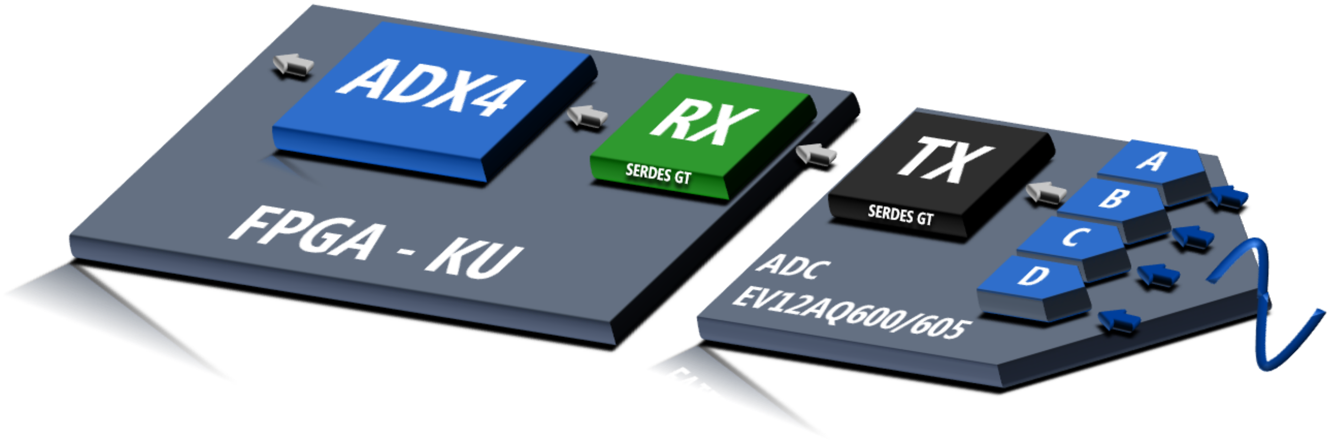
ADX4 IP

Xilinx Kintex Ultrascale



1 Summary

2 Introduction

Teledyne e2v provides the ADX4 IP and a vhdl wrapper to reduce EV12AQ600/605 ADC time interleaving spurious effects (fc/4-fin, fc/4, fc/4+fin and fc/2-fin) when the ADC is configured in 1 channel mode (4 ADC cores are interleaved).

When the ADC is configured in 2 channels mode, the ADX2 IP must be used to reduce time interleaving spurious effects (fc/2-fin).

When the ADC is configured in 4 channels mode, all cores are independent and no ADX IP is required. No time interleaving spur to remove in 4 channels mode.

ADX4 IP improves cancellation of time interleaving mismatch effects like gain, phase and DC-offset by real time post processing treatment in the FPGA at any temperature, signal input frequency and ADC sampling rate.

ADX4 IP must be implemented on a Xilinx Kintex Ultrascale FPGA.

For another FPGA reference, please contact your local FAE or send an email at GRE-HOTLINE-BDC@Teledyne.com.

To learn more on how the ADX IP and time-interleaved ADC error correction work, Teledyne SP Devices provides technical details in a webinar and a white paper available here: https://www.spdevices.com/technology/interleaving.

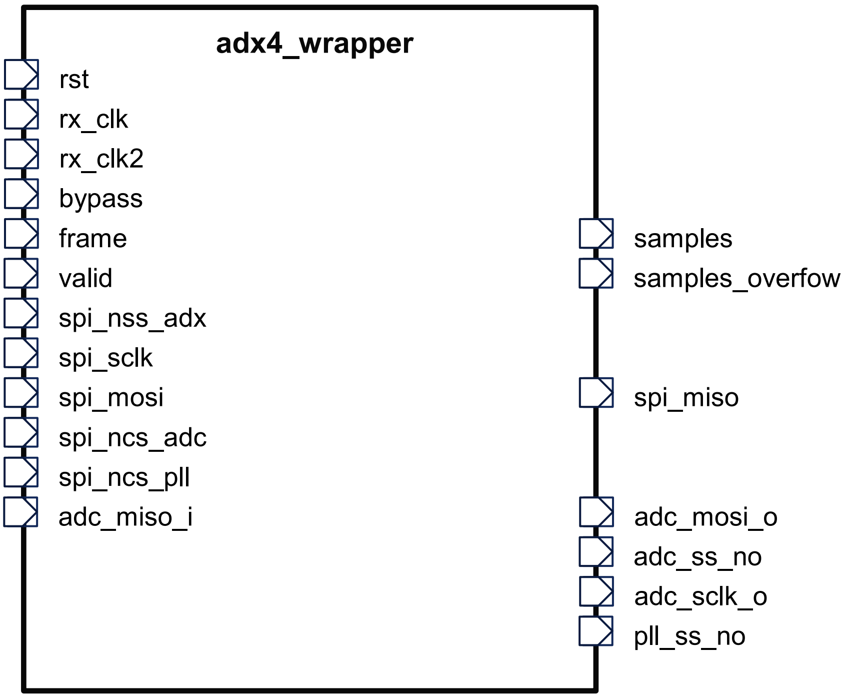


Figure 1 – ADX4 IP VHDL wrapper overview.

3 Implementation

To **start your development** and **download ADX4 IP design example and user guide**, please contact your local FAE or send an email at GRE-HOTLINE-BDC@Teledyne.com.

The VHDL ADX4 wrapper (see Figure 1) contains the ADX IP in edif format.

The ADX4 wrapper does the following:

- It converts the EV12AQ600 ADC sample coding output from Bipolar Offset Binary (**BOB**) to 2s-complement (**2C**) to comply with the ADX IP coding input.
- It converts the EV12AQ600 ADC 12-bit resolution to 16-bit to comply with the ADX IP resolution. As the ADC coding output is not strictly Bipolar Offset Binary, the 2s-complement 12-bit value needs to have half a bit added by extending with "1000" instead of "0000" to create the 16-bit sample.
- It converts the Control Bit 1 (**CB1**) of each ESIstream frame, configured with ADC InRange bit, to create the ADX IP overflow input bit (one per sample).
- It swaps the High-Speed Serial Lanes depending on layout and FPGA pinout and organizes the samples in the correct order (x_i ADX4 IP input).
- It multiplexes the ADX IP SPI master and the application SPI master (FPGA) to access the ADC SPI slave. The ADC needs to be configured over SPI by the user application and in addition the ADX IP must also be able to acquire the license information via the same interface. To solve this a mux is introduced to switch between SPI masters. The ADX4 wrapper provides a reference implementation.

Figure 2 shows an implementation overview.

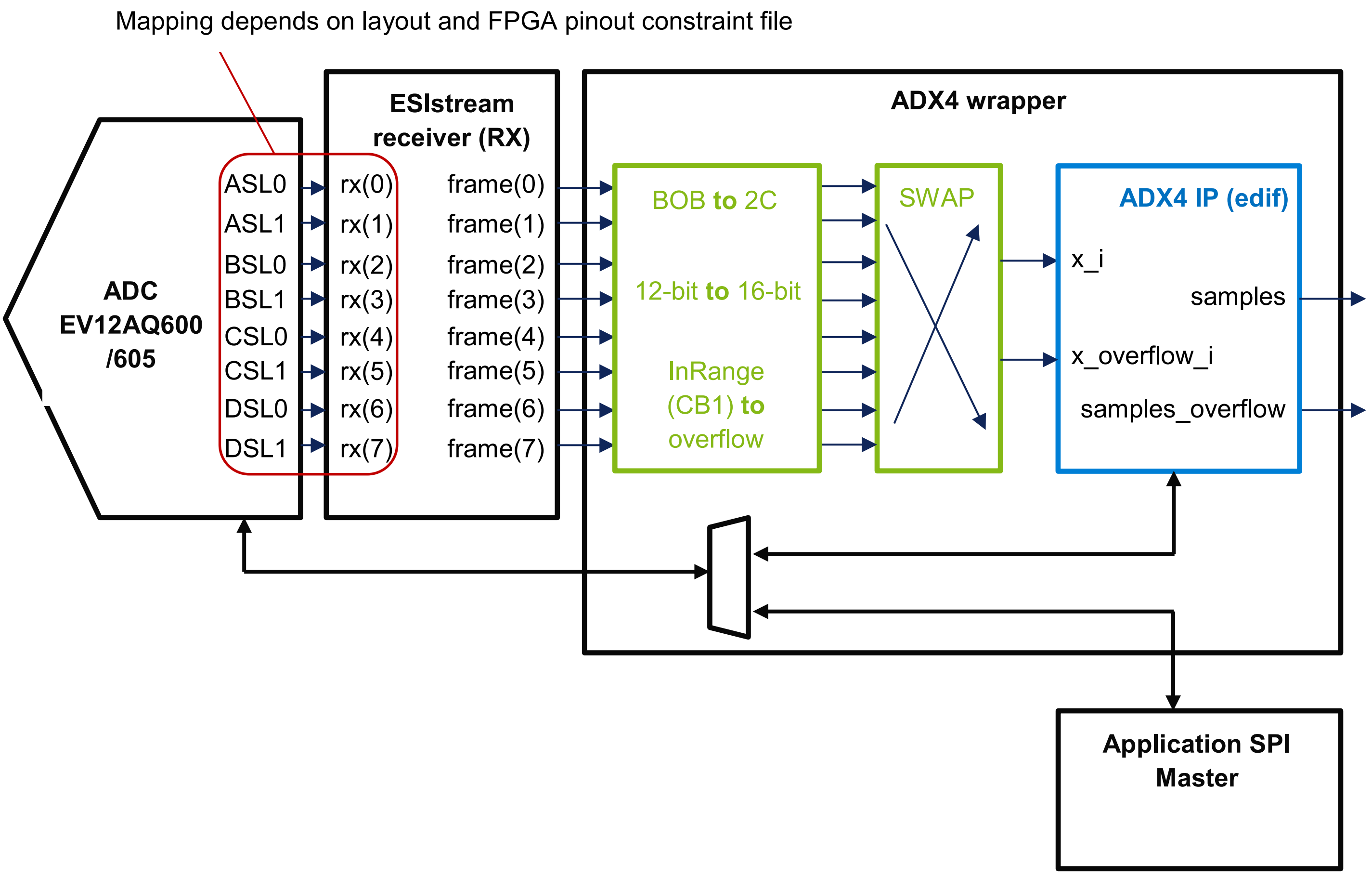


Figure 2 – implementation overview.

3.1 Interface description

3.1.1 Generic parameters

Parameter Name	Type	Description
SWAP_LANES	array [0:7] of Integer in range 0 to 7.	Allows to swap frame input signal according to lanes layout and mapping to the FPGA.
DESER_WIDTH	integer	ESIstream RX deserialization factor. Must be set to 64.
NB_LANES	integer	EV12AQ600 ADC number of lanes in 1 channel mode. Must be set to 8.

3.1.2 Input Outputs

Port name	Direction	Type	Description
rst	In	1-bit	Active high ADX IP reset.
rx_clk	In	1-bit	Frame clock frequency. $f_{rx\_clk}=f_{serial}/64=f_{adc\_clk}/32$
rx_clk2	In	1-bit	Half frame clock frequency. $f_{rx\_clk2}=f_{serial}/128=f_{adc\_clk}/64$
bypass	In	1-bit	Active high ADX IP bypass. When high, samples go through the IP without the ADX post-processing treatment.
frame	In	array [31:0] of 16-bit	ESIstream RX frame output signal. Contain 512 ESIstream frames: 12-bit sample, 2 control bits and 2 overhead bits.
valid	In	8-bit	Active high ESIstream RX valid output signal. After synchronizing the ESIstream serial link between the ADC and the FPGA and releasing data from ESIstream RX output buffers, this signal goes high when frame data are valid. One bit per lane.
samples	Out	512-bit	32 x 16-bit samples with **2s-complement coding**. <511 downto 0>= <N+31><N+30><N+…><N+1><N>
samples overflow	Out	32-bit	1-bit per sample indicating an overflow. Active high. Bit 0 corresponds to sample 0 (15 downto 0). Bit 1 corresponds to sample 1 (31 downto 16). … Bit 31 corresponds to sample 31 (511 downto 496).
spi_nss_adx	In	1-bit	SPI multiplexer command selecting ADC EV12AQ600 /605 SPI Master. When low, ADX IP has access to ADC SPI slave When high, Application SPI Master has access to ADC SPI slave or others SPI slaves sharing the same SPI bus, like a PLL.
spi_sclk	In	1-bit	Application SPI Master clock.
spi_mosi	In	1-bit	Application SPI Master data output.
spi_miso	Out	1-bit	Application SPI Master data input.
spi_ncs_adc	In	1-bit	Application active low ADC chip select.
spi_ncs_pll	In	1-bit	Application active low PLL chip select. In case a PLL share the SPI bus with ADC, else set high.
pll_ss_no	Out	1-bit	PLL active low slave select, FPGA output. If no PLL, leave open.
adc_miso_i	In	1-bit	ADC SPI data input (Miso, C1), FPGA output.
adc_mosi_o	Out	1-bit	ADC SPI data output (Mosi, C2), FPGA input.
adc_ss_no	Out	1-bit	ADC active low slave select (Csn, D4), FPGA output.
adc_sck_o	Out	1-bit	ADC SPI clock output (Sclk, D5), FPGA output.

3.1.3 ESIstream 14B/16B frame

See ESIstream 14B/16B protocol specification on www.esistream.com.

3.1.4 ADX4 wrapper frame input

If the SWAP_LANES generic parameter equals to (0, 1, 2, 3, 4, 5, 6, 7), the frame input signal must be organize as defined below:

frame(0)	sampleN
frame(1)	sampleN+8
frame(2)	sampleN+16
frame(3)	sampleN+24

frame(8)	sampleN+2
frame(9)	sampleN+10
frame(10)	sampleN+18
frame(11)	sampleN+26

frame(16)	sampleN+4
frame(17)	sampleN+12
frame(18)	sampleN+20
frame(19)	sampleN+28

frame(24)	sampleN+6
frame(25)	sampleN+14
frame(26)	sampleN+22
frame(27)	sampleN+30

frame(4)	sampleN+1
frame(5)	sampleN+9
frame(6)	sampleN+17
frame(7)	sampleN+25

frame(12)	sampleN+3
frame(13)	sampleN+11
frame(14)	sampleN+19
frame(15)	sampleN+27

frame(20)	sampleN+5
frame(21)	sampleN+13
frame(22)	sampleN+21
frame(23)	sampleN+29

frame(28)	sampleN+7
frame(29)	sampleN+15
frame(30)	sampleN+23
frame(31)	sampleN+31

Figure 3 shows the ADC EV12AQ600 / 605 samples mapping on High-Speed Serial Lanes in 1 channel mode:

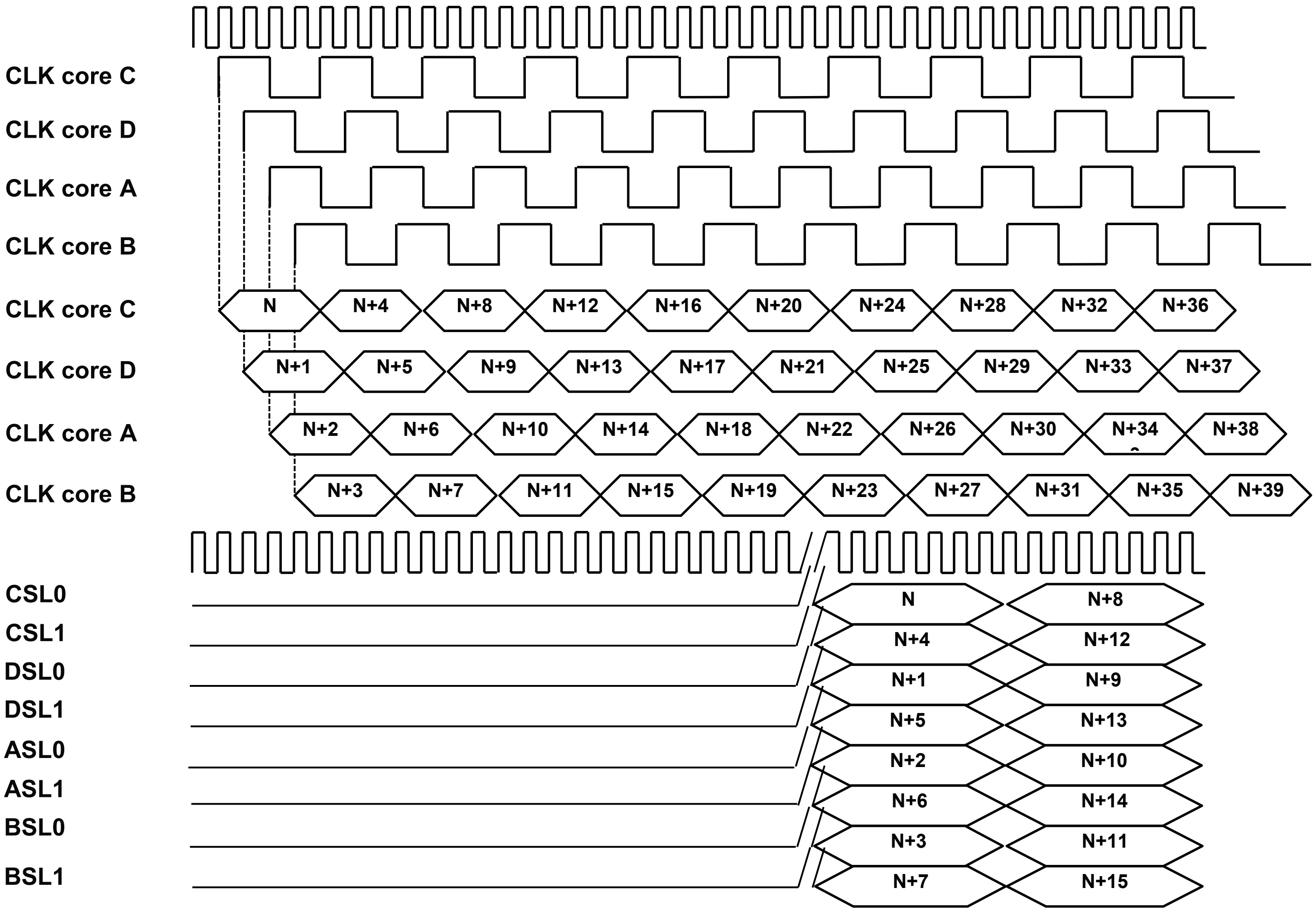


Figure 3 – ADC EV12AQ600, 1 channel mode, timing diagram.

3.2 Swap of High-Speed Serial Lanes

The generic parameter SWAP_LANES allows to compensate High-Speed Serial Lanes swapping on layout, on FPGA mapping or in Gigabit Transceiver (GT).

If the SWAP_LANES generic parameter equal to (**7**, 1, **6**, 3, 4, 5, **2**, **0**), the frame input signal must be organize as defined below:

frame(0)	**sampleN+7**
frame(1)	**sampleN+15**
frame(2)	**sampleN+23**
frame(3)	**sampleN+31**

frame(8)	**sampleN+6**
frame(9)	**sampleN+14**
frame(10)	**sampleN+22**
frame(11)	**sampleN+30**

frame(16)	sampleN+4
frame(17)	sampleN+12
frame(18)	sampleN+20
frame(19)	sampleN+28

frame(24)	**sampleN+2**
frame(25)	**sampleN+10**
frame(26)	**sampleN+18**
frame(27)	**sampleN+26**

frame(4)	sampleN+1
frame(5)	sampleN+9
frame(6)	sampleN+17
frame(7)	sampleN+25

frame(12)	sampleN+3
frame(13)	sampleN+11
frame(14)	sampleN+19
frame(15)	sampleN+27

frame(20)	sampleN+5
frame(21)	sampleN+13
frame(22)	sampleN+21
frame(23)	sampleN+29

frame(28)	**sampleN**
frame(29)	**sampleN+8**
frame(30)	**sampleN+16**
frame(31)	**sampleN+24**

3.3 ADX IP clocks generation

The ADX IP clocks (rx_clk, rx_clk2) must be synchronous with the ADC clock input (CLK).
These clocks can be generated directly from the ADC SSO (MGTREFCLK), from rx_usrclk (GTH clock output) when the GTH user data width is set to 64, or from an external synchronous clock using a FPGA global clock (GC) input.

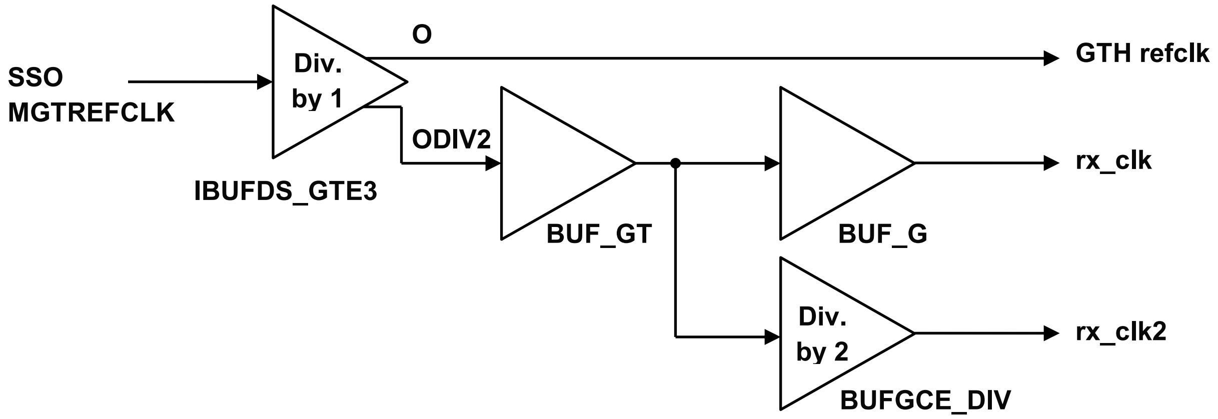


Figure 4 – ADX IP clocks generation from the ADC SSO.

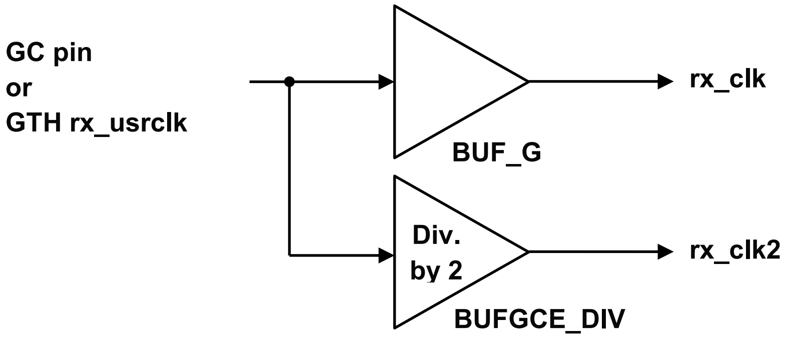


Figure 5 – ADX IP clocks generation from GC input or GTH rx_usrclk output.

4 Usage

For the license procedure to be successful and thus enabling ADX it is critical to perform the system start-up in the following manner.

1. Set **spi_nss_adx input high**. Application SPI Master can access to ADC SPI slave.
2. Set ADX IP in reset by holding **rst input high**. It is important to NOT communicate with any SPI slave before ADX is in reset to avoid access collision.
3. Perform the ADC configuration as described in the EV12AQ600 datasheet section "DIGITAL RESET AND START-UP PROCEDURE". Normal operation of the ADC.
4. Configure the ADC in 1 channel mode
   - ADC CPS_CTRL register address 0x000B bits 2:0 to
     i. "000": **ADC Input 0** to core C & D & A & B
     ii. "001": **ADC Input 3** to core C & D & A & B
   - ADC CLK_MODE_SEL register address 0x000A bits 1:0 to "00" **all core clocks are interleaved.**
5. Configure the ADC with Control Bit 1 (CB1) to InRange bit (default configuration).
   - ADC AB_HSSL_CFG register address 0x0013 bits 1:0 to "00" **InRange selected** for cores A and B.
   - ADC CD_HSSL_CFG register address 0x0014 bits 1:0 to "00" **InRange selected** for cores C and D.
6. Set **spi_nss_adx input low**. ADX4 IP SPI Master can access to ADC SPI slave.
7. Release ADX from reset by set **rst input low**. Application SPI Master can access to ADC PI slave.
8. Wait 10 ms. Application SPI Master can access to ADC SPI slave.

5 Dynamic performance

ADX4 IP achievable dynamic performances are described in the "*ILG recal at Fin*" column of the table and section "*Dynamic Performance - 1-channel mode – 6.4 GSps*" available in ADC EV12AQ600\605 datasheet.

6 Evaluating the ADX4 IP

The EV12AQ600-ADX-EVM demonstration kit is available to start evaluating the ADX4 IP and ADX2 IP.

EV12AQ600-ADX-EVM user guide available here.

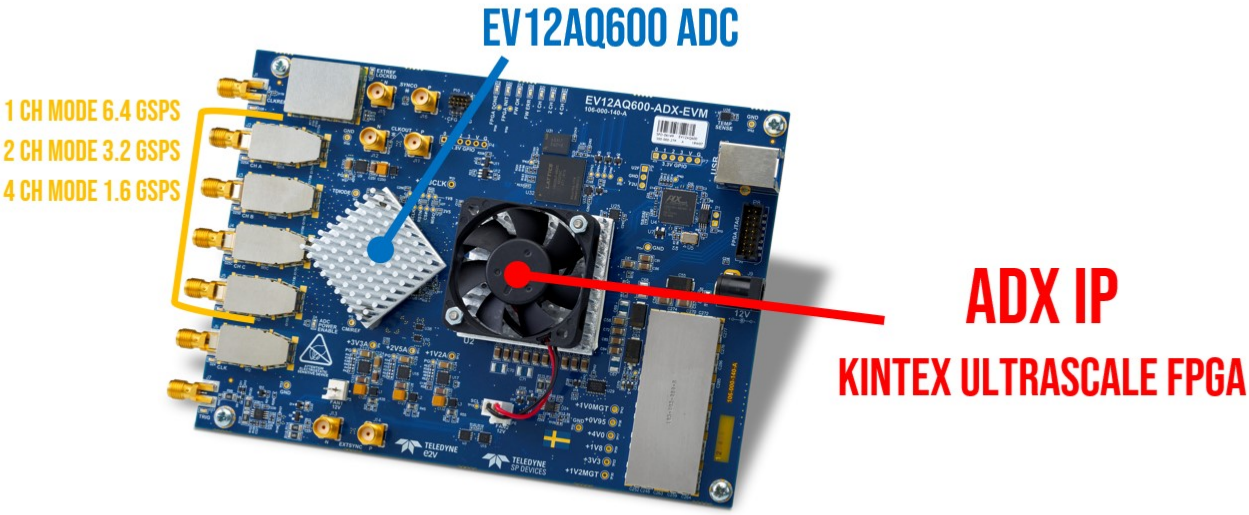


Figure 6 – EV12AQ600-ADX-EVM Demonstration Kit

To order the demonstration kit, please contact your local FAE or send an email at GRE-HOTLINE-BDC@Teledyne.com.